Uncertainty-Aware 3D Object Detection with Voxel-based Methods

Anirudh Achanta
University of Minnesota
achan009@umn.edu

Shreyas Kallapur
University of Minnesota
kalla120@umn.edu

Sidhanth Krishna
University of Minnesota
krish323@umn.edu

Abstract

Accurate and reliable object detection models are essential for safe autonomous driving systems. Anchor-based 3D models are not fully capable of accurately predicting the bounding boxes if they are not aligned to the vehicle's global axes. Bounding boxes can also be ambiguous due to occlusion and incorrect labeling. We propose a novel method for combating this task by re-parameterizing the bounding boxes using uncertainty-based Gaussian techniques, inspired by 2D detection models like Gaussian YOLOv3. We also incorporate a 2D representation of the orientation angle to prevent discontinuities during the calculation of the loss. We build upon existing state-of-the-art architectures using OpenPCDet and factor in these new representations to train our models on the KITTI dataset. Our model beats the Voxel R-CNN mAP by ***+2.6%*** *on the KITTI Moderate test set and also achieves higher mAP on the bounding box predictions in KITTI Hard (****+1.9%****).*

1. Introduction

3D Perception is a crucial component of state-of-the-art driving systems to detect and localize surrounding objects. Compared to 2D detection, 3D detection poses a different set of difficulties, such as 3D Point Clouds being much sparser when compared to 2D images. The regions where we would like to pay attention to also occupy a much smaller volume. This primarily poses a challenge in efficiency, but also adds to the complexity of designing a working solution. Since the sensory information is limited, it is unrealistic to expect that the predictions made by the detectors are fully accurate. It is thus important to also know the level of uncertainty in the predictions being made, and these uncertainties must be reliable for the system to use them correctly.

Existing 3D detection methods are either *voxel*-based or *point*-based. Point-based methods directly act on the raw point cloud, abstracting a set of point-representations. These methods are able to use precise point location information from the point cloud to make more accurate predictions about object localization, albeit at the cost of efficiency. Nearest-neighbor search in these methods usually is costly, and is one of its major drawbacks. Voxel-based methods (such as VoxelNet [23]) divide the input space into regular grids, which makes it easier to leverage Convolutional Neural Networks (CNNs) while providing a more memory-efficient way for feature extractions. The downsides of this are that we lose precise spatial information that point clouds provide. Voxel-based methods can thus benefit from modeling the uncertainty in their predictions, as we can improve the localization confidences while maintaining all their other advantages.

State-of-the-art methods today employ either of voxel or point-based models (or a combination of both) to perform the task of object detection. The detection heads in these networks are usually comprised of a classification head (to output the class of the object being detected) and a bounding box regression head (that predicts the bounding box parameters). These bounding boxes are usually derived from a set of defined anchors generated from the 2D bird-eye-view (BEV), and are regressed upon to improve their accuracy. In this work, we propose a novel method using Gaussian modelling to regress on the centers and sizes of the boxes alongside the traditional regression and classification heads. This re-structuring of the bounding box regression helps in generating more accurate bounding boxes, as well as giving us a measure of uncertainty that can be visualized and used further.

Our main contributions in this work thus include, the design of an extended Voxel R-CNN [2] network to predict the uncertainties in the bounding box and a new loss function that can take into account these uncertainties to improve predictions. We also parameterize the orientation angle into a 2D coordinate system and combine its loss with the aforementioned loss to complete the redesigned regression loss function. Our experiments confirm that our model is able to effectively learn how to reduce uncertainties in bounding boxes with an inference time very similar to our baseline as well as improve prediction performance in more occluded and ambiguous cases.

2. Dataset

We use the KITTI dataset [4], created using an autonomous driving platform where the data was recorded while driving a car with an attached Velodyne laser scanner and multiple high-resolution cameras. The dataset has Velodyne LiDAR Point Clouds, images (from four different cameras), classification labels and calibration matrices for each data point. It includes monocular images and bounding boxes with a total of 80,256 labeled objects spread across 7481 training data points and 7518 test data points. Our proposed model only uses the Velodyne LiDAR point cloud data for detecting objects and measuring uncertainty in 3D space. Additionally, we also use the ground planes generated for the KITTI dataset, as used by [8] to get a better estimation of the plane corresponding to the ground, and use that to augment our training data. We do not generate these ground planes during testing to reduce prediction times.

3. Related Work

With the continuous growth of computational power and increased number of research groups working on object detection we observed an expected rise in the object recognition and detection algorithms. There are mainly two types of object detection models – single-stage detectors and two-stage detectors. Single-stage detectors detect and classify objects in a single pass while two-stage detectors do this over two stages – in the first stage they detect the region where the object is likely to lie in and in the second stage they classify the object which has been detected and regress on its location. While single-stage detectors are fast, two-stage detectors have higher accuracy. We explore some of these detectors for both 2D and 3D in this section, and consider how uncertainties have been incorporated in related works in both 2D and 3D.

2D-object detection R-CNN [6] is a two stage object detector that searches for the possible location of the object using a selective search [19] algorithm, followed by a CNN which detects the possible objects in these regions. Fast R-CNN [5] improves on R-CNN by modifying it to train it in one go, thus increasing mAP. Faster R-CNN [15] removes selective search from R-CNNs and replaces it with a trainable Region Proposal Network (RPN) for faster inference and increased accuracies. YOLO [13] is a single-stage object detector which divides the image into grids where it searches for objects. In Gaussian YOLOv3, [1], models the bounding boxes of YOLOv3 [14] with Gaussian parameters and redesigns the loss function. [7] proposes using a KL Divergence Loss along with variance voting and soft-NMS to improve localization accuracies. Our model is inspired by these methods and extends the concept of localization uncertainty to 3D using a similar loss to estimate uncertainty.

3D-object detection PointNet [12] is a seminal work in 3D object detection in which the authors emphasize permutation, transformation invariance and interaction between points from raw point cloud data. VoxelNet [23] is also another prominent work that uses 3D voxels to process the input space using 3D CNNs. Sparse 3D convolutions have also been developed to optimize these sort of approaches [20]. PV-RCNN [16] integrates 3D voxel convolutional nets and PointNet-based feature set abstraction to generate better RoI proposals. PV-RCNN++ [17] extends this by producing more representative keypoints and aggregating local features better. PointNet-based methods [12, 11, 21] usually have a flexible receptive field owing to their learning strategy being based directly on the points instead of voxels. We primarily use the uncertainty-based additions with voxel-based methods, as the loss of precise point information can contribute to increased uncertainty. Frameworks like that of AFDet [3] on the other hand, propose a novel architecture for embedded systems by dropping the use of both anchor boxes and non-maximum suppression for efficient post-processing and thus avoiding any tuning of anchor parameters.

There exist multiple recent works in probabilistic object detection such as LaserNet[9] and similar works including [10] where they attempt to improve localization uncertainty by modeling the bounding box as a distribution directly, where the variance indicates the boundaries of the box. Our work instead focuses on reducing existing bounding box uncertainties without fully modeling the box itself as a distribution, rather, we predict the uncertainties by modeling the predicted box parameters as independent univariate Gaussians. All these works have the same motivation as ours, that traditional object detection methods predict a single bounding box for each detection with a probability score. Though this score helps us understand the existence and semantic uncertainty, it does not indicate anything about the localization uncertainty. To combat this, we explicitly output this uncertainty directly. A similar line of work was carried out in [22] where uncertainty was used but the primary intention was to aid in improving 3D tracking accuracy.

4. Baseline

We use the Voxel R-CNN model on the KITTI Cars Moderate dataset as our baseline. Voxel R-CNN is a SOTA architecture that uses a 3D backbone network such as VoxelNet, a 2D BEV Region Proposal Network (RPN) and a detect head. The model takes raw point clouds as inputs and first voxelizes them. The 3D backbone network then extracts features from the voxelized representation. The sparse 3D voxels are then converted to BEV representation with which we use the RPN to generate 3D region propos-

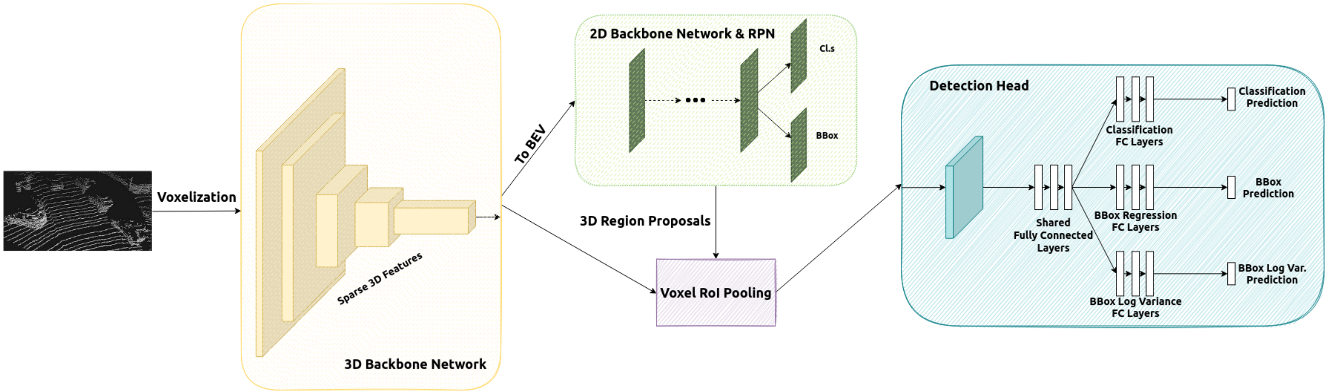

Figure 1: Network Architecture

als. Voxel RoI pooling is done on these and are fed to the detection module for generating the outputs. We use this model as our base network, and extend this further to include our variance predictions. The results of the baseline model are shown in Table 2. The baseline model does not account for localization variance at all. It only accounts for the existence and semantic uncertainty through the classification head. This limitation is what we wish to overcome with our model.

5. Method

5.1. Network Architecture

In this section, we present the design of our network architecture, a voxel-based two stage framework for 3D object detection similar to Voxel R-CNN. Most of the backend network retains the same structure as Voxel R-CNN, as our method is extensible to any network predicting bounding box parameters. The 3D backbone network extracts sparse 3D features, which are fed to the 2D backbone. The Region Proposal Network that immediately follows this layer produces 3D region proposals, which are fed to the Voxel RoI pooling layer along with the sparse 3D features. These are followed by a detection head, where after a few shared layers, we have three separate heads. The classification head predicts the objectness score of the prediction. The regression head predicts the bounding box coordinates and the uncertainty head predicts the log-variances of the bounding box coordinates. This is outlined in Figure 1. Below we discuss some of these modules in detail. Since our innovation mainly lies in uncertainty prediction, we discuss it first.

5.2. Uncertainty Prediction

We aim to estimate the uncertainty along with the location using the KL Divergence Loss. Formally, our network predicts a probability distribution instead of only bounding box parameters. This includes the coordinates of the center (x, y, z) as well as the parameters representing the size of the box (w, l, h). Though the distribution could be more complex ones like multivariate Gaussian or a mixture of Gaussians, in this paper we assume the above six parameters are independent and use univariate gaussian for simplicity. For any such parameter x and its estimated location x_e, we have

$$P_\rho(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x - x_e)^2}{2\sigma^2}\right)$$

Standard deviation σ measures uncertainty of the estimation. When $\sigma \to 0$, it means our network is extremely confident about estimated bounding box location. It is produced by a fully-connected layer on top of the Voxel R-CNN head. The ground-truth bounding box can also be formulated as a Gaussian distribution, with $\sigma \to 0$, which is a Dirac delta function.

$$P_D(x) = \delta(x - x_g)$$

where x_g is the ground truth bounding box. This leads us to modify our loss function to account for both our predictions and ground truths being formulated in terms of probability distributions.

5.3. Bounding Box Loss

To achieve the goals of better object localization, we need to estimate the ρ such that the KL-Divergence between $P_\rho(x)$ and $P_D(x)$ over N samples is minimized.

$$\rho = \arg\min_\rho \frac{1}{N} \sum D_{KL}(P_D(x) || P_\rho(x))$$

The KL-Divergence as the loss function L_{reg} for bounding box regression. The classification loss L_{cls} remains the same. For a single sample:

Model Name	$\text{AP}^{0.7}_{\text{easy}}(\%)$	$\text{AP}^{0.7}_{\text{moderate}}(\%)$	$\text{AP}^{0.7}_{\text{hard}}(\%)$
Voxel R-CNN (paper)	90.90	81.62	77.06
Ours	**92.21**	**84.79**	**82.66**

Table 1: Results of our model calculated by 40 recall positions for AP_{3D}

KL Loss	θ transform	$\text{AP}^{0.5}_{\text{easy}}$	$AP^{0.5}_{\text{moderate}}$	$AP^{0.5}_{\text{hard}}$	$AP^{0.7}_{\text{easy}}$	$AP^{0.7}_{\text{moderate}}$	$AP^{0.7}_{\text{hard}}$
		98.7242	94.9308	**94.5958**	92.1587	**85.0157**	82.4832
✓		98.5830	96.2694	94.5531	**92.2057**	84.7861	**82.6573**
✓	✓	**98.9024**	**96.6000**	94.5751	92.0782	84.8040	82.5419

Table 2: Ablation Study Results for AP_{3D} calculated by 40 recall positions using the OpenPCDet baseline and our models

$$\begin{aligned} L_{reg} &= D_{KL}(P_D(x)||P_\rho(x)) \\ &= \int P_D(x)\log P_D(x)dx - \int P_D(x)\log P_\rho(x)dx \\ &= \frac{(x_g - x_e)^2}{2\sigma^2} + \frac{\log\sigma^2}{2} + \frac{\log 2\pi}{2} - H(P_D(x)) \end{aligned}$$

$H(P_D(x))$ does not depend on the estimated parameters. Removing the constants from the above equation, we can write it as a proportionality:

$$L_{reg} \propto \frac{(x_g - x_e)^2}{2\sigma^2} + \frac{\log\sigma^2}{2}$$

We see that when $\sigma = 1$, this decomposes to the standard L2 loss:

$$L_{reg} \propto \frac{(x_g - x_e)^2}{2}$$

This loss function is differentiable as well, with

$$\begin{aligned} \frac{\partial}{\partial x_e} L_{reg} &= \frac{x_e - x_g}{\sigma^2} \\ \frac{\partial}{\partial \sigma} L_{reg} &= -\frac{(x_e - x_g)^2}{\sigma^3} + \frac{1}{\sigma} \end{aligned}$$

Since σ is in the denominator in the above equations, we can find that the gradient can sometimes explode during training, especially in the beginning as we initialize with very small weights. Our network thus predicts the log-variance, $\alpha = \log(\sigma^2)$. This would turn our loss function into:

$$L_{reg} \propto \frac{e^{-\alpha}}{2}(x_g - x_e)^2 + \frac{\alpha}{2}$$

During predictions, we convert back to σ. For $|x_g - x_e| > 1$, we adopt a loss similar to the Smooth-L1 loss:

$$L_{reg} = e^{-\alpha}\left(|x_g - x_e| - \frac{1}{2}\right) + \frac{\alpha}{2}$$

We initialize our variance layer weights with a Xavier normal distribution, and our KL loss will be similar to the Smooth-L1 loss at the beginning of training due to α being very small. This will allow the model to learn directly from the Smooth-L1 loss during the beginning phases of training, but later begins to account for uncertainties as well.

5.4. Orientation Angle Parameterization

The orientation angle θ, ranges from $-\pi$ to π. There is a point of discontinuity that arises when rotation values circularly jump from $-\pi$ to π. This can potentially cause a large jump in the loss values for a small difference in θ, which can hamper training efforts. Although many methods do not directly make any changes to account for this difference, we additionally wanted to evaluate if we can see an improved model performance by parameterizing this θ as a 2-D Cartesian coordinate pair to remove the discontinuity. The θ is now transformed as $(\cos\theta, \sin\theta)$, and we use the Smooth-L1 loss on the difference between these coordinates instead of the θs directly.

6. Results

As previously mentioned in Section 2, we use the KITTI Dataset to predict cars using our model. To train our model, we used the OpenPCDet [18] framework to create and modify our network. We specifically used the Voxel R-CNN baseline model and added our changes in the architecture and the loss function. We used one RTX 2070 GPU to train on the KITTI train set, and tested our models after training on the KITTI test set. We used the same hyperparameters as used for Voxel R-CNN. Our results are summarized in Table 1. Our model beats the baseline implementation used in the Voxel R-CNN paper by achieving higher mAP for across the KITTI dataset. We also plot our results for AP_{BBox} in Table 3, to check our predictions of bounding boxes. We see that our model performs better on the KITTI Hard dataset by almost 2%, indicating that the uncertainty measure helped in

Model	$AP^{0.7}_{easy}(\%)$	$AP^{0.7}_{moderate}(\%)$	$AP^{0.7}_{hard}(\%)$
Voxel R-CNN	98.77	**94.93**	92.56
Ours	**98.95**	94.73	**94.46 (+1.9%)**

Table 3: Bounding Box Results of our model calculated by 40 recall positions for AP_{BBox}

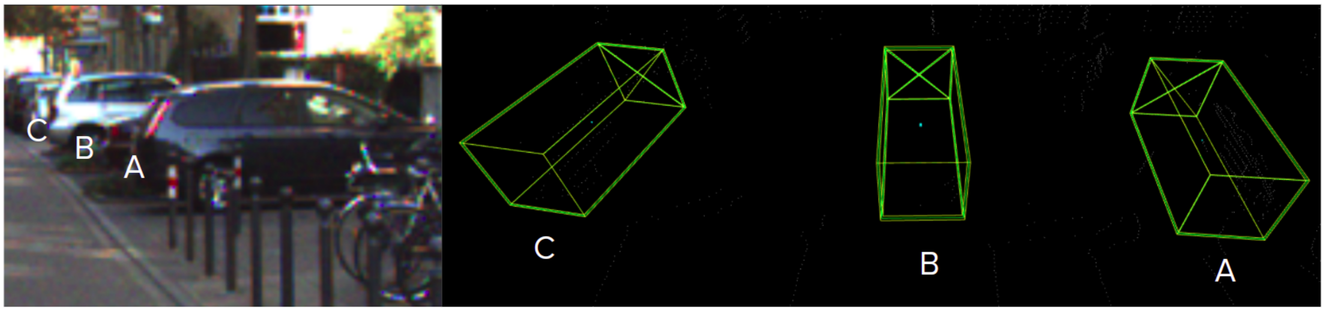

Figure 2: Visualizing uncertainty for occluded objects. Car B is occluded as it is right behind Car A, and has higher uncertainty as can be seen from the bigger center blue box at the center, as well as bigger standard deviations for the w, l, h values. Cars A and C are in direct lines of sight of the ego-vehicle and have higher confidence predictions.

more difficult cases as we intended.

6.1. Ablation Study

To ascertain the effectiveness of our additions, we perform an ablation study by training the network under the following scenarios -

1. Baseline implementation of Voxel R-CNN from OpenPCDet without any modifications to the network or loss
2. Our modified network with only the KL Divergence Loss, with only a Smooth-L1 loss on the orientation angle θ without any transformations
3. Our modified network, with both the additions of KL Divergence Loss and θ transformation

We see that the results with or without the θ transformation do not vary much, indicating that the transformation was not very effective in improving the predictions. Compared to the base network, the models using KL loss perform better in most cases.

7. Conclusion

In this paper, we present a novel method to improve the localization accuracy of 3D object detection models. From a standard 3D backbone network, we introduced how the modification of any standard loss function to that using a KL-Divergence Loss with the inclusion of the variance of the bounding box parameters adds to improve the bounding box predictions in various scenarios. We see optimistic results, especially in tough scenarios in KITTI Hard, where the bounding box precision increases. This formulation can be extended to any framework that leverages similar bounding box regression techniques to include the localization uncertainty. There is a minimal drop in inference times due to the extra log-variance head, but this can be mitigated by using a faster backbone network.

8. Contributions

Every member of the team was responsible for the individual tasks as mentioned below but all the members contributed to all the tasks through extensive discussions and exchange of ideas.

- Anirudh Achanta - Responsible for full code-review, testing and modifying Open3D visualizations for uncertainties, building VFE layer and developing large parts of the report.

- Shreyas Kallapur - Responsible for programming Region Proposal Networks and the ROI-heads. Substantial efforts were also made to utilize ground plane features for augmenting ground truth data.

- Sidhanth Krishna - Responsible for literature review, ideation of KL Divergence Loss and building uncertainty heads for the bounding box predictions.

9. Future Work

Our work can be extended to various use cases, for example in tracking applications, where this uncertainty can be a useful measure to incorporate. We can also modify our network to include uncertainty in further locations, such as using Soft-NMS instead of traditional NMS. As previously mentioned, this can be extended to different architectures for various uses, and it is not restrictive to the Voxel R-CNN framework used here. Real-time detection needs more accurate predictions more than ever. We believe that uncertainty is an important concept to include in 3D detection models for autonomous driving, as it can directly contribute to passenger and pedestrian safety. Since there is statistical significance, further elaborate modeling in this direction can solidify the uncertainty measures further into the network such as in probabilistic models.

10. Answers to reviewers' questions

Quantitatively on the KITTI Hard dataset our model has a 2% higher mAP for bounding box predictions indicating that our model predicts more accurate bounding boxes in the scenarios where there is high levels of occlusion and ambiguous object boundaries. We hypothesize that as distance from ego vehicle increases, the point cloud density decreases and the uncertainty in bounding box coordinates increases. This is visualized in the Figure 2 where Car A is closer to the ego-vehicle with higher point cloud density and is predicted with higher confidence (lower variance), Car B is highly occluded and has higher uncertainty measures. The effect of occlusion is more pronounced if we see that even though Car C is further behind it is in direct field of view and is predicted with lower uncertainty than Car B. Our model thus is able to predict better in more occluded scenarios.

11. Link to Code and Data

The code for the model can be found at the following GitHub link: https://github.com/achantaa/uncertainty-voxel

References

[1] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 502–511, 2019. 2

[2] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv preprint arXiv:2012.15712*, 1(2):4, 2020. 1

[3] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*, 2020. 2

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[7] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2888–2897, 2019. 2

[8] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. *IROS*, 2018. 2

[9] 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/CVF conference on computer vision and pattern recognition*, pages 12677–12686, 2019. 2

[10] Gregory P Meyer and Niranjan Thakurdesai. Learning an uncertainty-aware object detector for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10521–10527. IEEE, 2020. 2

[11] 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. 2

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[16] 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. 2

[17] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint arXiv:2102.00463*, 2021. 2

[18] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 4

[19] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

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

[21] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 2

[22] Yuanxin Zhong, Minghan Zhu, and Huei Peng. Uncertainty-aware voxel based 3d object detection and tracking with von-mises loss. *arXiv preprint arXiv:2011.02553*, 2020. 2

[23] 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. 1, 2