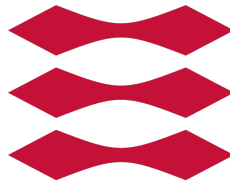


---

# VotANet: Vote-based Attentional Network for Indoor Object Detection

---

DTU



Authors	Student Number
Zhao Gong	s200101
Manxi Lin	s192230
Kun Du	s192231

August, 2020

Technical University of Denmark

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>2</b>
<b>3</b>	<b>VotANet Architecture</b>	<b>4</b>
3.1	Backbone . . . . .	4
3.1.1	Set abstraction . . . . .	4
3.1.2	Feature propagation . . . . .	5
3.2	Feature re-weighting . . . . .	5
3.3	Voting module . . . . .	6
3.4	Detector . . . . .	7
<b>4</b>	<b>Experiments</b>	<b>7</b>
4.1	Dataset . . . . .	7
4.2	Experiment environment . . . . .	7
4.3	Result . . . . .	9
<b>5</b>	<b>Conclusion</b>	<b>10</b>
	<b>References</b>	<b>11</b>

# 1 Introduction

3D object detection ground on deep learning is getting attention in recent years. 3D bounding box prediction outperforms 2D object detection in many tasks, e.g., indoor robot navigation, robot grasping, autonomous driving, and augmented/virtual reality [12, 16, 22–24, 30, 32]. Since LiDAR is quickly updated, deep learning makes self-driving cars possible. In the indoor scenario, RGB-D camera still plays an important role in 3D-scene understanding for indoor robots, because of the lower price and higher resolution. However, 3D data, especially the unstructured point clouds, make the detection task more challenging. Popular 2D object detection methods are difficult to be implemented directly on point clouds. The number of points directly affects the size and the computational complexity of the neural network, impacting the quality of feature learning [1]. Even though high-performance GPUs and deep neural networks provide the possibility of handling tons of data, to make a balance between precision and real time, well-designed network architectures are strongly needed.

A handful of ideas have been proved ingenious in this field. Recently, Vote-Net [24], which introduces Hough voting procedure into end-to-end 3D object detection, draws much attention. However, since Vote-Net is a simple combination of PointNet++ [26] and Hough voting modules, we deem there is still large room for improvement in VoteNet.

Fig. 1 presents the overview of our work, VotANet. Based on Vote-Net, by fusing attention mechanism and edge argumentation, we propose a novel end-to-end indoor object detection network. In addition, we evaluate the 3D detection performance of our architecture on SUN RGB-D data set [27].

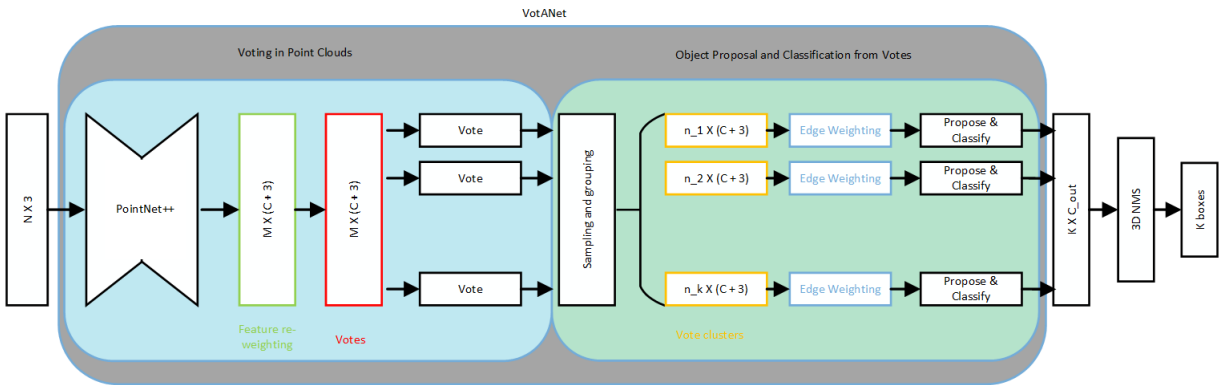


Figure 1: The overall sctructure of our VotANet

## 2 Related work

**3D object detection from point clouds** Previous 3D object detection methods can be roughly divided into point-based works [6, 15, 24–26] and grid-based works [7, 13, 20, 34, 35], even though some works [29, 33] try to incorporate the advantages from both methods. Point-based networks directly work on raw point clouds while grid-based ones tend to discretize the space and reproject the points before learning features so as to implement 2D or 3D convolutional neural network. Although grid-based methods have achieved state-of-the-art results in some famous outdoor data sets [10], in the indoor scenario [27], point-based works are more popular, because of the more flexible receptive fields. Qi et. al leveraged the first point-based work, PointNet [25], whose feasibility in handling unstructured data is mathematically proved and experimentally demonstrated. Applying Set Abstraction and Feature Propagation modules, PointNet++ [26] acts more like a convolutional neural network. HGNet [6] is based on graph convolutions and captures features from point clouds hierarchically. [15] describes a new point-based module to accelerate feature propagation. These point-based methods are mostly following the sequence of PointNet++: extracting local features via set abstraction, and then interpolate features to raw point clouds by feature propagation.

**Deep Hough voting** Hough transform is originally used to detect lines or circles by collecting peaks in parametric space. Through extending point votes to image-patch votes, [9, 17] first introduced this technique into object detection field. For 3D object detection, [3, 14] adopted a pipeline similar to traditional 2D detection. Vote-Net [24], shown in Fig. 2, is the first deep network incorporating Hough voting modules. In Vote-Net, each seed generated by PointNet++ votes for a certain number of potential object centroids, which are then put into a modified PointNet++ to regress 3D bounding boxes. In 3D object detection, it has been proven that the voting algorithm has higher efficiency and accuracy than the traditional Region Proposal Network [21]. Works [1, 32] based on Vote-Net followed this pipeline. MLCVNet [32] exploited contextual information and global features. [1] replaced Farthest Point Sampling with unsupervised clustering algorithm to optimize the output of Hough voting module. The result of [32] and [1] demonstrates the inadequacies of Vote-Net. In our work, we will explore possible further improvements.

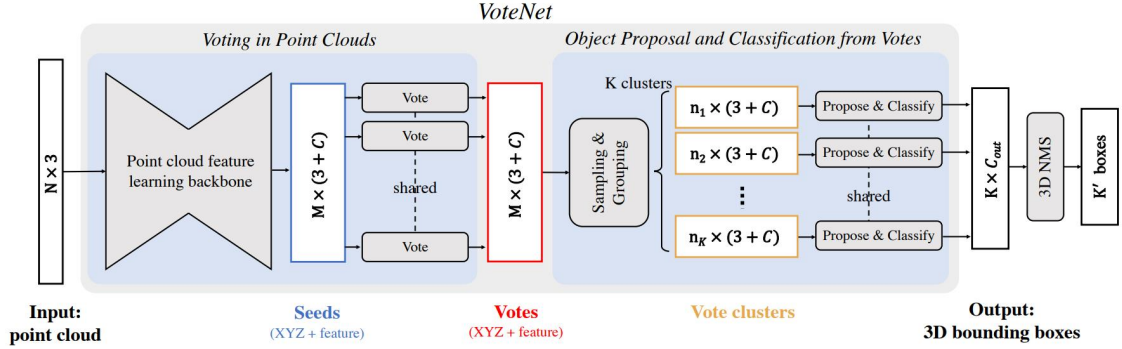


Figure 2: The structure of the Vote-Net [24]

**Attention mechanism** Attention mechanism is a tool that is widely used in the natural language process field, soon gotten researchers' spotlights after being introduced to the world of computer vision. The mechanism's work can be generally described in the following three regions. The first one would be space-based attention mechanism. Google Deepmind group introduced [13] to transform the input image for neural networks to learn easier. Their work would locate the position of the object and project it to an angle that easier for the network to learn. But this would consume high computation resources for the transformation of each object proposal. Never the less, [4] employs two sub-networks to alleviate this issue. By using the combination of the coarse model and fine model, [4] achieves higher accuracy while using fewer resources. The second one is the channel based attention. [11] demonstrates another approach to use this tool in the field of computer vision. After convolution, the data would be squeezed and excited for generating the weight of each channel. The generated weight would be used for multiplying with the original data to get channel attention based data. Since the computation process is squeeze, excitation, and multiple with the original data, this approach is much efficient. The last one combines the aforementioned two approaches. [31] combines the space-based attention mechanism and the channel-based approach together to form a novel tool. The data processed by convolution would be first sent to the former channel based module for re-weighting. After re-weighted, the data would be transformed in the later space-based module.

**Point cloud edge detection** [8, 18, 19] have shown that using edge detection as an auxil-

iary task improves performance for 3D semantic segmentation. EPN [2] introduced edge detection in 3D object detection. In EPN, 2D binary mask images, generated by detected edges, provide additional features for the detection task. On real world datasets [5], EPN outperformed all the state-of-the-art methods at that time.

### 3 VotANet Architecture

We are here to present our detection network: VotANet. As demonstrated in Fig. 1, the Whole pipeline would be introduced as following: the backbone, feature re-weighting module, voting module. In this section, we will describe the approach we take for tackling the project. We will first introduce the backbone network of our project, followed by the feature re-weighting module. After the introduction of the re-weighting module, the voting module would be analyzed.

#### 3.1 Backbone

The backbone of our VotANet is PointNet++ [26]. PointNet++, consisting of set abstraction and feature propagation modules, can be viewed as an extension of PointNet [25] with added hierarchical structure.

##### 3.1.1 Set abstraction

Set abstraction plays an important role in hierarchical point set feature learning. The set abstraction module is made of three layers: Sampling layer, Grouping layer and PointNet layer.

##### Sampling layer

Sampling is necessary for handling mass data. In Sampling layer, given input point clouds  $P$ , iterative Farthest Point Sampling (FPS) [] is implemented to get a subset of points  $P_s$ . In each iteration in FPS, the point in  $P$ , which is the farthest to  $P_s$ , is considered to be the new member in  $P_s$ . Compared with random sampling, FPS has better coverage of the entire point set. We denote the sampling points as centroids of the point set. Given a  $N$ -point set of size  $N \times 3$ , the centroids can be represented as  $N' \times 3$ .

##### Grouping layer

Grouping layer is used to capture local features of centroids in their small neighbourhoods. In this layer, the features of neighbour points are stacked over the features of their centroids. That is, the output of grouping layer will be of size  $N' \times K \times 3$ , if  $K$  is the number of points in the neighbourhood of centroids. PointNet++ applies ball query to find all points that are within a radius to the query point. An upper limit of  $K$  is set to keep the output of size  $N' \times K \times 3$ .

### PointNet

PointNet is a combination of max-pooling layer and multi-layer perceptron (MLP) networks. PointNet is applied to encode the local features learned in Grouping layer.

#### 3.1.2 Feature propagation

In set abstraction layer, the original point set is subsampled. In some cases, such as semantic segmentation, we want to obtain point features for more points other than centroids. The feature propagation module is used to propagate features from subsampled points to a given number of points by interpolating feature values of points at coordinates. More specifically, the interpolation is based on weighted average of inverse distance to  $k$  nearest neighbours. In our VotANet, raw point clouds are first put into PointNet++. Following the format below, the output of PointNet++ is of size  $M \times (3 + C)$ , where  $M$  and  $C$  are determined by feature propagation and the architecture of PointNet respectively. These  $M$  points from PointNet++ are called seeds.

## 3.2 Feature re-weighting

Typically, networks treat the features equally. This method is grounded on that we, the designers, do not know which features are more or less important. In this work, we introduce the channel-based attention mechanism for improving work. After being processed by the backbone network of the Vote-Net [24], features are treated equally and sent to the next module. However, the result of the network produced might be influenced by some trivial information. Based on the information theory [28], we argue that channels that carry more information have more information entropy than that deliver less information. Fig. 3 demonstrates the process this module. To make the higher information entropy channels are treated with more weight, we use a score function that will mark the grade

of each channel. By multiplying the channels with their grades, the channels are thus to be re-weighted. In other words, channels are weighted by their information entropy.

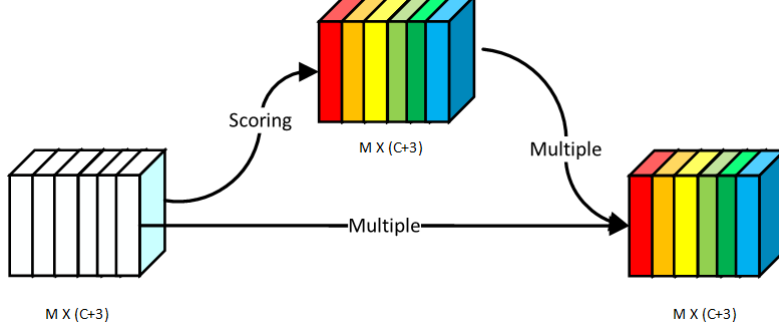


Figure 3: The workflow of the feature weighting module

### 3.3 Voting module

**Voting module** In Vote-Net, voting module is a multi-layer perceptron network with fully connected layers, ReLU and batch normalization. Given a set of seeds  $\{S_i\}$  from PointNet++, which is of size  $M \times (C + 3)$ , the network takes seed features as input and outputs the Euclidean space and feature offset  $\Delta F_i \in \mathbb{R}^{C+3}$  for each seed point. The network is supervised by a regression loss presented below.

$$L_{voting} = \frac{1}{M} \sum_i^M \|\Delta x_i - \Delta x_i^*\| \cdot b \quad (1)$$

where  $M$  is the number of seed points,  $\Delta x_i \in \mathbb{R}^3$  is the Euclidean space offset predicted by the network, offset  $\Delta x_i^* \in \mathbb{R}^3$  is the displacement from the seed position to the bounding box center of the object it belongs to in ground truth,  $b$  is a binary value indicating whether the seed point is on the object.

Following the definition in Vote-Net, in VotANet, we define vote points  $\{V_i\}$  as the sum of  $\{S_i\}$  and  $\{\Delta F_i\}$ . In short, vote points are generated by shifting the seed points and the offset is given by a multi-layer perceptron network.

**Vote abstraction** Even though vote points have aggregated local features from raw point clouds, to generate object proposals, we apply another set abstraction module to aggregate vote point features. After FPS and ball query, vote points are divided into a given number of clusters. The centroid of each cluster is thought as our prediction on the object bounding box center.



**Edge weighting** Since edge points describe the shape of objects in 3D point sets, they can provide more information than points inside the objects. EPN [1] applies an effective method for point cloud edge detection that evaluates symmetry of a group of nearest neighboring points. In our VotANet, we reweighted features from each vote clusters based on the possibility of a vote point being an edge point. Given a query vote point  $p_i$ , considering its  $k$ -nearest neighbours denoted as  $V_i = \{n_1, n_2, \dots, n_k\}$ , the weight  $W_{p_i}$  is expressed as follows:

$$W_{p_i} = \frac{\|c_i - p_i\|_2 - \lambda \cdot \min_{n \in V_i} \|p_i - n\|_2}{\sum_{p_j \in C_i} W_{p_j}} \quad (2)$$

where  $C_i$  is the vote cluster that  $p_i$  belongs to and  $c_i$  is the centroid point of the cluster generated by FPS.

### 3.4 Detector

We follow the work of Vote-Net [24]. We used a shared PointNet to regress a bounding box for each vote cluster, predicting the orientation, position, size and class of a 3D bounding box. The loss function includes center regression, heading angle estimation and box size estimation and all the loss is smooth- $L_1$ .

## 4 Experiments

### 4.1 Dataset

In this work, we use SUN RGB-D as our network dataset. SUN RGB-D [27] is a single-view RGB-D dataset for 3D scene understanding. It was captured by four different sensors and contains 10,000 RGB-D images annotated with amodal oriented 3D bounding boxes for 37 object categories. To feed the data to our network, we first convert the depth images to point clouds using the provided camera parameters. We follow a standard evaluation protocol and report performance on the 10 most common categories.

### 4.2 Experiment environment

#### MATLAB

We use Matlab to preprocess the SUN RGB-D data to convert the depth images to point

clouds.

### Training server

We used the AWS GPU instance as the training server.

CPU	4 cores
GPU	NVIDIA K80
Memory	61 GB
Unbuntu	16.04
Python	3.7
Pytorch	1.3+
CUDA	10.2

Table 1: The environment of training server

### Toolbox

Our work is based on the MMDetection3D toolbox. The toolbox is from MMLab in Chinese University of Hong Kong which specially focuses on the study of computer vision and deep learning.

### 4.3 Result

Classes	AP@0.25	AR@0.25	AP@0.5	AR@0.5
bed	0.8546	0.9709	0.5597	0.7029
table	0.4961	0.8654	0.2109	0.4344
sofa	0.6745	0.9123	0.5072	0.6730
chair	0.7748	0.8848	0.5603	0.6761
toilet	0.9076	0.9862	0.6167	0.7379
desk	0.2436	0.8082	0.0597	0.3114
dresser	0.2897	0.7798	0.1138	0.4037
night_stand	0.5985	0.8588	0.4113	0.6431
bookshelf	0.3334	0.7057	0.0634	0.2376
bathtub	0.7623	0.8980	0.4275	0.5306
overall	0.5935	0.8670	0.3531	0.5351

Table 2: The baseline of 3D object detection on SUN RGB-D val set

Classes	AP@0.25	AR@0.25	AP@0.5	AR@0.5
bed	0.8474	0.9592	0.5216	0.6660
table	0.4945	0.8560	0.1955	0.4152
sofa	0.6684	0.8963	0.5022	0.6667
chair	0.7735	0.8888	0.5448	0.6729
toilet	0.8848	0.9724	0.5018	0.6621
desk	0.2722	0.8140	0.0581	0.2928
dresser	0.3436	0.8165	0.2154	0.4771
night_stand	0.6105	0.8784	0.3965	0.6000
bookshelf	0.3288	0.7092	0.0675	0.2199
bathtub	0.7696	0.8776	0.2905	0.4286
overall	0.5993	0.8669	0.3294	0.5101

Table 3: The result of 3D object detection on SUN RGB-D val set



Figure 4: Qualitative results of 3D object detection in SUN RGB-D val set

## 5 Conclusion

In this work, we introduced an improved Vote-Net: a simple, yet powerful 3D object detection network inspired by Hough voting, attention mechanism and point cloud edge detection. The network learns to vote for object centroids directly from point clouds and learns to aggregate votes through their features and local geometry to generate high-quality object proposals. To improve performance, we add the attention mechanism in the features extracted by SA layer of PointNet++ and design a score to evaluate the possibility that points belongs to the edge of the object. It can be seen from the result that the model shows an improvement of the original Vote-Net in some aspect. In the future work, we intend to use more CUDA programming methods to improve computing efficiency, enlarge our training set and find more reasonable parameters to increase the accuracy of detection.

## References

- [1] S. M. Ahmed. Density Based Clustering for 3D Object Detection in Point Clouds. pages 10608–10617.
- [2] S. M. Ahmed, P. Liang, and C. M. Chew. EPN : Edge-Aware PointNet for Object Recognition from Multi-View 2 . 5D Point Clouds. pages 3445–3450, 2019.
- [3] R. S. Alexander Velizhev and K. Schindler. Implicit shape models for object detection in 3d point clouds. *International Society of Photogrammetry and Remote Sensing Congress*, 2012.
- [4] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. Courville. Dynamic capacity networks. In *International Conference on Machine Learning*, pages 2549–2558, 2016.
- [5] M. N. A. D. M. Y. C. R. Qi, H. Su and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *IEEE conference on computer vision and pattern recognition*, pages 5648–5656.
- [6] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu. A Hierarchical Graph Network for 3D Object Detection on Point Clouds. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 392–401, 2020.
- [7] X. Chen, H. Ma, J. Wan, B. Li, and T. 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.
- [8] J. S. G. Bertasius and L. Torresani. High-for-low and low-for- high: Efficient boundary detection from deep object features and its applications to high-level vision. *IEEE International Conference on Computer Vision*, pages 504–512.
- [9] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. *Decision forests for computer vision and medical image analysis*, pages 143–157, 2013.
- [10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

- [11] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [12] A. Inc. Scanning and detecting 3d objects. Website:[https://developer.apple.com/documentation/arkit/scanning\\_and\\_detecting\\_3d\\_objects#see-also](https://developer.apple.com/documentation/arkit/scanning_and_detecting_3d_objects#see-also). Accessed June 30,2020.
- [13] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.
- [14] M. P. Jan Knopp and L. V. Gool. Scene cut: Class-specific object detection and segmentation in 3d scenes. *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 180–187, 2011.
- [15] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. 2018.
- [16] M. B. A. D. John McCormac, Ronald Clark and S. Leutenegger. Fusion++: Volumetric object-level slam. In *International Conference on 3D Vision (3DV)*, pages 32–41, 2018.
- [17] N. R. L. V. G. Juergen Gall, Angela Yao and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, pages 2188–2202, 2011.
- [18] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. 2015.
- [19] G. P. K. M. L.-C. Chen, J. T. Barron and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. *IEEE conference on computer vision and pattern recognition*, pages 4545–4554.
- [20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

- [21] D. Z. W. C. H. T. M. Engelcke, D. Rao and I. Posner. Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1355–1367, may 2017.
- [22] J. Park, D. Seo, M. Ku, I. Jung, and C. Jeong. Multiple 3d object tracking using roi and double filtering for augmented reality. In *2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering*, pages 317–322, 2011.
- [23] Y. Park, V. Lepetit, and Woontack Woo. Multiple 3d object tracking for augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 117–120, 2008.
- [24] C. R. Qi, O. Litany, K. He, and L. J. Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5099–5108. Curran Associates, Inc., 2017.
- [27] S. L. S. Song and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [28] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.
- [29] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. 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.

- [30] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.
- [31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [32] Q. Xie, Y.-k. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, and J. Wang. MLCVNet : Multi-Level Context VoteNet for 3D Object Detection.
- [33] M. Yan, Z. Li, X. Yu, and C. Jin. An End-to-End Deep Learning Network for 3D Object Detection From RGB-D Data Based on Hough Voting. 8, 2020.
- [34] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.
- [35] Y. Zhou and O. 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.