VENet: Voting Enhancement Network for 3D Object Detection

Qian Xie[1], Yu-Kun Lai[2], Jing Wu[2], Zhoutao Wang[1], Dening Lu[1], Mingqiang Wei[1], and Jun Wang *[1]

[1]Nanjing University of Aeronautics and Astronautics
[2]Cardiff University

Abstract

*Hough voting, as has been demonstrated in VoteNet, is effective for 3D object detection, where voting is a key step. In this paper, we propose a novel VoteNet-based 3D detector with vote enhancement to improve the detection accuracy in cluttered indoor scenes. It addresses the limitations of current voting schemes, i.e., votes from neighboring objects and background have significant negative impacts. **Before voting**, we replace the classic MLP with the proposed Attentive MLP (AMLP) in the backbone network to get better feature description of seed points. **During voting**, we design a new vote attraction loss (VALoss) to enforce vote centers to locate closely and compactly to the corresponding object centers. **After voting**, we then devise a vote weighting module to integrate the foreground/background prediction into the vote aggregation process to enhance the capability of the original VoteNet to handle noise from background voting. The three proposed strategies all contribute to more effective voting and improved performance, resulting in a novel 3D object detector, termed VENet. Experiments show that our method outperforms state-of-the-art methods on benchmark datasets. Ablation studies demonstrate the effectiveness of the proposed components.*

1. Introduction

3D object detection is an active research topic in computer vision with a wide range of applications, such as autonomous driving [29], robotic manipulation [39] and high-level semantic SLAM (Simultaneous Localization and Mapping) [45]. However, locating and classifying objects from scanned 3D point clouds in cluttered indoor scenes is still a challenging problem, without color information in particular. Although many efforts have been made to improve its performance over past few years [44, 32, 2, 27, 42], driven by the success of deep learning techniques, the performance is still far from being satisfactory.

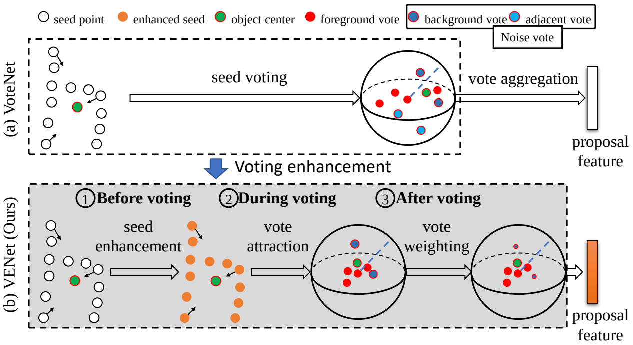
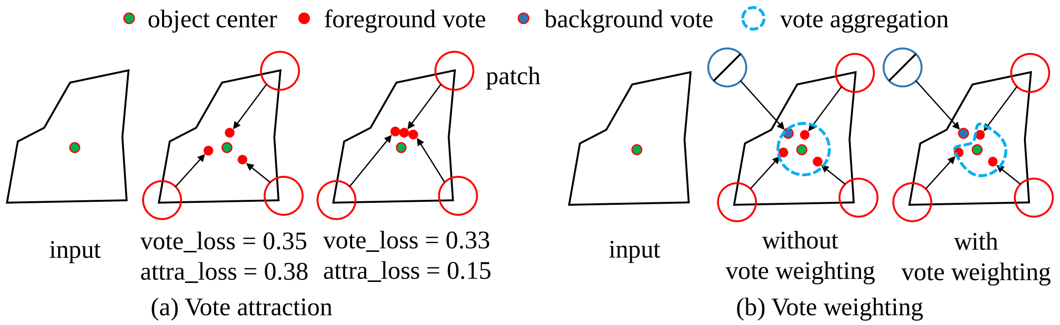


Figure 1. Comparison with VoteNet. (a) VoteNet. (b) Our approach. Our approach enhances the voting procedure from three aspects (i.e., seed enhancement, vote attraction and vote weighting) to get better proposal features.

Recently, a deep Hough voting network, VoteNet [28], was proposed to detect 3D objects directly from scanned point clouds, and has achieved significant improvements on several benchmark datasets. This method first samples seed points from the whole point cloud, and then extracts high-dimensional features of these seed points using PointNet++ [30]. Then, inspired by Hough voting in 2D object detection, these seed points produce vote centers based on the extracted features. The voting process is formulated as center point regression and implemented via MLP (Multi-Layer Perceptron). These votes are then clustered and aggregated to generate object proposal features, which are used to classify objects and regress their locations. Voting, as the essence in VoteNet, plays a vital role in information aggregation for object detection.

However, there are two disturbing factors in voting using the current VoteNet architecture, i.e., *object-noise*: votes from adjacent objects, and *background-noise*: votes from background seed points. As shown in Figure 1, VoteNet will choose a vote as the cluster center, and then aggregate information with no difference from all the votes within the bounding sphere to form the aggregated feature for the cen-

*Corresponding author: wjun@nuaa.edu.cn



(a) Vote attraction

(b) Vote weighting

Figure 2. Motivation of the proposed (a) vote attraction loss and (b) vote weighting. (a) The original vote loss could make votes locate loosely around the object center, while our attraction loss increases the compactness of these votes. (b) Foreground prediction can make the detector re-weight votes during vote aggregation to suppress the impact of background noisy votes.

ter. However, since the objects in indoor scenes are highly cluttered and close to each other, this simple clustering strategy may include votes from other adjacent objects. Moreover, as VoteNet does not apply any constraints or penalty to votes from background seed points, these background votes may also be included in the subsequent vote aggregation. In addition to the two disturbing factors, the voting effectiveness also highly depends on the seed point features. We argue that the classic MLP features, which merely depend on the last layer of MLP, lack information from the former layers, leading to loss of useful information.

As a solution, in this work we propose VENet (Voting Enhancement Network), a 3D object detector based on VoteNet. VENet improves the voting procedure in all the three stages (i.e., before, during and after), by enhancing feature description of seed points and handling noisy votes from both adjacent objects and background patches. Specifically, **before voting**, we first propose an Attentive MLP (AMLP) to enhance seed point feature description by adaptively considering multi-layer information in classic MLP. Then, **during voting**, to relieve the negative impact of votes from adjacent objects, we expect the votes not only to be close to their ground truth centers, but also to be close to each other if they belong to the same object, as illustrated in Figure 2(a). We thus design a novel loss function for seed point voting, called *vote attraction loss*, to decrease the internal distances between votes associated with the same object centers. The increased compactness reduces the possibility of gathering information from adjacent objects, i.e., the *object-noise*. Lastly, **after voting**, to reduce the impact of meaningless and misleading votes from background seed points, we propose to predict foreground probability of seed points, and weight their votes accordingly during aggregation. That is, we expect votes from seed points with higher foreground probabilities to contribute more during vote aggregation. As illustrated in Figure 2(b), this strategy can suppress the negative impact of background votes, i.e., the *background-noise*.

The contributions of the work can be summarized as:

- We propose a voting enhancement architecture to improve the voting procedure for Hough voting-based 3D object detection from point clouds, which obtains new state-of-the-art performance on public datasets.
- Before voting, we introduce an AMLP (Attentive MLP) to enhance the feature encoding of seed points.
- During voting, we design a vote attraction loss (VA-Loss) to enforce votes to locate compactly and closely to the corresponding object centers.
- After voting, we present a vote weighting module to integrate foreground seed point prediction into the vote aggregation to reduce background noise.

2. Related Work

Many efforts have been made to automatically detect 3D objects in both indoor and outdoor scenes [38, 15, 12, 26, 47, 32, 46, 22], which can be divided into 3 categories based on the input modalities: 2D, 2D-3D and 3D.

For outdoor scenes, merely taking 2D images as input, GS3D [17] proposed a purely monocular approach for obtaining coarse cuboid boxes for the objects resulted from reliable 2D detection. M3D-RPN [1] and some other works [40, 36] were also proposed for 3D object detection from monocular 2D images. In multi-sensor processing, i.e., 2D-3D, [4] and [14] extracted features from LiDAR bird-view and camera images, and projected 3D proposals to the corresponding 2D feature maps for the task of 3D object detection. ContFuse [20] further introduced a continuous fusion layer to perform feature fusion for camera image and LiDAR bird-view feature combination. LaserNet++ [23] fused image data with LiDAR data, and expanded object detection to 3D semantic segmentation.

Fusing 2D-3D features heavily relies on 2D detectors. Instead, some works [50, 5, 24, 6, 34] have been proposed to process 3D point data independently. VoxelNet [50] unified feature extraction and bounding box prediction into a single-stage, end-to-end trainable deep network, which removed the need of manual feature engineering for LiDAR point clouds. Likewise, PointPillars [16] used a grid-based feature description with a feature pyramid network. The whole input point cloud is divided into pillars, whose features are combined with anchors to perform joint regression and classification. Instead of projecting a point cloud to voxels, PointRCNN [33] directly generated 3D proposals from point clouds, and then introduced further refinement for proposals. Fast PointRCNN [5] utilized both a voxel representation and the raw point cloud data to exploit respective advantages for 3D object detection. LaserNet [24] used a fully convolutional network to predict a multi-modal distribution for each point and then fused these distributions to generate a prediction for each object.

For indoor scenes, works in [21], [29] and [31] integrated both 2D and 3D, and both object and scene context information for indoor 3D object detection from RGB-D data. In addition, PointFusion [43] introduced a novel framework, in which the image data and the raw point cloud data are independently processed by a CNN (Convolutional Neural Network) and a PointNet architecture respectively, followed by a fusion network combining their output results. Instead of utilizing both 2D and 3D information, [35] took 3D point data only, and utilized the geometric and hierarchical contextual information for 3D object detection. Recently, with only 3D input, VoteNet [28] introduced a deep learning-based Hough voting strategy for 3D object detection from point clouds. These methods locally select a set of seed points to generate votes and then combine these votes to generate object proposals. Further, ImVoteNet [27] was built on top of VoteNet and proposed a 3D detection architecture specialized for single-view RGB-D scenes, which fused 2D votes in images and 3D votes in point clouds. However, this method may be sensitive to lighting conditions by using image information. Moreover, both works [28, 27] ignored negative impact from other adjacent objects and background seed points in the voting stage. As a result, the subsequent vote aggregation could include noisy votes, which affect the final object detection results. In this work, we target a more effective voting strategy to enhance vote aggregation and tackle these issues by vote attraction and foreground weighting, using geometry information alone.

3. Method

Our VENet inherits the deep Hough voting network (VoteNet) [28] for indoor scene object detection, and improved it with the proposed AMLP (Section 3.1), vote attraction loss (Section 3.2), and the vote weighting module (Section 3.3).

The original VoteNet [28] can be summarized into three modules, i.e., voting module, vote aggregation module and object proposal module. The voting module is to regress object centers from each of the seed points, and the vote aggregation module is to combine features from different seed points to vote for the object centers. The object proposal module then classifies and regresses the accurate locations and sizes of 3D objects from the aggregated features.

Let $s_i = [x_i; f_i]$ be a seed point, where $x_i \in \mathbb{R}^3$ and $f_i \in \mathbb{R}^C$ are the coordinates and extracted features respectively. According to the set abstraction mechanism in PointNet++ [30], f_i encodes the information of the seed point s_i and its surrounding points. In the voting module, VoteNet uses an MLP layer to simulate the voting procedure via regressing the offset $\Delta x_i \in \mathbb{R}^3$, from which the predicted object center y_i is obtained by adding the offset, i.e., $y_i = x_i + \Delta x_i$. A vote regression loss $L_{\text{vote-reg}}$ is defined to supervise the predicted object centers to approach the ground truth ones.

$$L_{\text{vote-reg}} = \frac{1}{|S_{\text{pos}}|} \sum_i \|\Delta x_i - \Delta x_i^*\| \, 1 \, [s_i \text{ on object }] \tag{1}$$

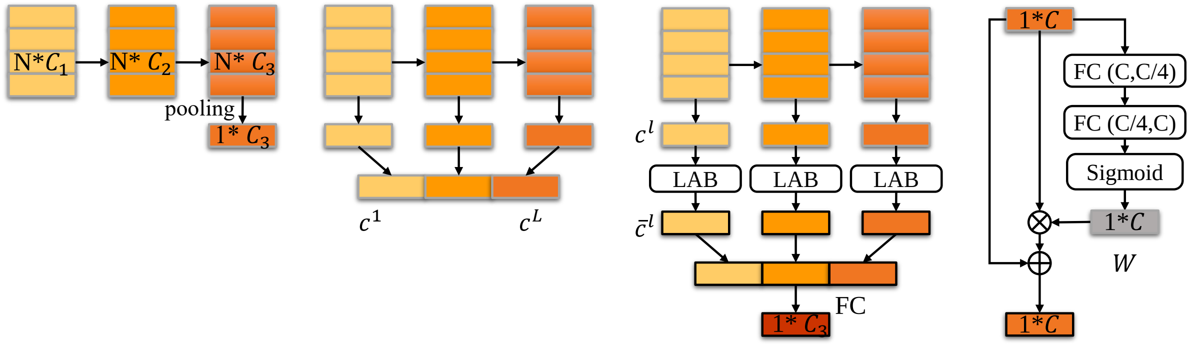
where Δx_i^* is the ground truth offset, $1 \, [s_i \text{ on object }]$ indicates whether a seed point s_i is on an object surface, and S_{pos} is the set of all the positive seeds, i.e., those on the object surfaces.

In Equation 1, due to the use of the indicator function, seeds on background were discarded during training. However, during testing, there are no constraints or guidance applied to background seeds (i.e., those not on object surfaces) to restrain their voting. On the other hand, for votes from the foreground (object) seed points, the regression loss in Equation 1 enforces the closeness of the predicted centers to their ground truth ones, but not the 'compactness' between those belonging to the same objects, which may result in some predicted centers adversely affecting the aggregation for other objects. Therefore, the above vote regression loss is unable to handle noisy votes. Moreover, the seed point feature extraction using PointNet++ is through the classic MLP layers which lack the information from the former layers. As a result, the extracted features are not informative enough to support effective voting.

3.1. Attentive MLP

We first introduce an improved MLP, termed Attentive MLP (AMLP), which is integrated into the backbone of PointNet++ to get better feature description of seed points.

In VoteNet, the feature description of each seed point is obtained by simply pooling the feature vectors of its neighboring points at the last layer, which can be seen as the classic MLP, as shown in Figure 3(a). However, as indicated in [13], this simple pooling operation does not take into consideration of low- and mid-level features which contain rich local information. PF-Net [13] addresses this by designing a



(a) Classic MLP (b) CMLP (c) AMLP (d) LAB

Figure 3. Comparison of different MLP-based feature extraction architectures. (a) Classic MLP gets the pooled feature vector in the last level; (b) CMLP combines pooled features from multiple levels; (c) the proposed AMLP first assigns different weights to pooled features according to their importance, and then adaptively combines them. (d) Level Attention Block (LAB). In this illustration, we assume the number of levels $L = 3$.

Combined-MLP (CMLP). As shown in Figure 3(b), CMLP pools the feature vectors at multiple MLP layers, and then concatenates the pooled features. It improves the performance of shape classification with the combined features, as demonstrated in their experiments. However, we argue that the combination of multi-layer features could be more adaptive. Therefore, we propose an Attentive MLP (AMLP), which adaptively combines multi-layer features by learning the weights for the pooled features before concatenation, as shown in Figure 3(c). That is, our AMLP introduces adaptive weights to better measure the importance of features at different layers inside MLP.

Specifically, for each point p, we first pool features from each of the layers, generating $(c^1, \cdots, c^L)$. L is the layer number of perceptrons in MLP. Then instead of directly concatenating the pooled features like PF-Net, we insert a Level Attention Block (LAB) at each layer, as shown in Figure 3(d). In each LAB, a pooled feature vector c^l is first fed into two FC (fully connected) layers with output sizes of $C/4$ and C. ReLU is used as the activation function for the first FC layer. Sigmoid function is used to normalize the output weights to be in the range of $(0, 1)$. c^l is then multiplied by the learned weights W^l and added to itself, i.e.,

$$\mathbf{c}^l = c^l + W^l * c^l \tag{2}$$

where $\mathbf{c}^l$ is the enhanced feature vector. The enhanced feature vectors from all layers are then concatenated to form the combined feature vector which then goes through a further FC layer to output the feature description of the desired size (the same size as the PointNet++ output).

$$\mathbf{C} = FC(Concat(\mathbf{c}^1, \cdots, \mathbf{c}^l)) \tag{3}$$

In this way, AMLP enhances the feature descriptions of seed points.

3.2. Attraction Loss

To reduce the number of false votes from adjacent objects in vote aggregation, we should not only require votes to be close to their ground truth object centers, but also enforce votes to locate compactly with each other when they are from the same object. To this end, we propose a new vote attraction loss (VALoss) for better voting supervision, which tries to minimize the internal distances between votes associated with the same object centers, as illustrated in Figure 4(a). In other words, the VALoss is designed to consider the attractiveness between votes. Specifically, we use the ℓ_1 loss to measure the distance between the vote y_{ij} and the average center $\overline{y}_i$ of object i, and design the VALoss as:

$$L_{\text{vote}-\text{attr}} = \frac{1}{|B_{\text{gt}}|} \sum_i \left(\frac{1}{|Y^i_{\text{vote}}|} \sum_j \|y_{ij} - \overline{y}_i\| \right), \tag{4}$$

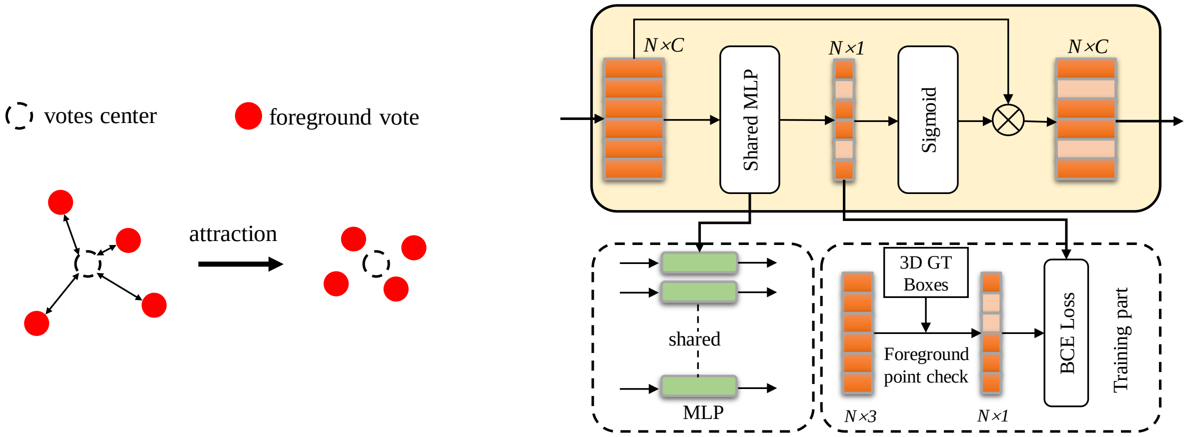
where B_{gt} is the set of ground truth boxes (each box corresponds to an object), and Y^i_{vote} is the set of votes associated with the i-th ground truth box. $i \in \{1, ..., |B_{\text{gt}}|\}$ is the index of ground truth boxes, and $j \in \{1, ..., |Y^i_{\text{vote}}|\}$ is the index of votes. Hence, y_{ij} represents the j-th vote of the i-th ground truth box. $\overline{y}_i$ is the average center of all the votes associated with the i-th ground truth box, which is calculated as:

$$\overline{y}_i = \frac{1}{|Y^i_{\text{vote}}|} \sum_j y_{ij} \tag{5}$$

The intuition behind the above equations is that good votes from the same object should all be close to their mean center, i.e., locate compactly with each other. Finally, the new vote loss is:

$$L_{vote} = L_{vote-reg} + \alpha L_{vote-attr} \tag{6}$$

where $L_{vote-reg}$ is defined in Equation 1, α is the hyperparameter to balance the two loss terms, which is set to 0.5



(a) Vote attraction

(b) Vote weighting module

Figure 4. (a) Illustration of the vote attraction mechanism. Our vote attraction loss tries to minimize the distance between votes and the center of these votes, i.e., to make votes locate compactly with each other. Thus, it can reduce the possibility of including votes from adjacent objects when performing vote aggregation. Note that the vote center is not the same as the real object center. (b) Architecture of the vote weighting module. The feature maps are of size $N \times C$ where N is the number of seed points and C represents the feature dimension. BCE: Binary Cross Entropy.

in our experiments. The new vote loss incorporates both the regression term and the attraction term, which is a multi-task loss pushing votes towards the corresponding ground truth object centers, while minimizing the internal distances between votes associated with the same object centers.

3.3. Vote Weighting for Background Suppression

After voting, these votes will be further clustered and aggregated to generate proposal features. In the original VoteNet, votes within clusters are treated without difference, regardless of whether they come from foreground or background seed points. Intuitively, only votes from the foreground seed points should contribute to the proposals, while the ones from the background seed points should be discarded during aggregation. However, as mentioned, the current VoteNet architecture cannot suppress the voting from background seed points during testing.

As a solution, we design a new vote weighting module, which assigns different aggregation weights to votes according to their seed points' foreground probabilities. Specifically, as illustrated in Figure 4(b), we first use a shared MLP with three layers to predict a score for each seed point, which reflects its possibility of belonging to foreground. The prediction is trained with the ground truth foreground/background labels as supervision, which are obtained by checking seed points' status of inside/outside ground-truth 3D boxes. The vote features are then enhanced by re-weighting the original vote features using the predicted scores. Formally, given the vote feature f_i, the re-weighted vote feature $\widetilde{f}_i$ is formulated as:

$$\widetilde{f}_i = \delta(f_i) \otimes f_i \tag{7}$$

where $\delta(\cdot) = sigmoid(MLP(\cdot))$ is the transform function to predict a foreground confidence between 0 and 1, and $\otimes$ is element-wise multiplication. The proposed weighting scheme allows the detector to focus on votes more likely to be from foreground regions (large weights), and neglect votes from background (small weights) before aggregation for object proposal.

4. Experiments

4.1. Experimental Setup

The proposed 3D detector follows the architecture of deep Hough voting network [28]. To generate foreground/background labels for sample points, we regard all the points within labeled 3D bounding boxes as foreground points, and the points outside all boxes as background points. We optimize the network using the Adam algorithm, which is trained on an RTX 2080Ti GPU with batch size of 8. We set the initial learning rate to be 0.01, and decay it by 0.1 at the steps of (120,140,180). We train the network from scratch with 200 epochs in total. Due to several sub-sampling and other random operations, there is a small variance with the evaluated mAP results upon convergence (after around 140 epochs). Thus, the mAP results reported in the paper are the mean results over training the model for 3 times, in order to reduce the effect of randomness.

4.2. Comparison

Datasets. We evaluate the performance of the proposed VENet on two datasets of indoor scenes: ScanNet dataset [7] and SUN RGB-D dataset [37]. ScanNet dataset is a richly annotated dataset of 3D meshes. 3D scenes in

Method	Conference	mAP@0.25	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	refrigerator	showercurtrain	toilet	sink	bathtub	ofurn
Feng [9]	arXiv2019	48.5	31.08	83.1	85.86	77.5	56.27	30.55	25.1	34.84	4.09	38.5	59.11	35.32	33.7	46.29	88.6	40.27	82.0	20.9
Griffiths [10]	ECCV2020	50.2	43.0	70.8	58.3	16.0	44.6	28.0	13.4	58.2	4.9	69.9	74.0	75.0	36.0	58.9	79.0	47.0	77.9	48.2
VoteNet [28]	ICCV2019	58.65	36.27	87.92	88.71	89.62	58.77	47.32	38.1	44.62	7.83	56.13	71.69	47.23	45.37	57.13	94.94	54.7	92.11	37.2
GRNet [19]	ISPRS2020	59.14	39.45	**88.78**	89.18	88.34	58.16	48.46	32.7	46.97	4.94	63.48	69.81	48.46	49.06	66.37	94.07	49.7	90.9	35.6
SPOT [8]	ECCV2020	59.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HGNet [3]	CVPR2020	61.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SESS [49]	CVPR2020	62.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GSDN [11]	ECCV2020	62.84	41.58	82.5	92.14	86.95	61.05	42.41	40.66	51.14	10.23	64.18	71.06	54.92	40.0	70.54	**99.97**	**75.5**	**93.23**	53.07
DOPS [25]	CVPR2020	63.7	53.2	83.3	91.6	82.6	60.5	54.8	45.2	41.0	**26.3**	51.9	73.7	53.9	49.2	64.7	98.0	71.3	86.6	**59.2**
LGR-Net [18]	arXiv2020	64.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MLCVNet [41]	CVPR2020	64.5	42.45	88.48	88.98	87.4	63.5	56.93	46.98	**56.94**	11.94	**63.94**	76.05	63.94	60.86	65.91	98.33	59.18	87.22	47.89
H3DNet [48]	ECCV2020	67.2	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
VENet(Ours)		**67.7**	**50.4**	87.7	**92.7**	88.1	**68.6**	60.7	46.0	55.2	18.2	**70.2**	**77.5**	59.9	58.4	**75.9**	95.1	67.2	92.3	54.4

Table 1. Performance comparison on ScanNetV2 Val set.

	mAP@0.25	mAP@0.5
VoteNet [28]	57.7	32.9
H3DNet [48]	60.1	39.0
LGR-Net [18]	62.2	-
HGNet [3]	61.6	-
SPOT [8]	60.4	36.3
Feng [9]	59.2	-
MLCVNet [41]	59.2	-
VENet(Ours)	**62.5**	**39.2**

Table 2. Performance comparison on SUN RGB-D validation set.

	Training time (s)	Inference time (s)	# Params (million)	mAP @0.25
H3DNet [48]	420	0.70	4.7	67.2
VENet(Ours)	**85**	**0.32**	**2.8**	**67.7**

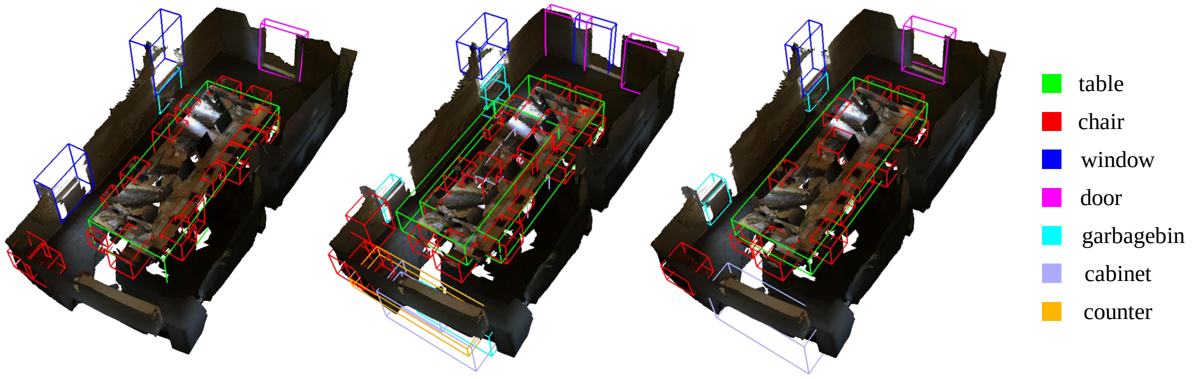
Table 3. Performance comparison with the previous state-of-the-art method, H3DNet [48] on ScanNet dataset.

this dataset are all captured in indoor scenes by portable RGB-D sensors. Note that our method does not require RGB information, and directly works on 3D point clouds. The dataset contains 1,513 scanned indoor scenes with 3D bounding boxes annotated. It is split into two sets, Train and Val containing 1,201 and 312 scenes respectively. Results in this paper are all evaluated on the Val set, as VoteNet does. SUN RGB-D dataset contains 10,335 scenes captured by RGB-D sensors from a single view, with 5,285 for training and 5,050 for validation. Each scene is converted into a 3D point cloud representation with annotated indoor objects.

Quantitative comparison. Table 1 shows the results on ScanNet dataset using different 3D object detection methods. As shown, the proposed VENet outperforms its baseline VoteNet by 9.0% and achieves the new state-of-the-art performance in the mAP@0.25 evaluation. Moreover, VENet achieves the best results in 6 out of the 18 classes, which doubles that of the second ranked H3DNet [48] which has best results only in 3 classes. This demonstrates that the proposed vote enhancement strategies can effectively improve the subsequent object localization and classification tasks. Table 2 shows results on SUN RGB-D dataset. For a fair comparison, we only compare the results from methods using 3D geometric information only. As shown, the proposed VENet again achieves the state-of-the-art performance on SUN RGB-D dataset with 62.5% mAP@0.25. The overall improvement is not as significant as on ScanNet. We think it is because most scenes in SUN RGB-D cover smaller areas and have fewer objects (as seen in Figures 5 and 6), making noisy votes a less prominent problem in SUN RGB-D.

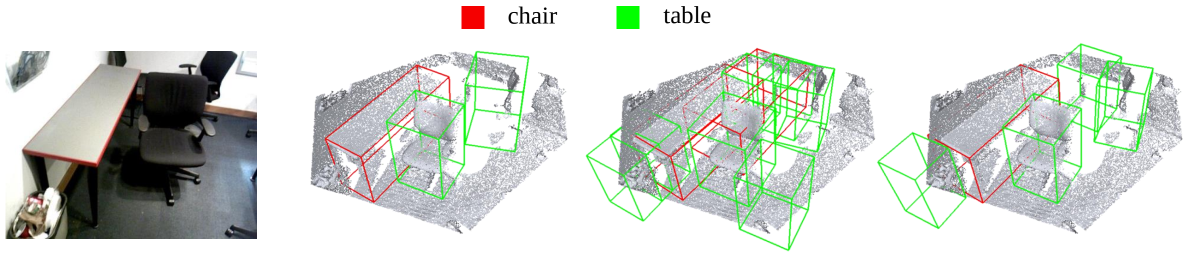
Speed and model size. The most recent H3DNet [48] has the second best performance in terms of mAP. Both of H3DNet and our VENet are developed based on VoteNet. However, we notice the difference in terms of training/inference times and model size. As shown in Table 3, the number of trainable parameters of our network is 2.8 million, while H3DNet is 4.7 million. This indicates the network architecture of our VENet is simpler and has much fewer parameters. For training time, H3DNet takes around $420s$ for one epoch, while VENet takes much less time, $85s$. For inference time, we measure the time for one scene in ScanNet dataset. As shown, H3DNet takes $0.70s$, while ours is $0.32s$. Our model is more than $2\times$ faster than H3DNet. We reckon it is because voting in H3DNet happens three times for object, face and edge centers. These quantitative results demonstrate that our VENet is not only more effective but also more efficient than H3DNet. Moreover, H3DNet has the assumption that objects should have obvious structures of faces and edges, while VENet has no such assumption and is thus more suitable for general object detection.

Qualitative comparison. Figure 5 and Figure 6 visualize the detection results using VoteNet and the proposed VENet. We observe that VENet can obtain better detection results with less false positives and more accurate bounding boxes than the original VoteNet on both ScanNet and SUN RGB-D datasets. From Figure 5, we can see that two over-



(a) Ground truth (b) VoteNet (c) Ours

Figure 5. Qualitative comparison results of 3D object detection on ScanNetV2. As shown, our voting enhancement strategies enable more accurate object classification and localization. *Note that color is only used for better visualization, and not utilized in our method.*



(a) Image (b) Ground truth (c) VoteNet (d) Ours

Figure 6. Qualitative comparison results of 3D object detection on SUN RGB-D.

Method	AMLP	VALoss	VW	SUN RGB-D	ScanNet
Baseline				57.8	59.6
VENet	✓			59.1	62.3
VENet	✓	✓		61.6	64.8
VENet	✓	✓	✓	62.5	67.7

Table 4. Ablation study on the test dataset ($mAP@0.25$). VW: vote weighting. Baseline is trained and evaluated by ourselves.

lapping boxes of the table are detected using VoteNet, while VENet accurately detects the single bounding box of the table. Figure 6 shows that the proposed VENet gives better detection results with less overlaps compared to VoteNet. This suggests that the improved compactness of votes reduces the sparse distribution of object centers, which contributes to the reduced detection of overlapping boxes.

4.3. Ablation Study

To analyze the importance of the three proposed strategies, we conduct several experiments using different combinations of the proposed components on both ScanNet and SUN RGB-D datasets. We use the original VoteNet as our baseline model and we train VoteNet from scratch to get the results using the evaluation strategy in Section 4.1. The results are presented in Table 4. The second row shows that the AMLP improves the performance significantly from 59.6% to 62.3% on ScanNet. Adding VALoss further improves the result to 64.8%, demonstrating its effectiveness to reduce *object-noise* in voting. The additional improvement to 67.7% with the vote weighting module further demonstrates the module's effectiveness to suppress background-noise in vote aggregation. The best mAP results are achieved with all the proposed components equipped, both for ScanNet and SUN RGB-D datasets.

	Classic MLP	CMLP	AMLP
mAP@0.25	59.6	61.1	**62.3**

Table 5. Performance comparison with CMLP and classic MLP.

To demonstrate the effectiveness of AMLP, we independently compare the detection performances of AMLP, CMLP [13], and classic MLP on ScanNet dataset. We replace classic MLP in VoteNet with AMLP and CMLP. Results are shown in Table 5. The proposed AMLP achieves the best performance, which indicates that AMLP has a better feature extraction ability.

To illustrate the positive guidance of our attraction loss, we visualize the voting results in Figure 7. As seen in the green boxes, VALoss can effectively enforce the votes associated with the same object center to locate more compactly with each other, which helps to reduce noisy information from other objects and thus improves the performance.

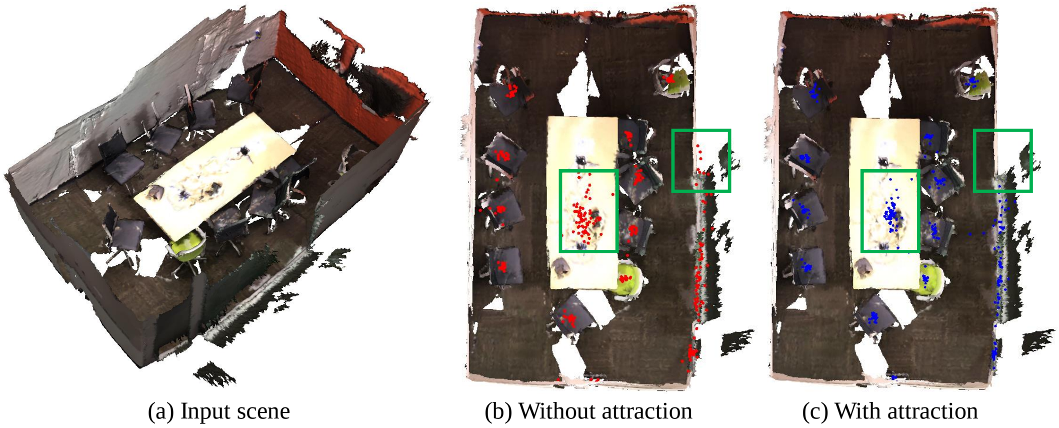
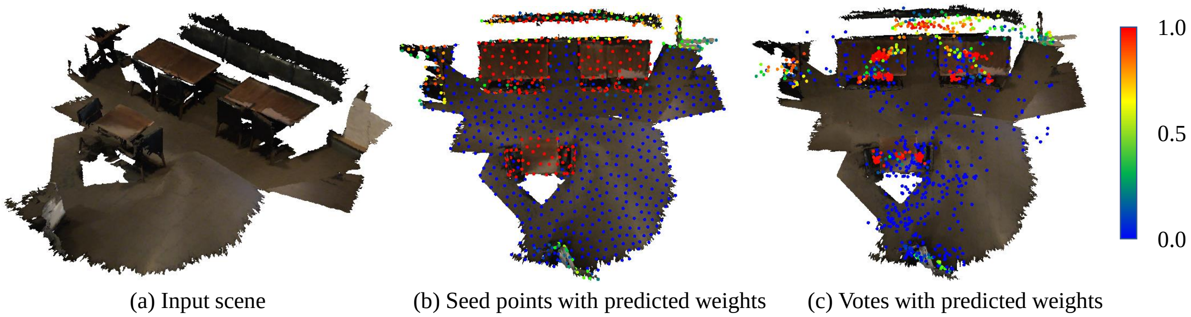

(a) Input scene (b) Without attraction (c) With attraction

Figure 7. Voting comparison with our VALoss. (c) Votes (blue points) are located more compactly with the proposed loss, compared with red points in (b), obtained without attraction loss.



(a) Input scene (b) Seed points with predicted weights (c) Votes with predicted weights

Figure 8. Vote weighting result. The predicted weights between 0-1 are mapped to blue-red according to the color bar. As can be seen, the votes with high predicted weights (c) are almost coming from foreground seed points (b).

To verify the effectiveness of the proposed vote weighting module, we further visualize the predicted weights in Figure 8. We observe that the predicted weights are almost consistent with their foreground/background labels, as observed from seed points with weights in Figure 8(b). Also as observed in Figure 8(c), votes with high weights are closer to object centers than those with low weights. That is, votes from object seed points have higher contributions to feature aggregation, which is as expected.

5. Conclusion

In this paper, we propose a novel 3D object detector, VENet, with enhanced feature description and vote aggregation based on VoteNet framework. Specifically, before voting, to enhance the feature description of seed points, we present an Attentive MLP (AMLP) to adaptively integrate multi-layer information in classic MLP. During voting, we design a vote attraction loss (VALoss) to relieve the negative impact of votes from seed points in adjacent objects, by enforcing the votes to be not only close to the corresponding object centers, but also compactly located with each other. Moreover, after voting, to reduce the meaningless votes from background seed points, we propose a vote weighting module to predict foreground probability for seed points, and use this information to achieve more effective vote aggregation. Our method achieves the state-of-the-art detection accuracy on the ScanNet and SUN RGB-D datasets with only geometric information given, demonstrating the effectiveness of the proposed approach. Although the focus of this paper is voting based 3D object detection for indoor scenes, our proposed techniques are generally applicable to other applications and methods using Hough voting.

In the immediate future work, we plan to explore a more effective sampling algorithm for vote aggregation. The farthest point sampling algorithm currently used equally samples votes from the whole set, which results in the majority of votes in the background. One potential solution is to give higher possibility to sample from foreground votes, which may reduce the number of false positives.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China (2019YFB1707504), National Natural Science Foundation of China (61772267, 62032011), and the Natural Science Foundation of Jiangsu Province under Grant BK20190016.

References

[1] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9287–9296, 2019. 2

[2] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. *ECCV*, 2020. 1

[3] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3D object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 392–401, 2020. 6

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[5] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9775–9784, 2019. 2, 3

[6] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12536–12545, 2020. 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 5

[8] Hongyuan Du, Linjun Li, Bo Liu, and Nuno Vasconcelos. SPOT: Selective point cloud voting for better proposal in point cloud object detection. In *ECCV*, 2020. 6

[9] Mingtao Feng, Syed Zulqarnain Gilani, Yaonan Wang, Liang Zhang, and Ajmal Mian. Relation graph network for 3D object detection in point clouds. *IEEE Transactions on Image Processing*, 30:92–107, 2020. 6

[10] David Griffiths, Jan Boehm, and Tobias Ritschel. Finding your (3D) center: 3D object detection using a learned loss. *ECCV*, 2020. 6

[11] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3D single-shot object detection. *ECCV*, 2020. 6

[12] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 2

[13] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point fractal network for 3D point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 3, 7

[14] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3D proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[15] Jean Lahoud and Bernard Ghanem. 2D-driven 3D object detection in RGB-D images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4622–4630, 2017. 2

[16] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 3

[17] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. GS3D: An efficient 3D object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 2

[18] Jianan Li and Jiashi Feng. Local grid rendering networks for 3D object detection in point clouds. *arXiv preprint arXiv:2007.02099*, 2020. 6

[19] Ying Li, Lingfei Ma, Weikai Tan, Chen Sun, Dongpu Cao, and Jonathan Li. GRNet: Geometric relation network for 3D object detection from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165:43–53, 2020. 6

[20] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 2

[21] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1417–1424, 2013. 3

[22] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. TANet: Robust 3D object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11677–11684, 2020. 2

[23] Gregory P Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3D object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[24] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2019. 2, 3

[25] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. DOPS: Learning to detect 3D objects and predict their 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11913–11922, 2020. 6

[26] Anshul Paigwar, Ozgur Erkent, Christian Wolf, and Christian Laugier. Attentional PointNet for 3D-object detection in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[27] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. ImVoteNet: Boosting 3D object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 1, 3

[28] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 3, 5, 6

[29] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 1, 3

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 3

[31] Yuzhuo Ren, Chen Chen, Shangwen Li, and C-C Jay Kuo. Context-assisted 3D (C3D) object detection from RGB-D images. *Journal of Visual Communication and Image Representation*, 55:131–141, 2018. 3

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1, 2

[33] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 3

[34] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, and Hongsheng Li. Part-aˆ 2 net: 3D part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670*, 2019. 2

[35] Yifei Shi, Angel X Chang, Zhelun Wu, Manolis Savva, and Kai Xu. Hierarchy denoising recursive autoencoders for 3D scene layout prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1771–1780, 2019. 3

[36] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3D object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1991–1999, 2019. 2

[37] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 5

[38] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 2

[39] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1

[40] Xinshuo Weng and Kris Kitani. Monocular 3D object detection with pseudo-LiDAR point cloud. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 2

[41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10447–10456, 2020. 6

[42] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Vote-based 3D object detection with context modeling and SOB-3DNMS. *International Journal of Computer Vision*, 129(6):1857–1874, 2021. 1

[43] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 3

[44] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, pages 6737–6746, 2019. 1

[45] Shichao Yang and Sebastian Scherer. CubeSLAM: Monocular 3D object SLAM. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. 1

[46] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-dense 3D object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019. 2

[47] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving. In *ICLR*, 2020. 2

[48] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3DNet: 3D object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 6

[49] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11079–11087, 2020. 6

[50] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2