3D Detection for Distant Objects

Seung Wook Kim
Department of Computer Science
University of Toronto
seung@cs.toronto.edu

William Ngo
Department of Computer Science
University of Toronto
wingo@cs.toronto.edu

Juan Carrillo
Department of Computer Science
University of Toronto
juan.carrillo@mail.utoronto.ca

Abstract

Having an accurate 3D object detection model for autonomous driving is crucial for safe planning. However, object detection is especially challenging for distant objects due to sparse point cloud samples for LiDAR or Radar sensors. Moreover, it is hard to detect distant objects in camera image as they become too small to infer their identity. For this project, we extend the CenterFusion [1] model that fuses Radar and image modalities to detect objects in 3D. We investigate different ways to improve detection performances, focusing on distant objects.

1 Introduction

In the context of autonomous vehicles, 3D object detection is defined as an automated capability that enables vehicles to effectively determine the location, size, and orientation of other nearby vehicles and road users. While this problem has been studied before, most of the previous research focus on detection up to 50m from the ego vehicle. However, to ensure safety and reach level 5 autonomy, it is imperative to build autonomous vehicles that operate reliably at high speeds and can detect objects at long range. For instance, a car moving at 50MPH (22m/s) will take approximately 80m to stop; therefore, it needs to detect objects accurately at a distance of at least 100m or more.

3D object detection at long range involves additional challenges that require special considerations to ensure a competitive performance. Some of these challenges are closely related to the ability of the sensors to obtain quality data. For instance, distant objects appear significantly smaller (few pixels) in camera images and this hinders the effectiveness of visual only methods. On the other hand, objects far away contain very few returns in LiDAR and RADAR point clouds due to the inherent sparsity of these raw data streams at longer distances. Figure 1 shows the effects of sparsity in a LiDAR point cloud and the subsequent degradation in detection performance.

2 Related work

Building on the success of Deep Learning methods for 2D object detection on images [3, 4, 5] researchers have recently worked on the extension of detection methods for applications in 3D. However, 3D object detection is significantly more challenging due to the higher complexity in both the data sources and the specific performance requirements. Metrics for 3D object detection usually require proposals to have a minimum of 0.5 volumetric intersection over union when compared to ground truth boxes, this makes the task even more stringent considering the higher degrees of freedom in 3D space.

Preprint. Under review.

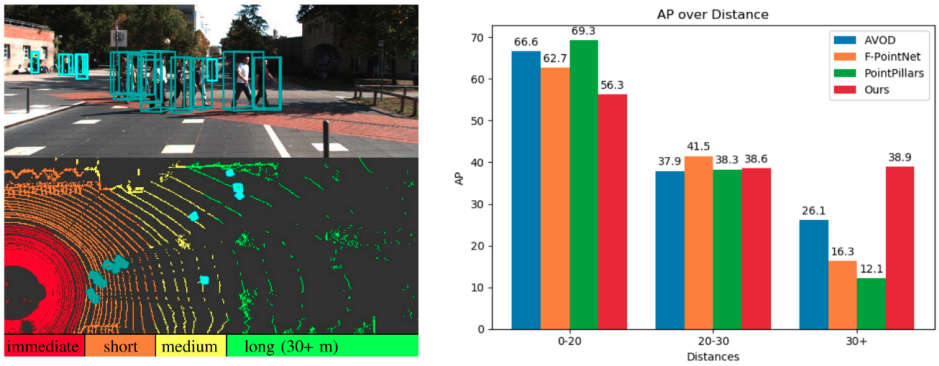


Figure 1: Left: An example of a LiDAR point cloud and how it becomes more sparse at distance. Right: Degradation in detection performance as objects are further away from the sensors. From [2].

The goal in 3D object detection is to determine accurately the pose (location, orientation) and size of objects in 3D space. Current methods for 3D object detection can be further organized depending on the sensors utilized to obtain the input data streams. Monocular methods rely only on one RGB image at a time [6] [7] while stereo methods use image pairs to obtain a better prior of the object's depth [8] [9]. However, image-only methods lack performance when compared to those using ranging sensors such as LiDAR or RADAR that provide rich 3D features directly.

As there are multiple widely used 3D representations, various models have been proposed for each representation type. Of these point cloud representations, 3D voxels and 2D projections are used to form a structured representation where classical convolution operators can be used. [10, 11] uses sparse 3D convolution operations to process 3D voxels. For 2D projection methods, point clouds are projected onto a plane, which is then discretized into a 2D image-like representations. MV3D[12] generates 3D bounding boxes from the bird's eye view representation of 3D point cloud. In contrast, learning representations from raw point clouds is also possible. Vote3Deep[13] proposes efficient voting process to compute representations from sparse 3D point clouds.

We decide to build upon *CenterFusion: Center-Based Radar and Camera Fusion for 3D Object Detection*[14] as it achieves state-of-the-art performance in 3D detection task and computationally more efficient than previous approaches. Centerfusion, as described in the next section, is a multistage framework. There have been similar approaches that also consider first leveraging 2D detection models on extracted 3D features [15, 16, 17]. However, Centerfusion is different in that it incorporates RADAR data points within a frustum defined by an image region proposal, resulting in efficient computation time.

3 Methods

We are interested in improving 3D object detection quality for driving data with focus on distant objects. For autonomous driving, it is crucial to have an accurate 3D object detection algorithm for a reliable planning model. However, distant object detection is especially challenging due to sparsity in point cloud samples. *CenterFusion: Center-Based Radar and Camera Fusion for 3D Object Detection* [14] proposes a two-stage framework in which it first extracts object proposals using CenterNet [18], a 2D object detection architecture from camera images. For each object, it predicts the center of the object along with bounding box dimensions and rotation angle to uniquely define a bounding box. Then it creates a frustum for each detected object and runs a frustum association process with the Radar point cloud to associate Radar point clouds to corresponding objects. As Radar point cloud has depth and velocity information, they can be used as an additional source of information to refine the initial bounding box predictions.

Because of its two-stage process, CenterFusion can efficiently find the center points and only use the 3D features at those locations, reducing processing time. We plan to extend the method so that it performs better for detection of objects at long range with three approaches: Data augmentation (Section 3.1), Vertex loss (Section 3.2), and Depth focal loss (Section 3.3).

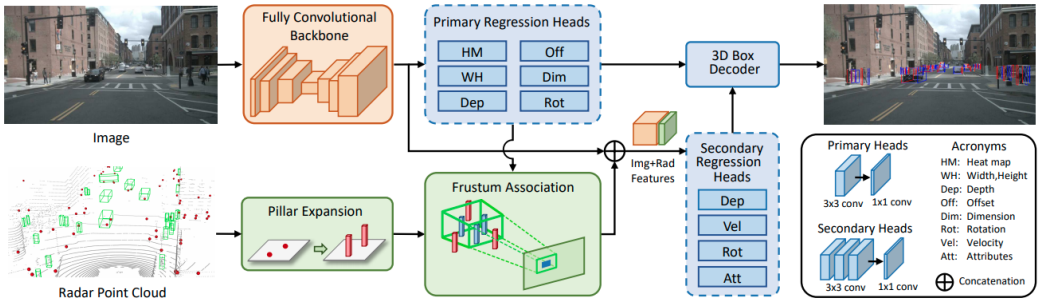
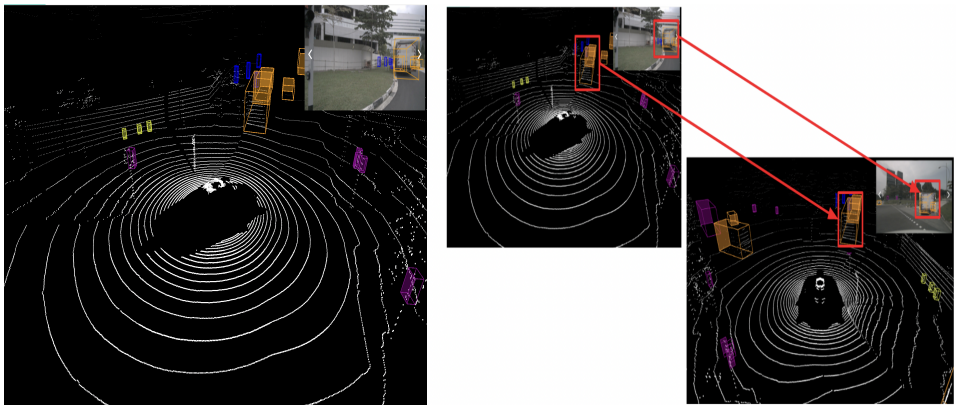


Figure 2: CenterFusion [14] model. CenterFusion infers 3D object bounding boxes with a two-stage framework that predicts the initial bounding boxes from camera images and then refines using the additional Radar information.

3.1 Data augmentation

Long-range prediction is challenging as distant objects are represented with only a few points in point cloud representation and with few pixels on an image plane, and the number of such instances is small. The usual data augmentation process in computer vision consists of distorting sensory inputs (e.g. flipping images). Because our task is object detection using image and Radar data which has small number of points, we investigate augmenting 3D objects (Figure 3). We implement a data augmentation pipeline that preserves the characteristics of the RADAR point clouds in terms of distance and sparsity, as well as resulting 3D bounding boxes that do not intersect or break real world spatial topology rules.



(a) Sample data (b) Data augmentation

Figure 3: Given 3D Radar point clouds and corresponding image, we extract object point cloud and RGB segmented object and paste them onto other scenes to increase the number of data.

The first step is to use a 2D image segmentation network to extract RGB objects. We use Mask-RCNN [5] to extract objects from images. As we do not want low-quality predictions from the segmentation network, we only keep objects if their prediction confidence is higher than 0.75 and if Intersection-over-Union (IoU) of the predicted bounding box and 2D camera plane projection of the 3D ground-truth bounding box is higher than 0.5. Let us denote $\mathcal{O} = \{o_1, ..., o_n\}$ as the set of extracted objects. For each extracted object o_i, we record the corresponding 3D ground-truth radar points inside the 3D bounding box. The radar points are in local coordinate specific to the radar sensor and the ego car. Therefore, we convert them to their global coordinates using the recorded sensor and car transformation matrices. At training time, we randomly sample extracted objects o_i and place them on training data. We first translate the point cloud of o_i to the ego car location of the training data and convert to the local radar sensor frame using transformation matrices. Then, we project the points to 2D camera plane and paste the RGB segmentation of o_i to the corresponding bounding box on the camera plane.

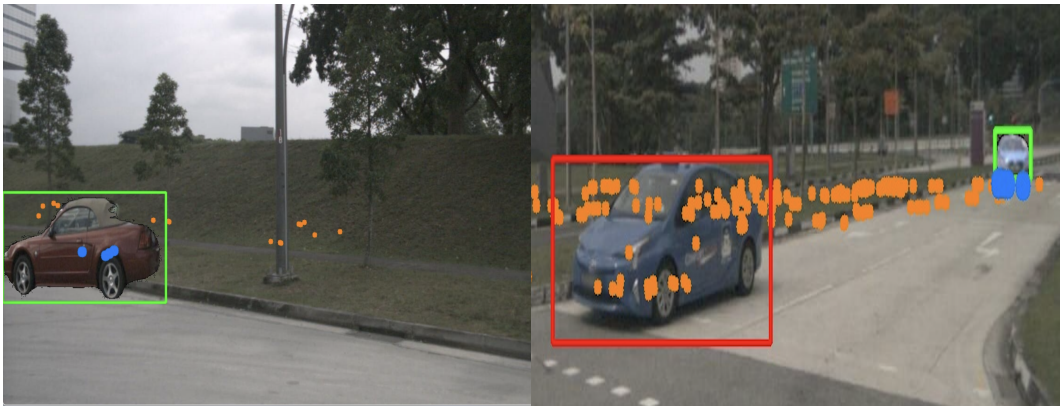
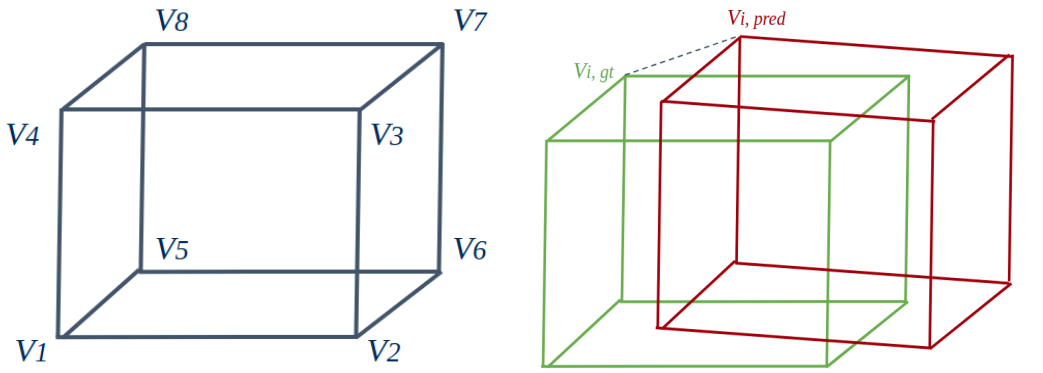


Figure 4: Examples of data augmentation: Objects inside green bounding boxes are pasted. Blue point clouds are augmented point clouds.

3.2 Vertex loss

Modern 3D object detectors predict 7 outputs to represent each bounding box $b = (x, y, z, w, l, h, \theta)$ consisting of the center coordinates (x, y, z), 3D size (w, l, h), and yaw rotation (θ). Loss functions are most commonly designed to minimize the SmoothL1 function as described in [11].

Directly predicting the offset of the yaw angle yields adverse effects in cases such as angles of 0 and π, as these two represent the same box but rotated. These cases should be penalized less than a misaligned box. Certain 3D detectors attempt to fix this by taking the SmoothL1 of sine of the offset as in [10, 19, 20]. Additionally, to address treating boxes with opposite direction as being the same, they also add a direction classifier to the output of the RPN and thus additionally predict if the orientation is correct. Other approaches introduce loss functions that combine smooth L1 along with other formulations specifically designed for training specialized architecture modules. Some of these modules refine proposals in multi stage architectures [15], incorporate a classification loss [21], resolve rotation ambiguity by a directional loss that computes sine and cosine of the yaw angle [22], or apply adaptations [23] of focal loss [24] [25].



(a) 8 Vertices of bounding box

(b) Distance between ground truth and predicted vertices

Figure 5: Vertex Loss

We propose to add an additional term in the loss function by penalizing the L_1 distance between the predicted vertices and ground truth vertices. We will denote this loss as $L_{vertices}$. For each bounding box, the vertices can be calculated using the location, size and yaw angle. For each bounding box, each vertex can be indexed, as shown in Figure. 5a. Once each vertex is indexed, we can define $L_{vertices}$ as follows:

$$L_{vertices} = \sum_{i=1}^{8} \|\mathbf{v}_{i,gt} - \mathbf{v}_{i,pred}\|_1^1$$

Where $\mathbf{v}_{i,gt}$ corresponds to the ground truth vertex and $\mathbf{v}_{i,pred}$ corresponds to the predicted vertex. Upon our literature-review, we have not seen vertex-based losses been incorporated for 3D object detection and such may lead to a better representation for bounding box localization. Calculating this term effectively measures the joint performance between the bounding box size, orientation and location.

3.3 Depth focal loss

The hypothesis we consider is that the reduction in the performance of 3D object detection models at longer ranges is in part due to the class imbalance between close and far objects in the training data. We can observe this in the nuScenes dataset [26] as objects within the dataset are largely concentrated in the 0-45m region as shown in Figure 6. In addition, the quality of the data coming from all sensors is better for closer objects; therefore, we have better features for objects close by. Since the loss functions used by most models do not consider how far the objects are from the ego vehicle, most models learn to perform better on closer objects because they are easier to characterize from the data. Moreover, since common metrics to evaluate 3D object detection methods do not dissaggregate the performance across range the effectiveness of models at long range is often difficult to determine or compare between models.

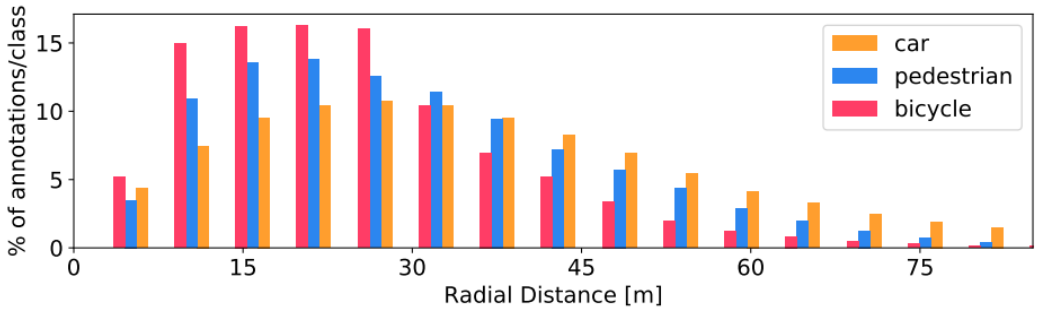


Figure 6: Data inbalance at multiple ranges in the nuScenes dataset. From [26].

The goal of the Depth Focal Loss is to force the models to perform better at longer ranges. In other words, we pick one from a family of monotonically increasing functions to obtain a weighting factor α to scale the conventional loss functions (L1, L2, etc.). Similarly, we define the concept of Range-focal-loss (RFL) if range is used instead of depth. As a consequence, the original loss functions will have slightly larger values in objects that are further away from the sensor.

We define the weighting factor α using any of the functions below; however, any monotonically increasing function will potentially serve the same purpose if configured with the right parameters. The selection of such parameters is guided by experiments in our project. Further analysis to provide stronger theoretical foundations to our Depth Focal Loss is left as future work. In our case all functions depend on depth d and have two parameters, the maximum evaluation distance m and the desired scaling factor at such maximum distance b. Both parameters are highly dependent on the specific sensors and data labeling criteria since some datasets capped the objects to a maximum range by default.

- Linear

$$\alpha = 1 + d \cdot \frac{b-1}{m}$$

- Exponential

$$\alpha = e^{\frac{d}{m} \cdot ln(b)}$$

- Logarithmic

$$\alpha = 1 + ln(1+d) \cdot \left(\frac{b-1}{ln(1+m)} \right)$$

In Figure 7 we show examples of each function using m = 100 and b = 20.

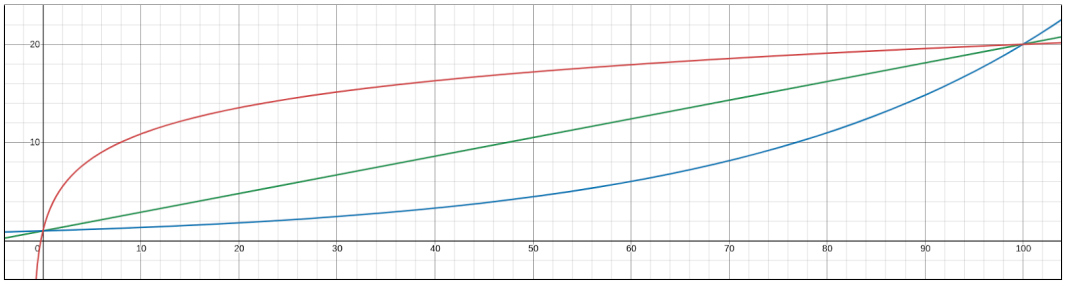


Figure 7: Depth focal loss example curves. Red: Logarithmic, Green: Linear, Blue: Exponential.

4 Experiments

4.1 Evaluation metrics

We evaluate our extensions using the 3D object detection metrics used on the nuScenes detection task namely the 3D IoU-based box AP and the nuScenes Detection Score (NDS) [26]. We split our evaluation by partitioning the objects into 3 different categories by the depth-range of the ground-truth object (0-25m, 25-50m, 50-75m). It is worth noting that in the nuScenes detection task, all ground truth and predicted detections beyond 50 meters are not considered and removed due to the combination of difficulty and lack of instances, but we did not do this as to observe model performance at long-range.

4.2 Implementation Details

We trained and evaluated using 70 training scenes and 15 evaluation scenes (10% of the full nuScenes dataset) since we do not have the same computational capacity as the original paper. As in the original *CenterFusion* paper, we start with a CenterNet model that is pretrained for 140 epochs on the nuScenes dataset and train for an additional 60 epochs. We evaluated the different weighting scheme for the depth focal by using a default scaling factor of 4 and a maximum distance of 100m. For the exponential weighting scheme, a scaling factor of 2 and 8 was also tested to observe the sensitivity of this parameter. Due to the lack of time, we did not vary the scaling factor for the other weighting schemes nor did we vary the maximum distance parameter. For the data augmentation, we tested the performance of cropping 1 or 3 additional object onto the image/scene. As we only train using 70 training scenes and 15 evaluation scenes, our baseline Centerfusion model cannot be directly compared to the model they report in their paper.

4.3 Data Augmentation Results

When observing mAP in Figure 8 and Table 1, the data augmentation method performs better than the the baseline at ranges 0-75m, beating out the baseline significantly at the 0-25m range, but performing worse in both the 25-50m and 50m-75m. As the data augmentation method yields better results in 0-75m and 0-25m on the evaluation set, but worse on 25-50m and 50-75m, we know that the distribution of object depths is heavily concentrated in the 0-25m range.

As the random images that were selected to be cropped were the ones that had a confidence score above 0.75 and the 3D IoU is above 0.5 from Mask-RCNN, we can assume that most added objects are ones will a low depth value, as naturally Mask-RCNN would be more confident in large and near objects compared to small and far objects. As the distribution of nearby objects is increased, we can assume that the network learned better representation for objects of close range, and focused less on far objects which explains our evaluation results.

4.4 Vertex Loss Results

In Figure 8 and Table 1 we can see that the vertex loss yields better results in each of the 3 range bins. This is indicative that the network is learning based on a joint representation between the

bounding box size, orientation and location, namely the vertices. These results show a simple method to improve 3D detection models as the vertex loss term term can be added to any instance level loss, given that the network predicts a bounding box size, orientation and location.

4.5 Depth Focal Loss Results

Similarly, we observe that the Depth Focal Loss is sensitive to both the choice of the weighting function and the scaling factor. We can see that using a exponential weighting scheme with a scaling factor of 2 yields our best model, but when using an exponential weighting scheme of 8, the model performs very poorly. As shown in Fig. 7, when the scaling factor and maximum depth is fixed, the exponential weighting scheme penalizes less the depths at closer ranges in comparison while penalizing equally as much at further ranges. It remains to test the linear and log weighting scheme with a scaling factor of 2 to understand if it is the scaling factor or the weighting function that is affecting performance.

Figure 8: mAP results across models. Units are %. An exponential weighting scheme with a scaling factor of 4 will be denoted as exp4, data-aug3 refers to cropping 3 objects onto an image, other names follow similarly.

Table 1: mAP results across models. Units are %.

model	0-25m	25-50m	50-75m	0-75m
baseline	26.95	14.38	1.19	16.35
vertex_loss	28.32	15.11	**1.71**	17.84
linear4	28.78	14.42	1.49	16.61
log4	22.22	11.23	0.77	12.12
exp2	**31.84**	**16.47**	1.55	**18.44**
exp4	23.05	14.89	0.95	14.84
exp8	2.77	1.35	0.09	1.37
data-aug3	31.29	7.39	0.66	17.67
data-aug1	26.55	8.44	0.48	16.00

4.6 General Discussion

Although we experience different results based on the changes we made, we do note that performance at long range is poor regardless. We strongly believe that this is due to the limitations of the model as CenterFusion first uses CenterNet to extract object proposals from camera images and then appends radar features with the frustrum created from each detected object. Since far images in the camera plane are very small, it is highly likely that CenterNet was not able detect these far objects, thus missing out on the 3D detection all together.

Results using the nuScenes detection score (NDS) [26] are shown in Fig. 9 and Table. 2. NDS is a weighted average between mAP and true positive metrics of different predicted values such as

velocity. Overall, we observe a similar outcome across models and combinations of parameters using the NDS metric. However, the numbers are higher across the board in this metric due to its flexibility to account for true positive detections. This is because in this case, the metric does not have a strict threshold to calculate 3D intersection over union and instead uses the horizontal distance between predicted and ground truth objects.

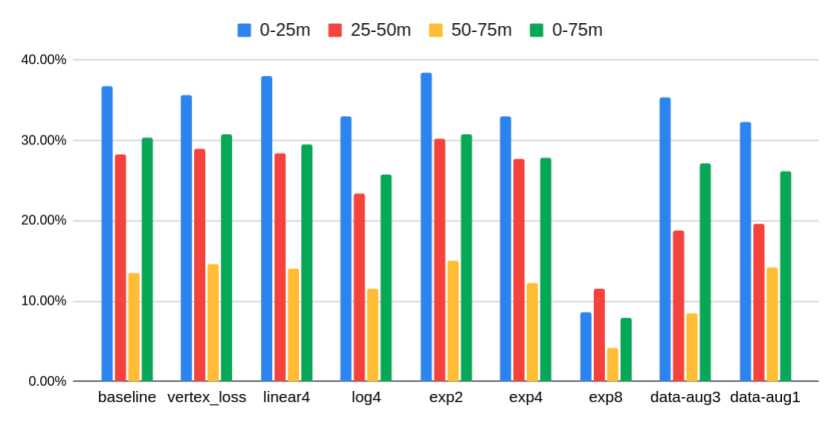


Figure 9: NDS results across models. Units are %.

Table 2: NDS results across models. Units are %.

model	0-25m	25-50m	50-75m	0-75m
baseline	36.72	28.31	13.45	30.38
vertex_loss	35.65	29.01	14.67	**30.79**
linear4	38.00	28.39	14.11	29.53
log4	32.95	23.37	11.53	25.80
exp2	**38.48**	**30.17**	**15.10**	30.77
exp4	33.01	27.74	12.23	27.79
exp8	8.59	11.51	4.12	7.89
data-aug3	35.35	18.85	8.45	27.10
data-aug1	32.37	19.64	14.15	26.14

5 Conclusions

We have shown three separate methods to try to improve performance at long ranges. Data augmentation methods have shown to improve performance at ranges where the cropped objects are concentrated. The addition of the vertex loss term seems to improve performance at every range due potentially, to the better joint representation of the bounding box size, orientation and location by the necessity of determining the vertices during training. For the depth focal loss, it seems like the results are sensitive to the choice weighting scheme and the choice of scaling factor. As the difference in performance between our models and the baseline are not significant, if given more time it would be wise to average multiple model performances across random weight initialization. In other words, training and evaluating the model over multiple seeds. We can also combine all 3 of our methods to see if they perform well jointly. To truly determine if our methods work, we will need to test them across different 3d detectors. Last but not least, as standard IoU-based AP metric is highly sensitive to small drifts for far objects, if given more time, it would be important to develop evaluation metrics that are more representative at long ranges.

References

[1] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1527–1536, 2021.

[2] M. Fürst, O. Wasenmüller, and D. Stricker, "Lrpd: Long range 3d pedestrian detection leveraging specific strengths of lidar and rgb," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, 2020.

[3] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[6] N. Gählert, J.-J. Wan, N. Jourdan, J. Finkbeiner, U. Franke, and J. Denzler, "Single-shot 3d detection of vehicles from monocular rgb images via geometry constrained keypoints in real-time," 2020.

[7] Z. Liu, Z. Wu, and R. Tóth, "Smoke: single-stage monocular 3d object detection via keypoint estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 996–997, 2020.

[8] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection," 2017.

[9] A. D. Pon, J. Ku, C. Li, and S. L. Waslander, "Object-centric stereo matching for 3d object detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8383–8389, IEEE, 2020.

[10] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[11] 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*, pp. 4490–4499, 2018.

[12] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," 2017.

[13] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1355–1361, IEEE, 2017.

[14] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1527–1536, January 2021.

[15] 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*, pp. 10529–10538, 2020.

[16] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[17] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1951–1960, 2019.

[18] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019.

[19] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," *Lecture Notes in Computer Science*, p. 720–736, 2020.

[20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2019.

[21] L. Wang, T. Chen, C. Anklam, and B. Goldluecke, "High dimensional frustum pointnet for 3d object detection from camera, lidar, and radar," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1621–1628, IEEE, 2020.

[22] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, IEEE, 2018.

[23] M. Shah, Z. Huang, A. Laddha, M. Langford, B. Barber, S. Zhang, C. Vallespi-Gonzalez, and R. Urtasun, "Liranet: End-to-end trajectory prediction using spatio-temporal radar fusion," *arXiv preprint arXiv:2010.00731*, 2020.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[25] P. Yun, L. Tai, Y. Wang, C. Liu, and M. Liu, "Focal loss in 3d object detection," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1263–1270, 2019.

[26] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.