Back-tracing Representative Points for Voting-based 3D Object Detection in Point Clouds

Bowen Cheng[1], Lu Sheng[1]*, Shaoshuai Shi[2], Ming Yang[1], Dong Xu[3]
[1]College of Software, Beihang University
[2]The Chinese University of Hong Kong [3]The University of Sydney
{chengbowen052, lsheng, viv}@buaa.edu.cn ssshi@ee.cuhk.edu.hk dong.xu@sydney.edu.au

Abstract

3D object detection in point clouds is a challenging vision task that benefits various applications for understanding the 3D visual world. Lots of recent research focuses on how to exploit end-to-end trainable Hough voting for generating object proposals. However, the current voting strategy can only receive partial votes from the surfaces of potential objects together with severe outlier votes from the cluttered backgrounds, which hampers full utilization of the information from the input point clouds. Inspired by the back-tracing strategy in the conventional Hough voting methods, in this work, we introduce a new 3D object detection method, named as Back-tracing Representative Points Network (BRNet), which generatively back-traces the representative points from the vote centers and also revisits complementary seed points around these generated points, so as to better capture the fine local structural features surrounding the potential objects from the raw point clouds. Therefore, this bottom-up and then top-down strategy in our BRNet enforces mutual consistency between the predicted vote centers and the raw surface points and thus achieves more reliable and flexible object localization and class prediction results. Our BRNet is simple but effective, which significantly outperforms the state-of-the-art methods on two large-scale point cloud datasets, ScanNet V2 (+7.5% in terms of mAP@0.50) and SUN RGB-D (+4.7% in terms of mAP@0.50), while it is still lightweight and efficient. Code will be available at https://github.com/cheng052/BRNet.

1. Introduction

As one of the fundamental tasks that aims at understanding 3D visual world, 3D object detection would like to predict amodal 3D bounding boxes and associated semantic labels of objects in real 3D scenes. 3D object detection technologies would significantly benefit various downstream real world applications such as augmented reality, robotics and *etc*. In this work, we focus on 3D object detection from point clouds. It is even more challenging because the irregular, sparse and orderless characteristics of this special 3D input make it a hard task to design reliable point-based 3D object detection systems by leveraging the recent progress in 2D object detection.

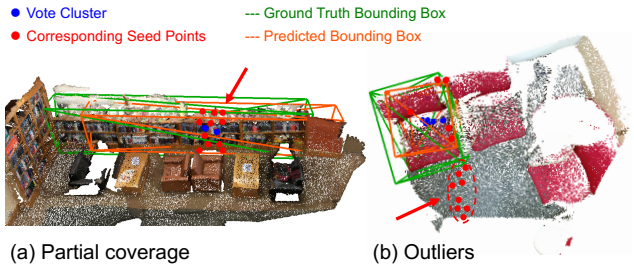


Figure 1. The votes generated by VoteNet [20] and its variants usually suffer from (a) partial coverage of the object surfaces, (b) outliers from the cluttered background. By examining the corresponding seed points, the generated proposals from these votes receive erratic features with respect to the objects, and may be less reliable for predicting accurate bounding boxes, orientations and even semantic classes. Best viewed on screen.

While earlier works resorted to reordering point clouds into regular forms [3, 7, 32, 33, 43], or applying predefined shape templates [15, 19, 40], VoteNet [20] and its variants [36, 41, 2, 1] have shown a great success in designing end-to-end 3D object detection networks based on raw point clouds. VoteNet reformulates the traditional Hough voting process into a point-wise regression problem, and generates an object proposal by sampling a number of seed points from the input point cloud whose votes are within the same cluster. The aggregated feature in each vote cluster is then used to estimate the 3D bounding box (*e.g.* center, size and orientation) and the associated semantic label.

Therefore, the quality of the regressed votes principally determine the reliability of the generated proposals, and then the performance on the object detector. However, al-

*Lu Sheng is the corresponding author.

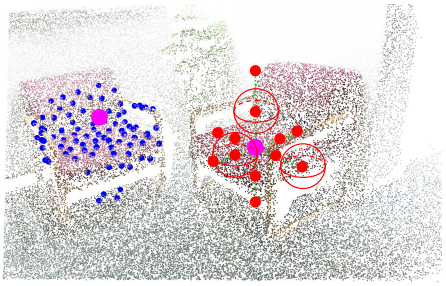


Figure 2. Back-tracing representative points and revisiting seed points. We show the vote cluster center for the two chairs (in purple points). The representative points are back-traced from the vote cluster center (in red points). We set the number of representative points per proposal as 12 in this case, which are illustrated on the right chair. Then, the seed points within a fixed distance of the representative points are revisited, shown in blue points on the left chair. The revisited seed points provide good coverage of the chair's surface, which imply the object shape and keep the structural details as the chair armrest. Best viewed on screen.

though the clustered vote centers are quite accurate, the votes are usually not as representative as our expectation. For example, as illustrated in Fig. 1, by retrieving the seed points of votes from the given vote clusters, these corresponding seed points either partially cover the underlying objects (Fig. 1(a)) or contain severe outliers from the cluttered background (Fig. 1(b)). Therefore as shown in Fig. 1(a), it is undoubted that we cannot accurately predict the bounding box of a long bookshelf if the votes only capture a small area surrounding the vote center. Likewise as shown in Fig. 1(b), the severe outliers make it impossible to accurately detect the chair based on the vote features. Moreover, these seed points are less informative due to the lack of knowledge from the votes, so that there will be less significant gains if we simply back-trace these seed features (as in conventional Hough voting [14]) to improve the voting-based 3D object detection methods.

However, in our point of view, back-tracing is still necessary and could partially address the aforementioned issues with a special design. To be specific, as shown in Fig. 2, we would like to backwardly generate (or trace) the virtual representative points from the center of each vote cluster, and use these virtual points to revisit their surrounding seed points. This generative back-tracing operation indicates possible object shape distributions around the vote center, while the revisited seed features provide complementary local structural clues that may not be fully discovered by the votes. This bottom-up and then top-down process can end up with a mutual interaction that associates the seed features and the vote features, which has the potential to enhance each other features and enable more robust object class prediction and more accurate bounding box regression.

To this end, we propose a new point cloud-based 3D object detection method, named as Back-tracing Representative Point Network (BRNet), by incorporating the end-to-end learnable back-tracing and revisiting operations into the voting-based framework. Specifically, we propose a representative points generation module that generatively samples uniformly distributed representative points within the 3D area of a candidate object, based on the features of a vote cluster center. The generated points can coarsely infer the object bounding boxes even though their sampling process is class-agnostic. The revisited seed points of each representative point are aggregated in a similar way as ROI grid pooling [28], but based on the spatial layout of the representative points. After fusing the aggregated features of the revisited seed points and the features of the vote cluster center, we obtain the refined proposals to eventually detect the objects. Note that the proposed bounding box regression scheme explicitly depends on the spatial distribution of the representative points, thus improves robustness with respect to shape variations within and across object categories.

The contributions of this work are three-fold: (1) the first 3D object detection network, named as BRNet, that successfully adapts the back-tracing step of Hough voting to 3D object detection. (2) an end-to-end learnable network that can generatively *back-trace* the representative points, reliably *revisit* the seed points, and then mutually *refine* the object proposals for more robust object classification and more accurate bounding box regression. (3) the state-of-the-art 3D object detection performance on two benchmark datasets, ScanNet V2 [4] (50.9% in terms of mAP@0.50) and the SUN RGB-D [31] (43.7% in terms of mAP@0.50).

2. Related Works

3D object detection on point clouds. Object detection from 3D point clouds is challenging due to the irregular, sparse and orderless characteristics of 3D points. Earlier attempts usually relied on projections onto regular grids such as multi-view images [3] and voxel grids [43, 37, 12, 7, 30], or based on the candidates from RGB-driven 2D proposal generation [21, 11] or segmentation hypotheses [8], where the existing 2D object detection or segmentation methods based on regular image coordinates can be effortlessly adapted. Other approaches also studied how to exploit discriminative [15, 19] or generative shape templates [40], and high-order contextual potentials to regularize the proposal objectness [16], or used sliding shapes [33, 32], or clouds of oriented gradients (COG) [27].

Thanks to PointNet [22], deep neural networks have become extensively employed onto raw point clouds. For instance, PointRCNN [29] introduced a two-stage 3D object detector, which is analogous to the two-stage 2D object detection methods such as Faster RCNN [26]. Inspired by the Hough voting strategy for 2D object detection and instance segmentation [14], VoteNet [20] was built upon the backbone of PointNet++ [23] and presented an end-

to-end trainable 3D object detector. Later on, the extensions of VoteNet [20], such as MLCVNet [36], HGNet [2] and 3DSSD [39], employed the contextual clues, the hierarchical graph neural networks and the feature-FPS sampling strategy to enable better generation of object proposals. However, these methods heavily depend on the unreliable vote clustering proposed in [20], which is inevitably affected by outliers and usually overlooks inlier seed points. H3DNet [41] partially tackled this issue by introducing a hybrid set of overcomplete geometric primitives to refine the initial bounding boxes predicted by the clustered votes. But these primitives centers are learned with less accurate supervisions and also collected by a similar clustering strategy, thus may still fail to eliminate the outliers or capture sufficient geometric clues to infer the target objects. In this work, we show how to leverage the representative points back-traced from the vote centers to complementarily profile the target objects, which enables more discriminative categorization and more robust bounding box regression.

Anchor-free 2D object detection. The implementation of the back-tracing representative points in our BRNet adopts similar anchor-free localization strategies in 2D object detection. Unlike two-stage 2D object detectors such as Faster RCNN [26], SSD [17] and YOLOv2 [25] that generate proposals with the predefined anchors, the anchor-free detectors [13, 35, 42, 34, 38, 24, 5], especially the regression-based approaches [34, 38, 10, 24, 5], either directly regress borders [24], regress the object boundaries with an iterative dynamic sampling strategy [38], or regress 4D offsets as the surrogate of the localization results[34]. Inspired by these methods, in our method, the back-tracing process relies on a *class-agnostic* offset regressor to retrieve the representative points that indicate the likely shape profile surrounding each vote center and thus provides more local structural clues for latter inference. Rather than localization constrained by predefined *class-aware* statistics, as in VoteNet [20] and its successors, the proposed BRNet benefits more flexible regression without losing its discriminative power.

Back-tracing in voting-based object detection and instance segmentation. Leibe *et al*. [14] applied the hough voting strategy for simultaneously 2D object detection and instance segmentation. The core part of this approach is a learned highly flexible representation for object shapes in a probabilistic extension of Generalized Hough Transform. Moreover, the work in [9] combined the top-down clues available from object detection and the bottom-up power of Markov Random Fields (MRFs) when performing class-specific object detection and segmentation in 3D scenes. These methods rely on a top-down strategy such as back-tracing object hypotheses to enhance the bottom-up strategy such as Hough voting. Their mutual agreement enhances each other, and thus devotes to the success of more reliable object detection. The proposed BRNet also follows this idea with a new end-to-end trainable back-tracing process based on the representative points. Recently, as a 3D instance segmentation method, 3D-MPA [6] applied a "direct" back-tracing strategy to cluster the surface points from the corresponding votes in one cluster. In contrast, our method alleviates the inherent partial coverage and outlier issues from the "generative" back-tracing strategy.

3. Methodology

In this section, we describe the technical details of our BRNet. Sec. 3.1 presents an overview of our method. In Sec. 3.2 to Sec. 3.5, we elaborate the network architecture and the learning objective of our BRNet.

3.1. Overview

As illustrated in Fig. 3, the input of our BRNet is a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, with a 3D coordinate for each of the N points. Such an input typically comes from multi-view stereo (*e.g*. ScanNet [4]) or depth sensors (*e.g*. SUN RGB-D [31]). The output is a collection of (oriented) bounding boxes $\mathcal{B}$, each box $b \in \mathcal{B}$ is associated with a predefined category label $l_b \in \mathcal{C}$, a center $\mathbf{c}_b = [c_b^x, c_b^y, c_b^z]^\top \in \mathbb{R}^3$ in a world coordinate system, the size of bounding box $\mathbf{s}_b = [s_b^x, s_b^y, s_b^z]^\top \in \mathbb{R}^3$, an orientation angle θ_b in the xy-plane of the same world coordinate system.

BRNet consists of four main modules: (1) vote generation and clustering, (2) back-traced representative points generation, (3) seed point revisiting, and (4) proposal refinement and classification followed by standard 3D NMS. In the first module, we follow the same network and training strategy as in VoteNet [20] to generate the seed points, the votes and the vote clusters. We will elaborate the other three modules in the following parts.

3.2. Generating Back-traced Representative Points

The conventional back-tracing step of Hough voting for identifying object boundaries [14] is less reliable for amodal object detection from partial observations, as it just picks up seed points that contribute to the selected votes. For example, in VoteNet [20], these back-traced seed points can only capture local geometric area near the cluster center while containing the outliers from the cluttered background in the meantime. VoteNet [20] circumvents this issue by removing the back-tracing step and using a PointNet-like set aggregation block just for votes, and then generates the object proposals and classifies them. However, the aforementioned incompleteness issue and the outliers within the votes (delivered from the seed points) are clearly harmful for the detection task. To this end, we argue that it is still beneficial to use back-tracing in point-based 3D object detection, but it requires a better tracing strategy to effectively find the representative seed points. In contrast to the conventional back-tracing strategy, we propose a representative

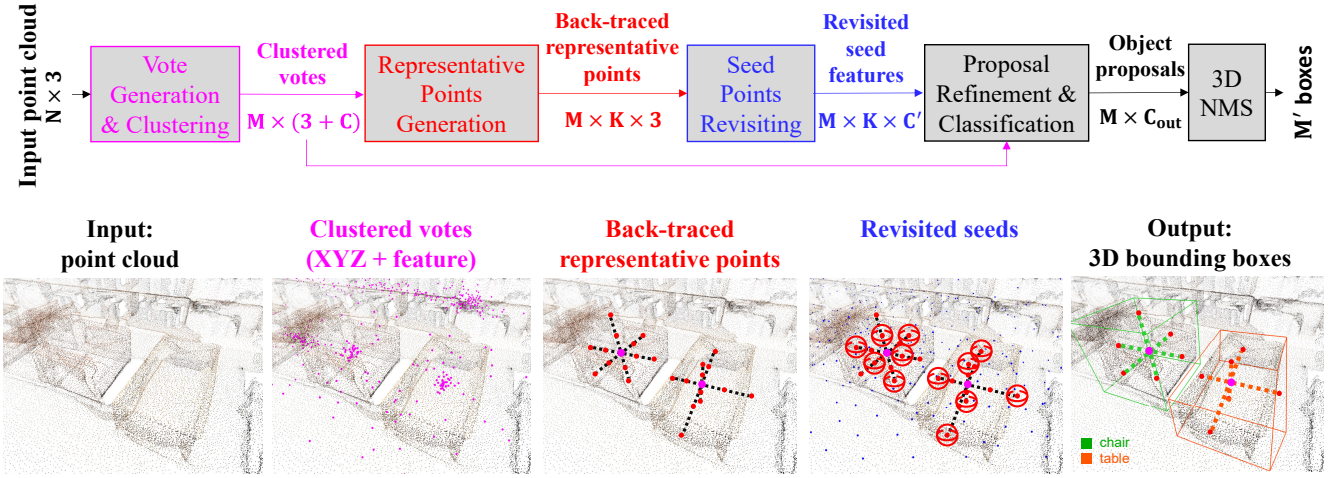


Figure 3. An overview of our proposed BRNet for 3D object detection in point clouds. Given an input point cloud consisting of N points with the XYZ coordinates, we generate votes from it and group the votes into M clusters as in VoteNet [20]. For each of the M vote cluster centers, we back-trace K representative points from it. The back-traced representative points imply the possible area of the object. We then revisit the seed points around the representative points and aggregate the surrounding seed point features to the representative points. The clustered vote features and the revisited seed features are fused and processed by the proposal refinement and classification module to produce the refined representative points and object's semantic category, which can be easily transformed into 3D object bounding boxes. The standard 3D NMS is eventually used to generate the final detection results. Best viewed on screen.

point generation (RPG) module to backwardly regress the *virtually* generated representative points from the votes in a *generative* manner. The generated representative points are uniformly distributed within the potential 3D area of a candidate object, which can also indicate 3D object shapes when interacted with their actual surrounding seed points.

To be specific, the vote sampling and grouping block generates a set of vote cluster centers $\{\mathbf{v}_i\}_{i=1}^{M}$, where $\mathbf{v}_i = [\mathbf{p}_i^\top, \mathbf{f}_i^\top]^\top$ with $\mathbf{p}_i \in \mathbb{R}^3$ as the vote's geometric position in the 3D space and $\mathbf{f}_i \in \mathbb{R}^C$ as its feature extracted from the preceding network, M is the number of vote clusters. Then, the RPG module generates a set of representative points for each vote cluster center. Rather than directly sampling the 3D coordinates of these points, this module simultaneously predicts the tentative orientation $\theta_i \in [0, 2\pi]$ of the potential object, and regresses the offset distances $\mathbf{x}_i \in \mathbb{R}^6$ from $\mathbf{v}_i$ to the tentative object's surface in 6 canonical directions (*i.e.* front/back/left/right/up/down), and then uniformly samples distributed representative points $\mathcal{R}_i = \{\mathbf{r}_i^k = (x_i^k, y_i^k, z_i^k)\}_{k=1}^{K}$ along these directions (which are skewed by the predicted orientation) within the range of the offset distances. K is the number of representative points. In this work, we sample 2 uniformly distributed points within the range of each offset, thus $K = 2 \times 6 = 12$ in total.

Network architecture and learning. The RPG module is implemented by using multi-layer perceptrons (MLP) with the ReLU activation function and batch normalization. It takes the feature $\mathbf{f}_i$ from the vote center $\mathbf{v}_i$ as the input, and its output is the set $\{\mathbf{x}_i, \theta_i\}$. We employ $\exp(\cdot)$ to map any real number to $(0, \infty)$ on the output of $\mathbf{x}_i$. This module is supervised by the ground-truth (GT) offsets as the vote center can be assigned to a GT object, *i.e.*

$$L_{\text{rep-off}} = \frac{1}{M_{\text{pos}}} \sum_{i=1}^{M} \|\mathbf{x}_i - \mathbf{x}_i^*\|_\rho \cdot \mathbb{I}[\mathbf{v}_i \text{ is positive}], \quad (1)$$

where $\mathbb{I}[\mathbf{v}_i \text{ is positive}]$ indicates whether the vote center $\mathbf{v}_i$ is around a GT object center (within a radius of 0.3). M_{pos} is the number of positive vote centers. ρ means smooth-ℓ_1 norm. And $\mathbf{x}_i^*$ is the GT offsets from the vote center $\mathbf{v}_i$ to the 6 faces of the GT bounding box. This module is also supervised by the GT orientation of the same GT object. To better predict the orientation angle, we adopt the bin-based angle prediction scheme as in [21], which predicts a classification score for each orientation bin and a regression offset in each bin, and then uses the cross-entropy loss for orientation bins, and the smooth-ℓ_1 loss for the regression offset. We term the orientation loss as $L_{\text{rep-ang}}$. Therefore, the final learning objective for this module is

$$L_{\text{rep}} = \lambda L_{\text{rep-off}} + L_{\text{rep-ang}}, \quad (2)$$

where $\lambda = 20$ is used to balance the two terms.

3.3. Revisiting Seed Points

By back-tracing the representative points $\mathcal{R}_i$ in a generative manner from a vote center $\mathbf{v}_i, i = 1, \ldots, M$, we can roughly obtain the size and the position of a possible object in a class-agnostic way, but it still requires mutual consistency from the actual seed points in order to reliably gener-

ate the object proposals for more accurate object localization, bounding box estimation and object class prediction. To be specific, we revisit the seed points $\{\mathbf{q}_j | \|\mathbf{q}_j - \mathbf{r}_i^k\| \leq \delta\}$ within a fixed radius (δ=0.2 in the work) surrounding a back-traced representative point $\mathbf{r}_i^k, k = 1, \ldots, K$, and aggregate the revisited seed features by using a PointNet-like block [22], denoted as $\tilde{\mathbf{g}}_{\mathbf{r}_i^k}$. This process is similarly implemented as ROI-grid pooling proposed in PV-RCNN [28], but with a different griding and radius selection strategy.

Thereafter, to each vote center (or called proposal) $\mathbf{v}_i$, the set of aggregated seed point features $\tilde{\mathbf{g}}_{\mathbf{r}_i^k}$ from each representative point $\mathbf{r}_i^k$ can be further fused into a single feature $\tilde{\mathbf{g}}_{\mathbf{v}_i}$, which is implemented by concatenating $\{\tilde{\mathbf{g}}_{\mathbf{r}_i^k}\}_{k=1}^K$ in a predefined order before being projected to a 128-dimensional feature. The predefined order should be consistent for each proposal, but different ordering strategies do not affect the performance. The revisited seed features are summarized into $\tilde{\mathbf{g}}_{\mathbf{v}_i}$, which thus captures the local object-level features from the relatively precise raw point clouds instead of the predicted vote points.

3.4. Proposal Refinement and Classification

The back-traced representative point set $\mathcal{R}_i$ helps to revisit the seed points and aggregate the local geometric clues from the potential object indicated by the vote center $\mathbf{v}_i$. The aggregated feature $\tilde{\mathbf{g}}_{\mathbf{v}_i}$ can be concatenated with the feature $\mathbf{f}_i$ of the vote center $\mathbf{v}_i$, and then refine the proposal and use for more discriminative object class prediction. To this end, the fused feature $\tilde{\mathbf{f}}_i = [\tilde{\mathbf{g}}_{\mathbf{v}_i}^\top, \mathbf{f}_i^\top]^\top \in \mathbb{R}^{256}$ is fed into a shared MLP to predict the residuals $\Delta\mathbf{x}_i$ and $\Delta\theta_i$ based on the preceding estimation results $\mathbf{x}_i$ and θ_i, and produce the final output set $\{\mathbf{x}_i + \Delta\mathbf{x}_i, \theta_i + \Delta\theta_i\}$. Meanwhile, we predict the objectness score and the semantic classification score for each fused feature, similarly as in [20]. Note that the final offsets $\mathbf{x}_i + \Delta\mathbf{x}_i$ can be reformulated as the bounding box size $\mathbf{s}_i = [s_i^x, s_i^y, s_i^z]^\top \in \mathbb{R}^3$ and the object center $\mathbf{c}_i = [c_i^x, c_i^y, c_i^z]^\top \in \mathbb{R}^3$, by min-max clipping the final representative point set $\tilde{\mathcal{R}}_i$ in the canonical coordinate.

3.5. The Learning Objective

In summary, the loss function of the entire framework of the newly proposed BRNet is defined as following:

$$L = L_{\text{vote-reg}} + \lambda_{\text{obj-cls}} L_{\text{obj-cls}} + \lambda_{\text{sem-cls}} L_{\text{sem-cls}} + \lambda_{rep} L_{\text{rep}} + \lambda_{\text{refine}} L_{\text{refine}} \quad (3)$$

Following the terms and label assignment strategy used in VoteNet [20], the loss terms $L_{\text{vote-reg}}$, $L_{\text{obj-cls}}$, $L_{\text{sem-cls}}$ indicate the per-point vote regression loss, the objectness loss and the semantic classification loss, respectively. L_{rep} is defined in Sec. 3.2. L_{refine} is used to supervise the residuals from the initial representative point sets to the final representative point sets:

$$L_{\text{refine}} = \frac{1}{M_{\text{pos}}} \sum_{i=1}^{M} (\lambda \|\mathbf{x}_i + \Delta\mathbf{x}_i - \mathbf{x}_i^*\|_\rho + \|(\theta_i + \Delta\theta_i - \theta_i^*\|_\rho) \cdot \mathbb{I}[\mathbf{v}_i \text{ is positive}] \quad (4)$$

ρ denotes the smooth-ℓ_1 norm. θ_i^* is the orientation angle of the ground-truth object bounding box. L_{refine} is computed only on the positive vote clusters. The weighting factors are $\lambda_{\text{obj-cls}} = 1$, $\lambda_{\text{sem-cls}} = 0.1$, $\lambda_{\text{rep}} = 1$, $\lambda_{\text{refine}} = 1$ and $\lambda = 20$.

4. Experiments

4.1. Setups and Implementation Details

Datasets. We evaluate our method on two large-scale indoor scene datasets, *i.e.* SUN RGB-D [31] and ScanNet V2 [4]. SUN RGB-D consists of $10,355$ single-view indoor RGB-D images annotated with the oriented 3D bounding boxes and the semantic labels for 37 categories. The point clouds are converted from the depth maps based on the provided camera parameters. The captured point clouds contain severe occlusions and holes, thus are challenging for 3D object detection. ScanNet V2 is a 3D mesh dataset about $1,500$ 3D reconstructed indoor scenes. It contains 18 object categories with densely annotated axis-aligned bounding boxes. The scans in the ScanNet V2 dataset are more complete with more objects than those in the SUN RGB-D dataset. For both datasets, we use the same data preparation and training/validation split as in VoteNet [20].

Input and data augmentation. The input of our method is a point cloud randomly sub-sampled from the raw data of each dataset, *i.e.*, $20,000$ points from a point cloud in the SUN RGB-D dataset, and $40,000$ points from a 3D mesh in the ScanNet V2 dataset. We also include the height feature to each point. To augment the training data, we add random flipping, rotating and scaling to the input point clouds, as the way employed by VoteNet [20].

Network training details. Our network is end-to-end optimized by using the Adam optimizer with the batch size as 8. The base learning rates are 0.001 for the SUN RGB-D [31] dataset and 0.005 for the ScanNet V2 [4] dataset. We train the network for 220 epochs on both datasets. The cosine annealing learning rate strategy[18] is adopted for learning rate decay. Based on PyTorch platform equipped with one NVIDIA GeForce RTX 2080 Ti GPU card, it takes around 4 hours to train the model on the ScanNet V2 dataset, while it takes around 12 hours on the SUN RGB-D dataset.

Inference and evaluation. Our method takes the point clouds of the entire scenes as the inputs and outputs the object proposals. The proposals are post-processed by a 3D NMS module with an IoU threshold of 0.25. The evaluation follows the same protocol as in [33] using mean average precision, especially mAP@0.25 and mAP@0.50.

Table 1. 3D object detection results on the ScanNet V2 validation set(left) and the SUN RGB-D V1 validation set(right). Evaluation metric is average precision with 3D IOU thresholds as 0.25 and 0.50. *Note for fair comparison, we report the results of H3DNet on the ScanNet V2 dataset under both 1 and 4 PointNet++ backbones (BB) settings. While we only report the result of H3DNet with 4 PointNet++ backbones (BB) on the SUN RGB-D dataset, as the work [41] only reports the result under this setting.

ScanNet V2	Input	mAP@0.25	mAP@0.50
DSS [33]	Geo + RGB	15.2	6.8
F-PointNet [21]	Geo + RGB	19.8	10.8
GSPN [40]	Geo + RGB	30.6	17.7
3D-SIS [7]	Geo + 5 views	40.2	22.5
VoteNet [20]	Geo only	58.6	33.5
HGNet [2]	Geo only	61.3	34.4
MLCVNet [36]	Geo only	64.7	42.1
H3DNet (1BB)* [41]	Geo only	64.4	43.4
H3DNet (4BB)* [41]	Geo only	**67.2**	48.1
Ours	Geo only	66.1	**50.9**

SUN RGB-D	Input	mAP@0.25	mAP@0.50
DSS [33]	Geo + RGB	42.1	-
COG [27]	Geo + RGB	47.6	-
2D-driven [11]	Geo + RGB	45.1	-
F-PointNet [21]	Geo + RGB	54.0	-
VoteNet [20]	Geo only	57.7	32.9
HGNet [2]	Geo only	**61.6**	-
MLCVNet [36]	Geo only	59.8	-
H3DNet (1BB)* [41]	Geo only	-	-
H3DNet (4BB)* [41]	Geo only	60.1	39.0
Ours	Geo only	61.1	**43.7**

Table 2. 3D object detection results on the ScanNet V2 validation set. The evaluation metric is the average precision with 3D IOU threshold as 0.50. *Note that for H3DNet only the per-category results with 4 PointNet++ backbones are reported in [41].

ScanNet V2	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	avg
VoteNet [20]	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
MLCVNet [36]	16.6	**83.3**	78.1	74.7	55.1	28.1	17.0	**51.7**	3.7	13.9	47.7	28.6	36.3	13.4	70.9	25.6	85.7	27.5	42.1
H3DNet* [41]	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	**28.7**	80.6	**81.9**	**80.6**	**60.8**	**35.5**	**22.2**	48.0	**7.5**	**43.7**	**54.8**	**39.1**	**51.8**	35.9	88.9	**38.7**	84.4	33.0	**50.9**

4.2. Comparisons with the State-of-the-art Methods

We compare our method with a list of reference methods, for example the earlier attempts, such as COG [27], DSS [33] and 3D-SIS [7], 2D-driven [11] and F-PointNet [21], and GSPN [40], and the recent point cloud-based state-of-the-art methods such as VoteNet [20] and its successors MLCVNet [36], HGNet [2] and H3DNet [41].

Quantitative results. The comparison results are summarized in Table 1. Our method outperforms all baseline methods by remarkable performance gains, for example more than 7.5% and 4.7% improvement in terms of the mAP@0.50 metric on the validation sets of ScanNet V2 and SUN RGB-D respectively. Note that mAP@0.50 is a fairly challenging metric as it basically requires more than 79% coverage in each dimension of a bounding box, which indicates that back-tracing representative points can significantly improve the localization accuracy. Notably, MLCVNet [36] works well on the ScanNet dataset but achieves relatively poor performance on the SUN RGB-D dataset, while HGNet [2] works well on the SUN RGB-D dataset but achieves poor result on the ScanNet dataset, especially in terms of the mAP@0.50 metric. Our method works well on both datasets, which indicates its stronger generalization ability for different detection scenarios. ScanNet contains relative complete 3D reconstructed meshes, while SUN RGB-D consists of single-view RGB-D scans with severe occlusions and holes. Moreover, H3DNet [41] ensembles 4 PointNet++ [23] backbones to achieve the reported result on the SUN RGB-D dataset, while our model only needs one backbone as the base feature extractor. It further validates it is effective to back-trace the representative points for reliably parsing the object proposals. As shown in Table 2, our method performs the best on 12 classes among 18 total classes from the ScanNet dataset in terms of mAP@0.50. While our method only uses one PointNet++ backbone for point cloud feature extraction, it outperforms H3DNet [41] with 4 PointNet++ backbones. Moreover, it achieves better performance on the categories (*e.g.* "cabinet", "chair", "sofa", "table", "counter" and "desk") with irregular sizes or shapes, as its back-tracing and revisiting process removes the outliers from the votes and enables better mutual agreement between the votes and the local object surfaces, whilst its class-agnostic regression strategy makes the estimation process robust to shape variations.

Qualitative results. In Fig. 4 and Fig. 6, we visualize the representative 3D object detection results, from our method and the baseline methods, such as VoteNet [20], MLCVNet [36] and H3DNet [41]. These results demonstrate that our method achieves more reliable detection results with more accurate bounding boxes and orientations. Our method also eliminates false positives and discovers more missing objects when compared with the baseline methods[1].

4.3. Ablation Study and Discussions

Class-agnostic bounding box regression. Our method regresses the representative points in a class-agnostic way, which are then converted to the proposal's bounding boxes.

[1]MLCVNet does not provide a checkpoint for the SUN RGB-D dataset [31] thus we cannot provide its visualization results on this dataset.

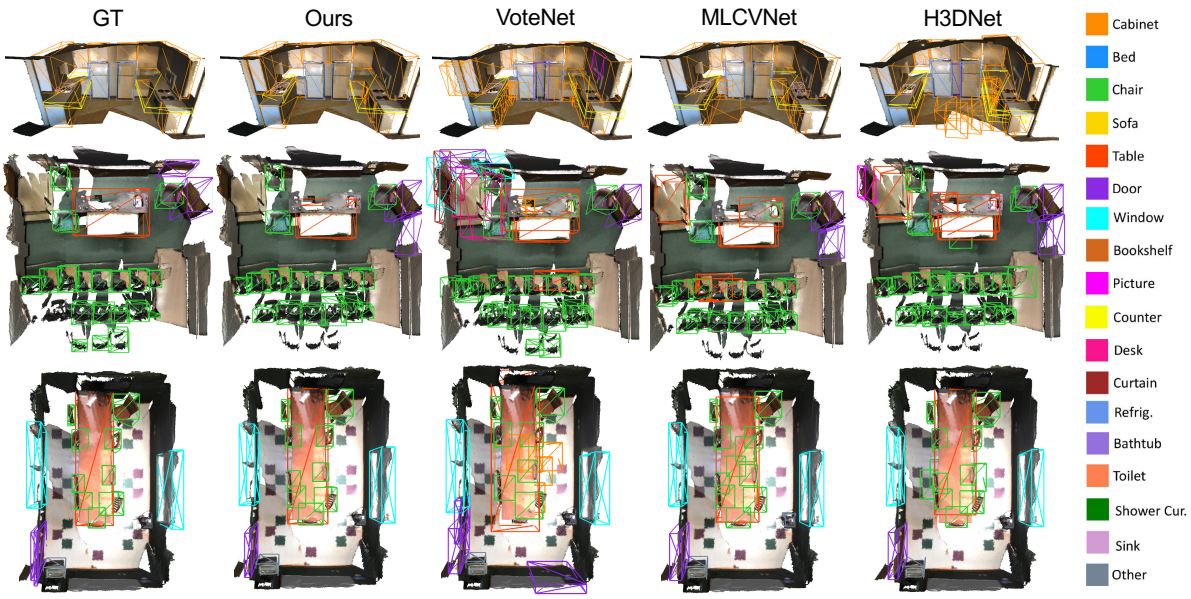


Figure 4. Qualitative results of different 3D object detection methods on ScanNet V2 dataset [4]. The baseline methods are VoteNet [20], MLCVNet [36] and H3DNet [41]. Best viewed on screen.

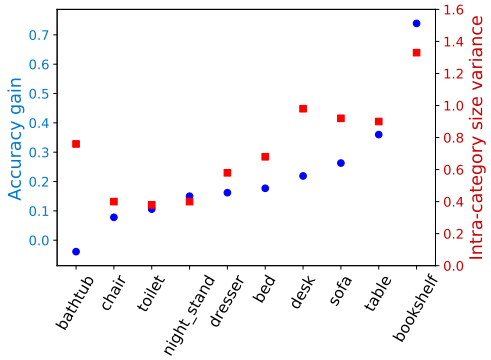


Figure 5. Class-agnostic bounding box regression works better on the categories with high intra-category size variances. For each category we show the relative accuracy gain (in blue dots) of the alternative method "VoteNet+CA-Reg" over VoteNet [20] and intra-category size variance(in red squares), which is normalized by the mean category size.

Note VoteNet [20] and its variants [2, 36, 41] have to estimate the sizes of object proposals in a class-aware way. Thus these baseline methods usually output the object sizes that can only moderately vary around the class-aware templates, and tend to falsely detect the objects when their sizes are unusual. To validate this observation, we implement an alternative method that employs a similar regression strategy as in our method but shares the same network as VoteNet [20]. We term this variant as "VoteNet+CA-Reg". As shown in Table 3, this variant significantly outperforms VoteNet. As shown in Figure 5, we also observe that this alternative method works better for the categories with high intra-category variance in sizes, and the mAP@0.50 gains of this alternative method over VoteNet on the SUN RGB-D dataset are positively related to size variances.

Table 3. Quantitative ablation experiments on ScanNet V2 and SUN RGB-D datasets. "+CA-Reg" means VoteNet [20] with a class-agnostic bounding box regressor, "+Seed-Pts" indicates VoteNet with votes fused with their corresponding seed points.

	ScanNet V2		SUN RGB-D	
	mAP@0.25	mAP@0.50	mAP@0.25	mAP@0.50
VoteNet	58.6	33.5	57.7	32.9
+CA-Reg	59.3	40.8	58.2	37.6
+Seed-Pts	59.1	37.6	59.5	33.6
Ours	**66.1**	**50.9**	**61.1**	**43.7**

Back-tracing, revisiting and refinement. Back-tracing the representative points should also be combined with the subsequent revisiting and refinement modules. As shown in Table 3, we find this complete method has significant performance gains ($\sim$ 10% mAP improvement on ScanNet and $\sim$ 6% mAP improvement on SUN RGB-D in terms of mAP@0.50) over the aforementioned baseline. The back-tracing operation gives rough estimation of the object extent, and the revisiting and refining operations further update the proposal features with the reliable seed features in the neighborhood, thus offering better chance to produce more accurate detection results. Moreover, as shown in Figure 7, the revisited seed points by our method compactly cover the object's surface, while the corresponding seed points retrieved by the votes can only partially cover the surface, and also suffer from the outliers.

Moreover, to validate whether the seed points can help improve the object detection results, we consider another variant (termed as "VoteNet+Seed-Pts") that VoteNet has its vote features fused with the corresponding seed points' features. In comparison to VoteNet, this alternative method also achieves non-trivial gains on both datasets, especially

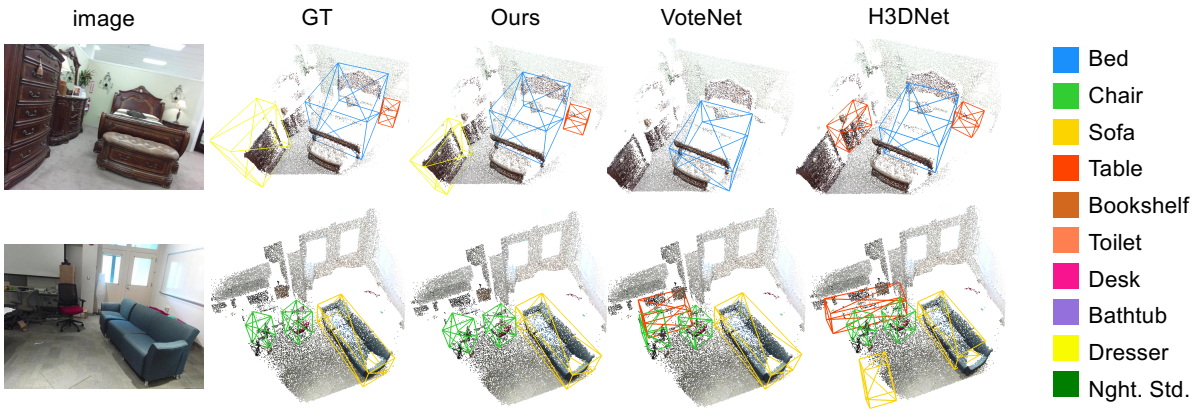
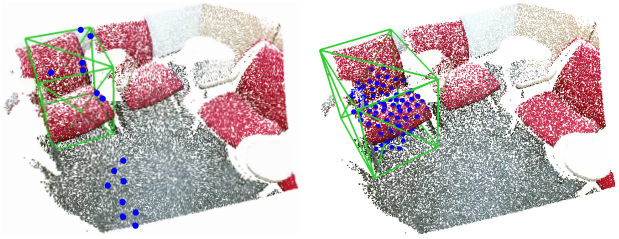


Figure 6. Qualitative comparison results of the 3D object detection methods on the SUN RGB-D dataset [31]. The baseline methods are VoteNet [20] and H3DNet [41]. Best viewed on screen.



(a) Corresponding seed points (b) Revisited seed points

Figure 7. Comparison between the corresponding seed points and the revisited seed points. The seed points are marked as blue points and the predicted bounding boxes are the green boxes. The revisited seed points completely cover the chair, while the corresponding seed points suffer from partial coverage and the outliers.

on ScanNet V2 in terms of mAP@0.50.

Sampling strategy of representative points. In Table 4, we compare different sampling strategies to generate our representative points. "Ray" means uniform sampling along 6 directions between 0 and the maximum offsets. "Grid" means uniform sampling within the 3D bounding box spanned based on the predicted offsets. "#Pts" is the number of sampled points. Our methods using different strategies are generally comparable.

Model size and speed. As listed in Table 5, our proposed method is efficient in comparison to VoteNet, and is 3× faster than the current state-of-the-art H3DNet [41], when evaluated on both datasets. Its model size is marginally increased from that of VoteNet, and around 4× smaller than that of H3DNet. Knowing that the proposed method has significant performance gains than these reference methods (as discussed in Sec. 4.2), its lightweight model validates that the proposed back-tracing strategy is significant for 3D object detection in point clouds[2].

Number of Backbones. Our BRNet can also be improved after using 4 backbones, and it achieves the result of 51.8% in terms of mAP@0.50 on ScanNet [4], which outperforms H3DNet (4 backbones) with a remarkable margin (+3.7%).

[2]Note that MLCVNet does not provide a checkpoint for the SUN RGB-D dataset, we omit its comparison on this dataset.

Table 4. Results of BRNet using different RP sampling strategies.

Types	#Pts	ScanNet V2		SUN RGB-D	
		mAP@0.25	mAP@0.50	mAP@0.25	mAP@0.50
Ray	6	65.0	48.3	60.3	42.7
Ray	12	**66.1**	**50.9**	**61.1**	**43.7**
Ray	18	65.8	48.4	60.4	42.9
Grid	8	65.4	49.1	59.9	42.2
Grid	27	66.0	49.2	60.2	42.5

Table 5. Model size and processing time comparison of different methods, which are evaluated on a NVIDIA GeForce RTX 2080 Ti GPU card with the same configuration. #BB means the number of backbones used for feature extraction.

Method	#BB	Model size	ScanNet	SUN RGB-D
VoteNet [20]	1	11.2MB	0.130s	0.076s
MLCVNet [36]	1	13.9MB	0.141s	-
H3DNet [41]	4	56.0MB	0.330s	0.241s
Ours	1	12.9MB	0.133s	0.079s

5. Conclusion

In this work, we have introduced a new approach to improve the voting-based 3D object detection method by generatively and class-agnostically back-tracing the representative points. We revisit the seed points around the back-traced representative points and extract fine object surface features to generate the high-quality object proposals. Comprehensive ablation studies show the importance and effectiveness of the proposed back-tracing, revisiting and refinement operations. Qualitative and quantitative results further demonstrate that our method remarkably outperforms the existing methods while bringing negligible increases in model size and executive time compared with VoteNet [20].

Acknowledgements. This work was supported by Key Research and Development Program of Guangdong Province, China, under Grant No. 2019B010154003, and the National Natural Science Foundation of China under Grant No. 61906012. We thank Zizheng Que and Zinuo You for valuable discussions and feedback.

References

[1] Syeda Mariam Ahmed and Chee Meng Chew. Density-based clustering for 3d object detection in point clouds. In *CVPR*, pages 10608–10617, 2020. 1

[2] 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 *CVPR*, pages 392–401, 2020. 1, 3, 6, 7, 11, 12

[3] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, pages 1907–1915, 2017. 1, 2

[4] 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. 2, 3, 5, 7, 8, 12, 13

[5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019. 3

[6] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, pages 9031–9040, 2020. 3

[7] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *CVPR*, pages 4421–4430, 2019. 1, 2, 6

[8] Byung-soo Kim, Shili Xu, and Silvio Savarese. Accurate localization of 3D objects from RGB-D data using segmentation hypotheses. In *CVPR*, pages 3182–3189, 2013. 2

[9] Jan Knopp, Mukta Prasad, and Luc Van Gool. Scene cut: Class-specific object detection and segmentation in 3D scenes. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 180–187. IEEE, 2011. 3

[10] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi. Foveabox: Beyound anchor-based object detection. *IEEE TIP*, 29:7389–7398, 2020. 3

[11] Jean Lahoud and Bernard Ghanem. 2D-driven 3D object detection in RGB-D images. In *ICCV*, pages 4622–4630, 2017. 2, 6, 12

[12] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2

[13] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 3

[14] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008. 2, 3

[15] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3D reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446. Wiley Online Library, 2015. 1, 2

[16] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *ICCV*, pages 1417–1424, 2013. 2

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 3

[18] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[19] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM TOG*, 31(6):1–10, 2012. 1, 2

[20] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14

[21] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, pages 918–927, 2018. 2, 4, 6, 12

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, pages 652–660, 2017. 2, 5

[23] 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. 2, 6

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 3

[25] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 3

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2016. 2, 3

[27] Zhile Ren and Erik B Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, pages 1525–1533, 2016. 2, 6, 12

[28] 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. In *CVPR*, pages 10529–10538, 2020. 2, 5

[29] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 2

[30] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint arXiv:1907.03670*, 2019. 2

[31] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015. 2, 3, 5, 6, 8, 12, 13, 14

[32] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In *ECCV*, pages 634–651. Springer, 2014. 1, 2

[33] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *CVPR*, pages 808–816, 2016. 1, 2, 5, 6, 12

[34] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 3

[35] Lachlan Tychsen-Smith and Lars Petersson. DeNet: Scalable real-time object detection with directed sparse sampling. In *ICCV*, pages 428–436, 2017. 3

[36] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. MLCVNet: Multi-level context votenet for 3D object detection. In *CVPR*, pages 10447–10456, 2020. 1, 3, 6, 7, 8, 11, 12, 13

[37] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[38] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. RepPoints: Point set representation for object detection. In *ICCV*, pages 9657–9666, 2019. 3

[39] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3D-SSD: Point-based 3D single stage object detector. In *CVPR*, pages 11040–11048, 2020. 3

[40] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. GSPN: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, pages 3947–3956, 2019. 1, 2, 6

[41] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3DNet: 3D object detection using hybrid geometric primitives. In *ECCV*, 2020. 1, 3, 6, 7, 8, 11, 12, 13, 14

[42] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, pages 850–859, 2019. 3

[43] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *CVPR*, pages 4490–4499, 2018. 1, 2

A. Supplementary

This supplementary provides more quantitative results of our method (Sec. A.1), more qualitative results (Sec. A.1), and finally implementation details (Sec. A.3).

A.1. More Quantitative Results

Finer performance evaluations. We try to evaluate our method using mean average precision with multiple IoU thresholds for finer performance evaluations in Table S1 and S2. We use mAP@0.25, mAP@0.50, mAP@0.75 to evaluate different methods, *i.e*. VoteNet [20], HGNet [2], MLCVNet [36], H3DNet [41] and our BRNet .

Our method performs the best on the metrics mAP@0.50 and mAP@0.75. Notably, mAP@0.75 requires more than 90% coverage in each dimension of a bounding box, which is very challenging for a detector. Our method gains -1.1%, 2.8%, 3.7% increase on mAP@0.25, mAP@0.50, mAP@0.75 compared with H3DNet [41] using 4 PointNet++ backbones and doubled input point clouds (*i.e*., $20,000$ points by our BRNet , and $40,000$ points by H3DNet) on the ScanNetV2 dataset. On more challenging evaluation metrics, our method has more gain, which shows the importance of our representative point generation, and its benefits for seed points revisiting and finer surface feature extraction to accurately detect objects with more reliable bounding boxes.

Per-category results. We show the per-category results on ScanNet V2 dataset with 3D IoU threshold 0.25 in Table S3, and the per-category results on SUN RGB-D with both 3D IoU thresholds 0.25 and 0.50 in Table S4 and S5. In terms of the accuracy about the object detection, our approach outperforms the baseline VoteNet [20] and prior state-of-the-art method H3DNet [41] significantly. For objects in the SUN RGB-D dataset, our approach can gain 7.9%, 10.9%, 6.7%, 7.6%, 6.3% increase on Bathtub, Bed, Dresser, Nightstand and Sofa compared with H3DNet [41]. These improvements are achieved by using back-tracing and seed points revisiting to better capture object surface features.

Table S1. 3D object detection results on ScanNetV2 dataset with multiple IoU thresholds. *Note that H3DNet [41] only provide the checkpoint with 4 PointNet++ backbones as we use here.

ScanNet V2	mAP@0.25	mAP@0.50	mAP@0.75
VoteNet [20]	58.6	33.5	3.4
HGNet [2]	61.3	34.4	-
MLCVNet [36]	64.7	42.1	7.4
H3DNet* [41]	**67.2**	48.1	15.4
Ours	66.1	**50.9**	**19.1**

Table S2. 3D object detection results on SUN RGB-D dataset with multiple IoU thresholds. *Note that H3DNet [41] only provide the checkpoint with 4 PointNet++ backbones as we use here. Also H3DNet use $40,000$ points for SUN RGB-D dataset as input, while others use $20,000$ points.

SUN RGB-D	mAP@0.25	mAP@0.50	mAP@0.75
VoteNet [20]	57.7	32.9	1.2
HGNet [2]	**61.6**	-	-
MLCVNet [36]	59.8	-	-
H3DNet* [41]	60.1	39.0	3.5
Ours	61.1	**43.7**	**5.3**

A.2. More Qualitative Results

We provide more qualitative comparisons between our method and the top-performing reference methods, such as VoteNet [20], MLCVNet [36] and H3DNet [41], on the ScanNet V2 and SUN RGB-D datasets, as shown in Fig. S2 and Fig. S3, respectively. Our method can generate high-quality and compact predicted bounding boxes compared with the other reference methods.

We also show two typical failure cases in Fig. S1. Our BRNet cannot avoid the existence of false positive predicted bounding boxes which appear on the hollow floor. Also, it is hard for our method to detect objects on the smooth wall, especially windows and pictures. We need to mention that these failure cases are also common, and hard for the reference methods. It is an interesting and significant future direction of our work to tackle these false positives when points are over sparse and increase the robustness when perceiving objects within the cluttered background.

A.3. Implementation Details

As mentioned in the main paper, the BRNet consists of four modules: (1) vote generation and clustering, (2) back-traced representative points generation, (3) seed points revisiting, and (4) proposal refinement and classification followed by 3D NMS. Here we elaborate the implementation details with respect to each module.

Vote generation and clustering. We follow the same network architecture and vote regression loss as in VoteNet [20].

Representative point generation. It has output sizes of 128, 128, $6 + 2 \times NH$ for the three MLP layers, where NH is the number of heading bins for estimating the orientations, 6 is the 6 distance offsets from vote point to object surface (front/back/left/right/up/down) in the canonical coordinate centered at the vote point. Then we sample 2 representative points on each skewed direction as the back-traced representative points, thus we have 12 representative points per proposal.

Seed point revisiting. We use the set abstraction module (SA module) to aggregate seed points features within 0.2m radius surrounding a back-traced representative point. The SA module has the output size of 128, 64, 32 for the MLP layers. After revisiting seed points, we get a 32 di-

Table S3. 3D object detection results on ScanNet V2 validation dataset. Evaluation metric is the average precision with 3D IOU threshold 0.25. *Note that H3DNet [41] only reports the per-category results with 4 PointNet++ backbones, while others with only 1 PointNet++ backbone.

ScanNet V2	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP@0.25
VoteNet[20]	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	58.7
MLCVNet[36]	44.6	**89.6**	91.4	87.2	67.1	56.8	45.9	**59.5**	15.1	56.7	74.3	53.4	54.7	73.1	97.8	55.6	91.3	50.9	64.7
H3DNet*[41]	**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	49.3	88.3	**91.9**	86.9	**69.3**	59.2	45.9	52.1	15.3	**72.0**	**76.8**	57.1	**60.4**	73.6	93.8	58.8	92.2	47.1	66.1

Table S4. 3D object detection results on SUN RGB-D val dataset. We show per-category results of average precision (AP) with 3D IOU threshold 0.25 as proposed in [31], and mean of AP across all semantic classes. For fair comparison with previous methods, the evaluation is on the SUN RGB-D V1 data. *Note that H3DNet [41] sub-samples $40,000$ points from every scene in SUN RGB-D dataset, while others use $20,000$ points. Also, H3DNet [41] only reports the per-category results with 4 PointNet++ backbones, while others with only 1 PointNet++ backbone.

SUN RGB-D	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP@0.25
DSS[33]	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG[27]	58.3	63.7	31.8	62.2	**45.2**	15.5	27.4	51.0	51.3	70.1	47.6
2D-driven[11]	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
F-PointNet[21]	43.3	81.1	33.3	64.2	24.7	32.0	58.1	61.1	51.1	90.9	54.0
VoteNet[20]	74.4	83.0	28.8	75.3	22.0	29.8	62.2	64.0	47.3	90.1	57.7
MLCVNet[36]	**79.2**	85.8	31.9	75.8	26.5	31.3	61.5	66.3	50.4	89.1	59.8
H3DNet*[41]	73.8	85.6	31.0	76.7	29.6	33.4	65.5	**66.5**	50.8	88.2	60.1
HGNet[2]	78.0	84.5	**35.7**	75.2	34.3	**37.6**	61.7	65.7	51.6	91.1	**61.6**
Ours	76.2	**86.9**	29.7	**77.4**	29.6	35.9	**65.9**	66.4	**51.8**	**91.3**	61.1

mensional feature vector for each representative point. We concatenate the representative point features in a predefined local-structure-aware order to a $32 \times 12 = 384$ dimensional feature vector per proposal. The feature vector is then projected to 128-dimensional as the captured surface feature of the object proposal.

Proposal refinement and classification. The input is the 256 dimensional fused feature vector which is the concatenation of 128-D vote cluster feature and 128-D revisited seed point feature. Then the fused feature is fed into a three-layer MLP, whose output sizes are 128, 128, $9 + N_C$. N_C is the number of semantic classes, *i.e.*, $N_C = 10$ for the SUN RGB-D dataset [31] and $N_C = 18$ for ScanNet V2 dataset [4]. In the first 9 channels, the first two are for objectness classification, the following one is for heading angle refinement and the last six are for distance offsets refinement.

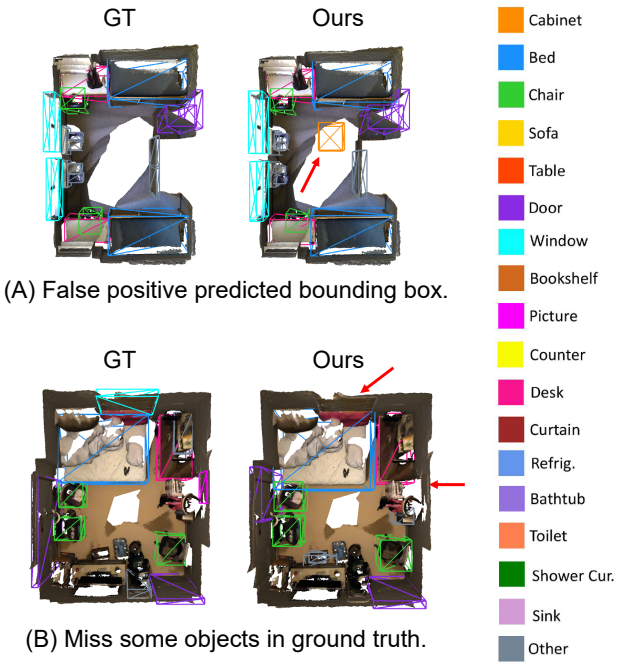


Figure S1. Samples of failure cases on ScanNet V2 dataset.

Table S5. 3D object detection results on SUN RGB-D val dataset. We show per-category results of average precision (AP) with 3D IOU threshold 0.50 as proposed in [31], and mean of AP across all semantic classes. For fair comparison with previous methods, the evaluation is on the SUN RGB-D V1 data. *Note that H3DNet [41] sub-samples $40,000$ points from every scene in SUN RGB-D dataset, while others use $20,000$ points. Also, H3DNet [41] only reports the per-category results with 4 PointNet++ backbones, while others with only 1 PointNet++ backbone.

SUN RGB-D	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP@0.50
VoteNet[20]	49.9	47.3	4.6	54.1	5.2	13.6	35.0	41.4	19.7	58.6	32.9
H3DNet*[41]	47.6	52.9	8.6	60.1	8.4	20.6	45.6	50.4	27.1	69.1	39.0
Ours	**55.5**	**63.8**	**9.3**	**61.6**	**10.0**	**27.3**	**53.2**	**56.7**	**28.6**	**70.9**	**43.7**

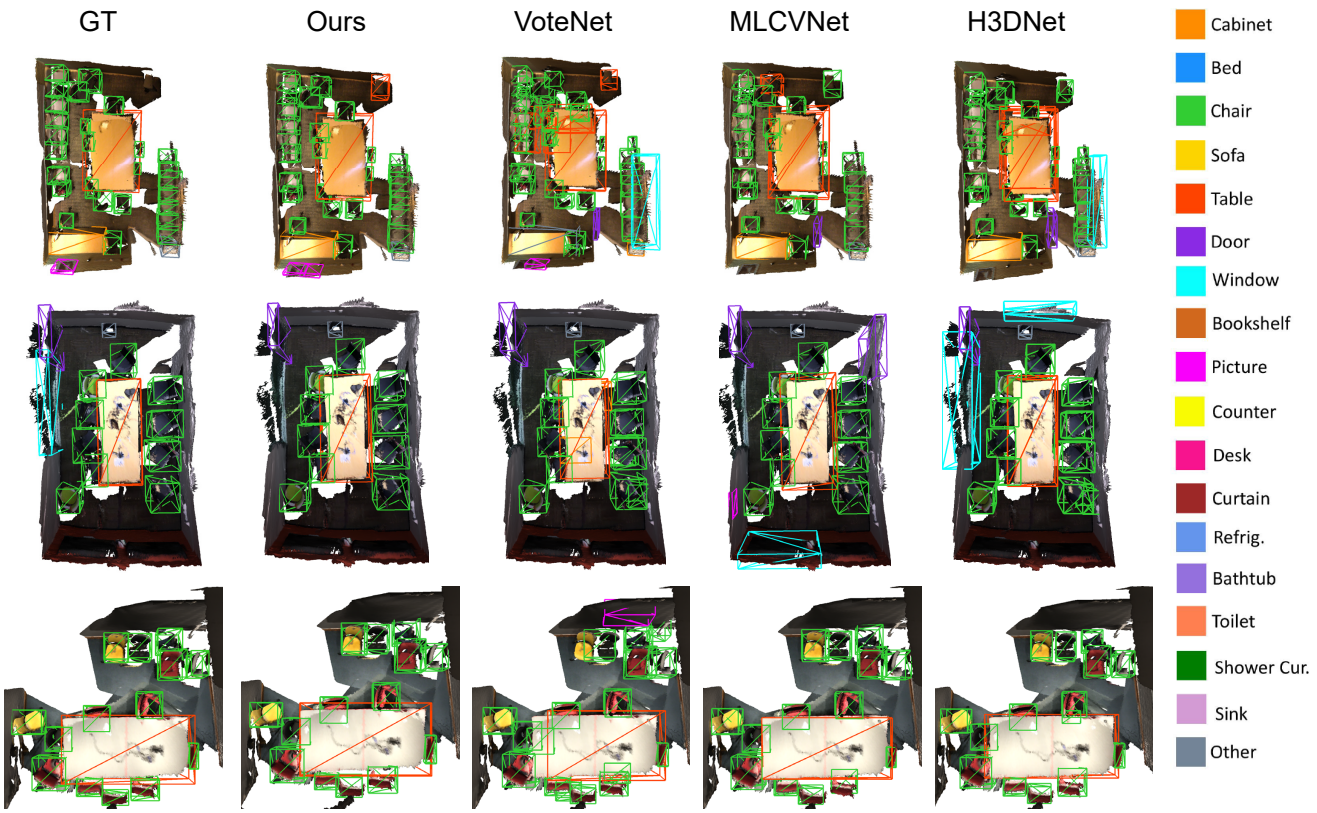


Figure S2. More qualitative results on ScanNet V2 dataset [4]. The reference methods are VoteNet [20], MLCVNet [36] and H3DNet [41]. Best viewed on screen.

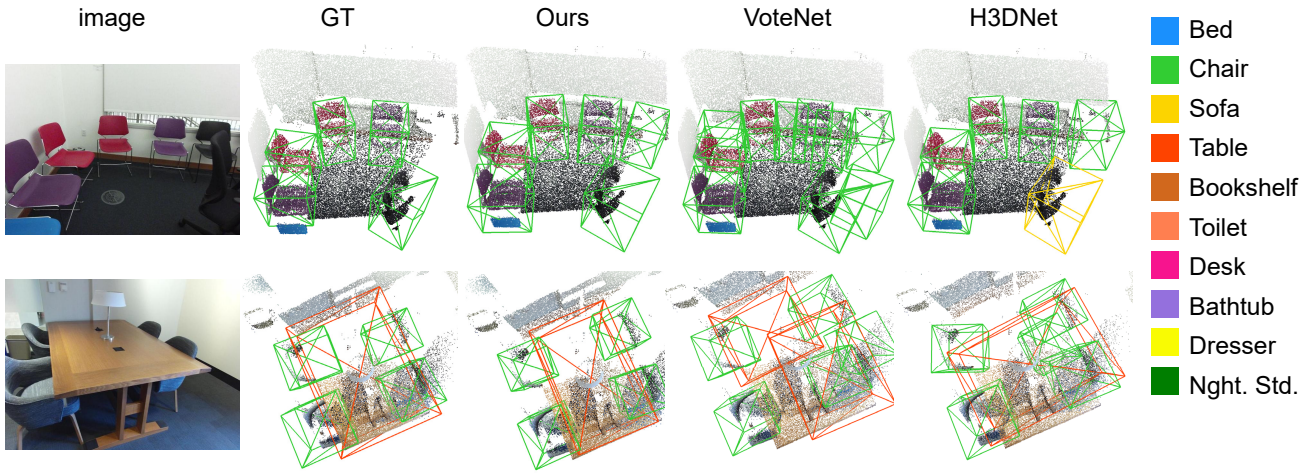


Figure S3. More qualitative results on SUN RGB-D dataset [31]. The reference methods are VoteNet [20] and H3DNet [41]. Best viewed on screen.