Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection

Alex Bewley[1⋆], Pei Sun[2⋆], Thomas Mensink[1],
Dragomir Anguelov[2], and Cristian Sminchisescu[1]

[1] Google Research
{bewley, mensink, sminchisescu}@google.com
[2] Waymo LLC
{peis, dragomir}@waymo.com

Abstract. This paper presents a novel 3D object detection framework that processes LiDAR data directly on a representation of the sensor's native range images. When operating in the range image view, one faces learning challenges, including occlusion and considerable scale variation, limiting the obtainable accuracy. To address these challenges, a range-conditioned dilated block (RCD) is proposed to dynamically adjust a continuous dilation rate as a function of the measured range, achieving scale invariance. Furthermore, soft range gating helps mitigate the effect of occlusion. An end-to-end trained box-refinement network brings additional performance improvements in occluded areas, and produces more accurate bounding box predictions. On the Waymo Open Dataset [33], currently the largest and most diverse publicly released autonomous driving dataset, our improved range-based detector outperforms state of the art at long range detection. Our framework is superior to prior multiview, voxel-based methods over all ranges, setting a new baseline for range-based 3D detection on this large scale public dataset.

Keywords: Object Detection, 3D, Range Image, Autonomous Driving.

1 Introduction

Recently, catalyzed by the application towards autonomous driving, 3D object detection from pointcloud data has gained significant relevance to the broader field of computer vision. There are several meta-architectures used for addressing 3D object detection. Firstly, voxelization methods [10,34,45] typically bin sparse Cartesian coordinates into discrete voxels and later process them with subsequent 3D convolutions. While such approaches perform well in practice, they are impeded by the memory and computational demands of 3D convolutions in large scenes. A related meta-architecture is projecting the sparse points into a birds-eye view (BEV) [6,18] in order to reduce the scene back to a 2D space while maintaining scale invariance at the price of lost information through quantization. Other meta architectures are built upon the PointNet framework

⋆ Indicates equal contribution. Ordered alphabetically.

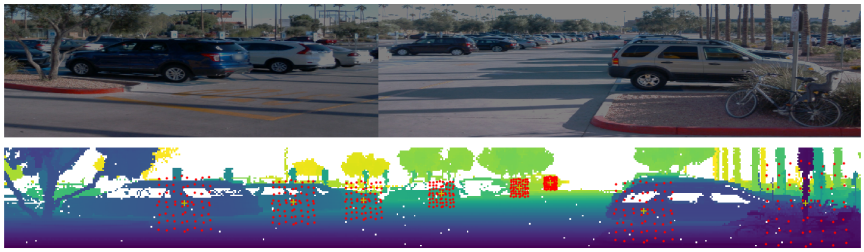


Fig. 1. Top: Reference RGB images of a scene. *For illustration purposes only.* **Bottom:** Range image showing the dynamic sampling of our proposed RCD layer at selected positions. Here the measured range at the yellow + is used to govern a scale of the local receptive field towards a geometrically consistent sample density at any range.

[24,25] but share similar issues around point sparsity and often require inefficient custom operations for defining point neighborhoods.

This paper focuses on an a less explored alternative (in the context of 3D detection) that exploits the intrinsic 2.5D manifold structure [1,37] of raw 3D point data in its native spherical coordinate form for efficient and direct 3D object detection. This *range image* representation is characterised as 3D Cartesian points projected onto unique pixels in a 2D image, where their range is encoded into the pixel values and their row and column indices correspond to inclination and azimuth angles respectively.

Operating on range images enjoys the benefits of applying mature 2D convolutional architectures and is naturally efficient due to the intrinsically compact representation [21]. Furthermore, it does not suffer from the issue of sparsity at long range. However, known challenges for learning such as scale variation and occlusion need further consideration. This paper addresses these issues by proposing a novel convolutional layer with a scale aware dilation rate for efficient reuse of filter weights at different scales. This directly leverages the measure distance in range images to compensate for the corresponding scale change. Combined with a soft range-based gating, both scale variation, and occlusion are appropriately handled within this framework. Occlusion is further addressed through the use of a second stage local box proposal refinement module.

In this work, we present an efficient range image-based two-stage 3D object detector. In particular, our key contributions can be summarized as follows:

1. a novel range conditioned dilated (RCD) convolutional operator is introduced that is capable of dynamically adjusting the local receptive field using the measured distance, to provide a consistent scale relative to the convolutional kernel at any distance (see Figure 1);
2. a region convolutional neural network (RCNN) based second stage network is investigated in the context of range image-based 3D object detection;
3. a new baseline is set for range image-based 3D object detection on a public dataset. The introduced RCD based model performs especially well at

long ranges (large distances), where voxel and sparse-point cloud based approaches suffer from point sparsity issues.

2 Related Work

2.1 3D LiDAR Detection

While many works combine color images with LiDAR data [23,17,36], here we restrict our review to works that only process 3D LiDAR data.

Voxel based methods. A popular way to do 3D object detection is to first project the points to birds-eye view (BEV) and constructs a 2D multi-channel image. The image is then processed by 2D CNN to get either BEV or 3D boxes. The transformation process is usually hand-crafted, some selected works MV3D [6], PIXOR [39], Complex YOLO [31]. VoxelNet [45] divides the point cloud into a 3D voxel grid and uses a PointNet-like network [24] to learn an embedding of the points inside each voxel. PointPillars [18], a compute efficient method, that divides the point cloud into 3D pillars and then extracts features similar as VoxelNet [45].

Point based methods. Another paradigm of methods are point based detection. It processes the raw point cloud with point cloud feature extraction methods like PointNet++ [25], Sparse Convolution [12], and then regresses 3D boxes in either downsampled BEV view or 3D point view directly. Some representative works PointRCNN [29], PVRCNN [28], STD [40]. We benchmarked our method against PVRCNN, which was evaluated on the KITTI dataset as well as the Waymo Open Dataset.

Range image based methods. Being the native representation of 3D LiDAR sensor data, the range image is compact and does not suffer from sparsity related issues which is the main challenge when developing 3D algorithms. This data representation is expected to become more popular when the LiDAR hardware improves its resolution. For example, the Waymo Open Dataset [33] releases its LiDAR data in range image format. This representation is not explored enough because a) range image based detectors require more data to train which is observed by LaserNet [21] as well. b) it is hard to generate high quality range images without knowing raw sensor information such as laser scan pattern, relative position at each laser shot. Both of these are addressed by the Waymo Open Dataset [33]. A representative work is LaserNet [21] which benchmarks on a private dataset. Range image based detection algorithms need to deal with scale variance (near range objects are larger) and occlusions. This paper presents a range image based detector that addresses both these problems.

2.2 Depth Adaptive Convolutions

The seminal work of Jaderberg *et al.* on transforming the input signal using spatial transformers [15], has led to several subsequent methods on learning the behaviour of convolutional layers. For example, for dilated convolutions, the dilation rate could be learned per filter and layer [14], or related to the size of

the 3D environment using RGBD as input [7]. Wang *et al.* [35] use depth as a constant weighting factor of the influence of neighboring pixels in convolutional layers and max-pooling layers. More generically, however, dynamic functions could be learned to generate the weights of the convolutional layer conditioned on its input values [8,32,46,9]. Ding *et al.* [9] propose a dynamic filter framework where multiple convolutions are simulated with integer shifts reflecting a dilatated convolution, these are then conbines using weights from RGB image features. Here fixed set of dilations are chosen a priori similar the atrous spatial pyramid pooling (ASPP) in DeepLab [5].

In contrast to these approaches, our method takes advantage of the available range observations to rescale the dilation rate of 2D convolutions. By doing so, the same kernel weights can be reused across multiple scales. This can be viewed as an extension of the 1D distance based input filtering of Beyer *et al.* [2] to 3D pointclouds represented as a range image. Furthermore, our sampler is continuous and adapts through the course of training mitigating the need to select a set of fixed dilation rates.

3 Methodology

In this section the proposed detection method is described. An overview of the method is shown in Figure 2. The entire detection pipeline can be broken down into two stages. The first stage consists of our proposed RCD convolutional block with a general CNN backbone and detection heads *i.e.* the foreground classification head and a pixelwise box parameter regressor. Boxes from the first stage are passed into the second stage which performs refinement to produce the final detections. This is similar in spirit to the two stage detectors used in sparse 3D convolutional networks [29] and the voxel-based ones [30].

3.1 Input Range Image

The input range image format follows what is provided by the Waymo Open Dataset (WOD) [33] with rows corresponding to LiDAR beam inclinations and columns corresponding to laser shot azimuth. The range image pixels encoded channels include: range, intensity, elongation [27,33], inclination, azimuth, x, y, z. Most of these channels are defined clearly in the WOD [33]. The channels x, y, z are the Cartesian coordinates of each point in the vehicle frame.

3.2 Range Conditioned Dilated Block

Figure 3 shows and overview of how the range conditioned dilated convolutions are combined with 1×1 convolutions [19] to form a key feed-forward block. The key unique part of this block is that the spatial convolution is replaced with a sampling which does not adhere to a regular grid/kernel but is both sparse and local, in a similar fashion to Deformable Convolution Networks [8,46].

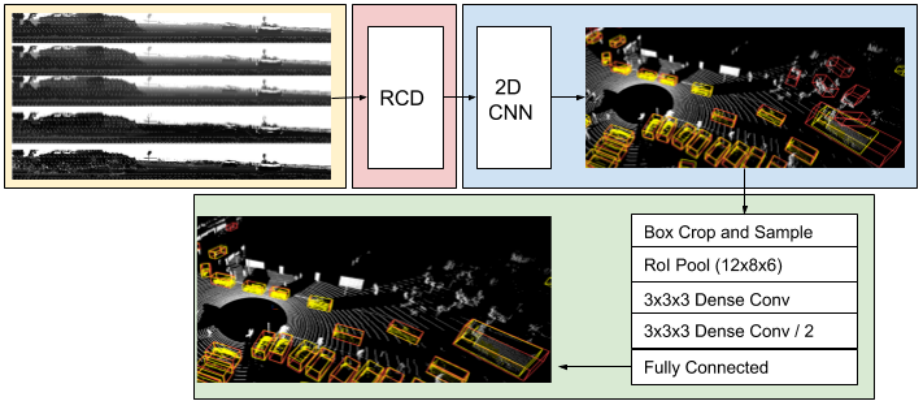


Fig. 2. Network architecture overview. The input range image is of size $64 \times 2650 \times 8$. The input is first passed to a range conditioned dilation layer, and a 2D convolution net. This first stage (in blue) generates 3D box proposal for each point with corresponding classification score and likelihood of representing the same box as neighbors. High scoring proposals are processed and then passed to an RoI pooler to divide each proposal to a $12 \times 8 \times 6$ grid. Then the second stage (in green) applies 3D convolution and finally a fully connected layer predicts a 3D box and a score.

To account for the scale variation observed when viewing objects at different distances, we introduce our Range Conditioned Dilation (RCD) operation that dynamically scales the spatial extent of a convolutional kernel using the measured range. In general a single convolutional filter could be seen as a weighted sum over neighbors around location i in the input:

$$y_i = \sum_{j \in \mathcal{N}_i} \boldsymbol{w}_j^\top \boldsymbol{x}_j, \tag{1}$$

where $j \in \mathcal{N}_i$ denotes the jth neighbor of i, $\boldsymbol{x}_\cdot \in \mathbb{R}^C$ denotes the C dimensional input features aligned with the neighborhood window $\mathcal{N}_i$, and $\boldsymbol{w}_j \in \mathbb{R}^C$ denotes the weights for neighbor j. For standard 2D convolutions, $\mathcal{N}$ forms a regular grid, commonly 3×3 consisting of the reference point and immediate neighbors. The distance from the reference location to the neighbors can be increased with integer increments to form a dilated convolution [5,41] with a static receptive field at all locations.

In contrast to normal (dilated) convolutions with a regular grid sampling, our proposed convolution generalizes the notion of a neighborhood $\mathcal{N}$ by performing a sparse point sampling of the input feature map. Specifically, for the context of processing LiDAR range images, the distance between the reference point and neighbors is conditioned on the range at location i. Specifically, we employ the use of bilinear sampling, using a sampling kernel pattern centered at each pixel location. This is equivalent to having a Spatial Transformer Network [15] at every pixel that is constrained to only adjust the spatial scale of the sampling pattern, based on the range imagery as input for these transformations, independently at

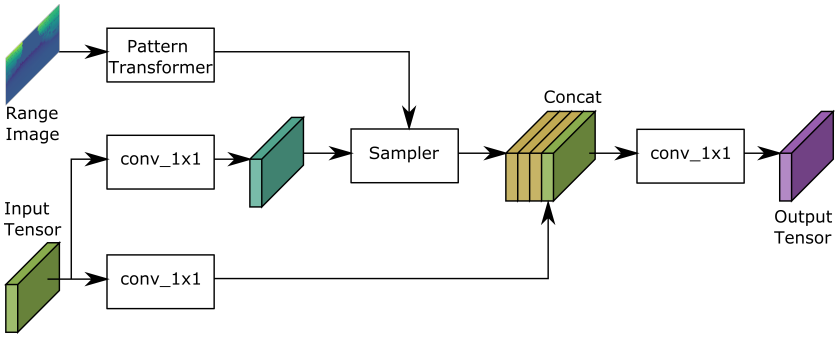


Fig. 3. An overview of the range-conditioned dilation module detailing the interaction between the various modules. The sampler is solely responsible for the spatial processing of the input tensor where the receptive field is driven by the input range image. Further details are described in text.

each pixel location, as illustrated in Figure 4. The proposed RCD module relates to the exhausted dilation combination of differently sized receptive kernels in PSPNet [43] and ASPP [5]. The key difference is instead of merging multiple scales together, RCD reuses the same set of filter weights over a continuum of dilations scaled according to the measured range and a learned nominal width parameter.

While initialization in this work follows the canonical pattern of a conventional convolutional kernel, it also generalizes to any spatial initialization scheme, *i.e.* Gaussian distributed points. After initialization, the relative spatial arrangement is free to deform as gradients are passed through the bilinear sampling process throughout training, similar to Spatial Transformer Networks [15].

Pattern Transformer Given that the input tensor to the RCD block is spatially arranged in angular space, we use the trigonometric relationship between the physical size of objects and the distance to the object, in order to set the amount of dilation to apply. Specifically, assuming a learnable nominal physical width of a (part of an) object λ, the spatial extent on the polar range image in angular space is:

$$\sigma(r_i, \lambda) = \arctan(\lambda / r_i), \tag{2}$$

where r_i is the measured range at the pixel or beam indexed at the ith pixel location. Here, λ is a learnable parameter shared across filters for a given layer. This trigonometric scaling results in a consistent sampling of an object with fixed size at different distances from the sensors. Figure 4 illustrates how the nominal width λ influences the rate of scale change for the kernel sampling. The *pattern transformer* component of the RCD takes in the range image and a shared sampling pattern to produce a dense tensor where each pixel location contains an independently scaled version of the 2D sampling pattern following

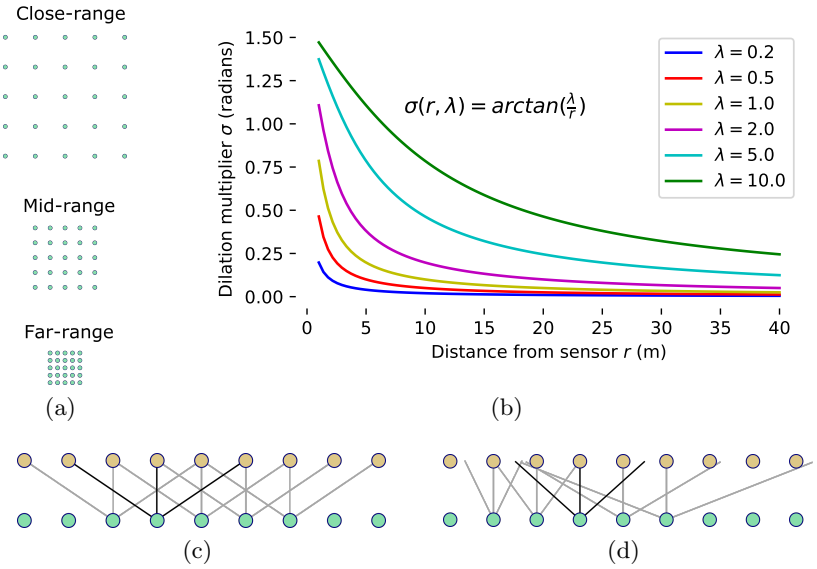


Fig. 4. (a) Illustration of dilated sampling at different ranges. (b) The rate of the convolutional filter dilation as a function of both the distance and the nominal object size λ. (c) A reference for a traditional convolution with a discrete dilation rate (here set to 2) while (d) illustrates our continuous dilation rates used by our range-conditioned-dilated convolution going form narrow dilation to wide, which corresponds to a change in range from far to near, respectively.

Eq. 2. The absolute 2D pixel coordinates is then added to each sample to shift each sampling pattern to center at their corresponding image location. Samples extending beyond the top and bottom range image boundaries are clamped to the first and last row respectively. For samples beyond the left and right boundaries, we take advantage of the 360° of the range image and perform a horizontal angular wrapping to the opposite side. This is implemented via the modulo operation with the image width to recover the wrapped absolute horizontal image coordinates.

Pointwise Convolutions and Sampler Before performing any spatial sampling, the input tensor is reduced to three channels using a standard pointwise (1×1) convolutional layer. This reduction effectively creates an information bottleneck to encourage the targeted learning of spatially discriminative features. The Sampler then performs bilinear sampling to produce a tensor with $N \times 3$ output channels where N is the number of samples. This sampled tensor is concatenated to the output of a pass-through branch (bottom path of Figure 3) comprised of a separate 1×1 convolutional layer aimed at maintaining pointwise information. Finally, another pointwise convolution is performed on the concatenated tensor. This last pointwise convolution is essentially an inner

product operation with a vector of weights and the flattened spatial sampling and pass through features to complete the RCD convolutional operation.

Soft Range Gating The aim of our RCD layer is to ensure that we capture the appropriate receptive field size for objects at any distance. However if in the perceptive view there is a significant change in disparity with neighboring pixels then the filter response could be adversely affected, creating difficulties for downstream training. While CNNs implicitly handle occlusions, we seek to make explicit use of the range to suppress features at significantly different distances relative to the reference pixel at the center. This is achieved through a Gaussian weighting using the relative distance or more formally weighting a sampled offset i with mask $m_i = \frac{1}{\sqrt{2\pi\gamma^2}} e^{-\frac{1}{2}\left(\frac{r_i - r_c}{\gamma}\right)^2}$, where r_c is the range at the sampling pattern and γ controls the width of the soft range gate (SRG).

3.3 Backbone 2D Network Architecture

Following on from the initial RCD block applied to the input, this section describes the main backbone network for feature extraction (referred to as 2D CNN in Figure 2). Inspired by LaserNet [21], we leverage the fully convolutional deep layer aggregation network architecture [42] to extract features after RCD. Unlike LaserNet, we downsample more aggressively to achieve larger receptive field and faster compute. The backbone is illustrated in Fig. 5. All pooling operators are performed along the horizontal axis only, *i.e.* using a pooling kernel like $[1, \cdot]$. This exploits better the structure of the range image, which has only 64 rows and 2650 columns. We also adopt the resnet bottleneck design [13] in the architecture which gives better performance and uses less compute.

Variations in RCD layers. RCD layers are not constrained to be used on the input only. In principle, any convolutional layer in the network could be replaced by a RCD layer. However, it is expected that features aggregation at a beginning of a block will have most impact. Hence several variants of our network are compared, where RCD layers always replaces a convolutional layer. We use the following abbreviations for different RCD variants:

- RCD_I: the first conv layer, before entering the backbone, is replaced by RCD;
- RCD_b: to replace the first conv layer in the b-th ResNet block ($b \in \{1, 2, 3\}$);
- RCD_O: when a RCD layer is placed after the backbone network.

The RCD layer always replaces a standard conv layer, and uses the same number of filters: 64 for RCD_I and RCD_1, 128 for RCD_2, and 256 for RCD_3 and RCD_O. Most of the experiments are performed with RCD_I.

3.4 RCNN

Soft range gating alleviates the occlusion effect. In order to further mitigate the occlusion effect, we add an RCNN stage similar to faster RCNN [26], point

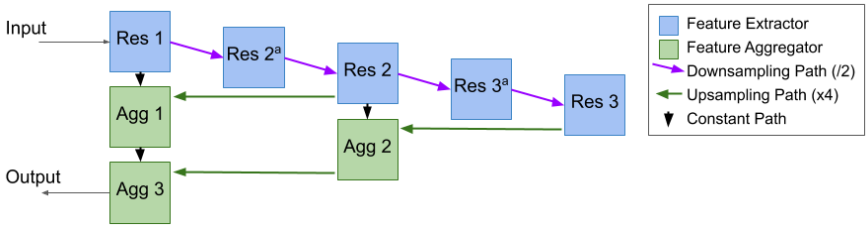


Fig. 5. 2D CNN backbone after the initial RCD layer. Number of bottlenecks units in each block: (Res1: 5), (Res2,Res2a: 7), (Res3,Res3a: 9), (Agg1,Agg2,Agg3: 4). The bottleneck is the one described in [13] with bottleneck depth set to the bottleneck input channel size divided by 4. The output channel size from each layer are 64, 128, 256.

RCNN [29], or PartA2 [30]. A second stage RCNN also greatly improves box prediction accuracy. See Figure 2 for the high-level architecture. In this two stage setup, the first stage, the RCD layers and the backbone, acts as a regional proposal network (RPN). Proposal boxes from this first stage form the input to the RCNN. For each box, we crop the raw points and also reuse processed feature embedding, predicted box parameters and semantic classification scores from the RPN. All the points cropped are transformed to the canonical box frame [29]. Then we divide the cropped boxes into a grid of $12 \times 8 \times 6$. All the features in each grid cell are pooled. Average pooling is applied on semantic features (points, classification score and RPN box parameters). Maximum pooling is applied on the feature embedding. We then apply 3D convolutions on the cropped data and finally a fully connected layer on the downsampled cropped box (size $6 \times 4 \times 3$) to generate final box parameters and a classification score. We adopt the bin loss proposed in point RCNN [29] for heading prediction. Other box dimensions are predicted based on residual loss [26].

During training, it is important to sub-sample boxes from the first stage to have an efficient training pipeline. Initially, the range image is divided into a top and bottom half, where the proposals corresponding to the highest scoring pixel for each column in the two (half) range images are kept. This coarse partitioning of the range image help maintain spatial diversity for training the RCNN network. Finally, the proposals are further reduced to 50 positive and 50 negative proposals from the 2650 (range image width) $\times$ 2 pixels. A proposal is positive if its intersection over union with its corresponding ground truth is greater than 0.5. Otherwise it is a negative proposal. During inference, we select top 400 boxes after running non-maximum-suppression on the RPN output.

3.5 Joint Training

Both stages are trained jointly where RCD layers in the RPN network and the RCNN network as the second stage are optimized using the following losses.

RPN Loss Following a similar loss weighting to LaserNet [21], each point in the range image a foreground classification score, and two additional scores representing the probability its top and left neighbors lay on the same object. As a training objective, we employ focal loss [20] to handle class imbalance and combine as follows:

$$L_{\text{cls}}^{\mathbf{rpn}} = \frac{1}{P} \sum_i (L_i + L_i^{\text{top}} + L_i^{\text{left}}), \tag{3}$$

where P is the number of valid points on the range image. L_i is the focal loss for foreground class classification, L_i^{top},L_i^{left} are the focal losses for top and left similarity classification.

For each point in the range image, we predict a 3D box by using the bin loss [29] for regression as follows:

$$L_{\text{reg}}^{\mathbf{rpn}} = \frac{1}{N} \sum_i \tfrac{1}{n_i} L_{\text{bin}}^i, \tag{4}$$

where N is the number of boxes, n_i is the number of points in box that contains point i. L_{bin}^i is the bin loss for the box proposal at point i. It divides the distance to the center point (x, y, z) and the heading of the ground truth box into bins and performs bin classification first, followed by a regression within each bin. Refer to [29] for details on this bin loss.

RCNN Loss The second stage refinement is optimized using cross entropy loss for box classification, a residual loss similar to [26] for box centers and box dimensions, while heading regression uses the bin loss [29]. These losses are aggregated over proposals as follows:

$$L_{\text{cls}}^{\mathbf{rcnn}} = \frac{1}{M} \sum_j L_j, \tag{5}$$

$$L_{\text{reg}}^{\mathbf{rcnn}} = \frac{1}{M} \sum_j L_{\text{box}}^j, \tag{6}$$

where M is the number of proposals (100 in training, 400 in inference) selected from RPN, L_{box} is the RCNN box regression loss which is a sum of smooth $L1$ loss towards ground-truth center x, y, z prediction, smooth $L1$ loss towards normalized length, width and height prediction and a bin loss for heading prediction.

Total Loss The total loss is sum of the losses above with equal weight from the individual losses as follows:

$$L = L_{\text{cls}}^{\mathbf{rpn}} + L_{\text{reg}}^{\mathbf{rpn}} + L_{\text{cls}}^{\mathbf{rcnn}} + L_{\text{reg}}^{\mathbf{rcnn}}. \tag{7}$$

4 Experiments

In this section, we introduce the implementation details of our proposed method, our experiment setup, training and inference details, comparison with other methods, and some RCD architecture search studies.

4.1 Datasets

We primarily benchmark on the challenging Waymo Open Dataset (WOD) [33] as it released its raw data in range image format while other autonomous driving datasets such as KITTI [11] or nuScenes [4] do not. While reconstructing a range image form a pointcloud is possible, it requires known laser angles and accurate pointwise timing to offset for the relative vehicle pose when in motion. WOD is also chosen since it is sufficiently large and suitable for training a range image detector as LaserNet [21] reported that small datasets like KITTI are prone to overfitting.

WOD provides information collected from a set of sensors on an autonomous vehicle, including multiple LiDARs and cameras. It captures multiple major cities in the U.S., under a variety of weather conditions and across different times of the day. The dataset provides a total number of 1000 sequences. Specifically, the training split consists of 798 sequences of 20s duration each, sampled at 10Hz, containing 4.81M vehicle and 2.22M pedestrian boxes. The validation split consists of 202 sequences with the same duration and sampling frequency, containing 1.25M vehicle and 539K pedestrian boxes. The effective annotation radius is 75m for all object classes. For our experiments, we evaluate both 3D and BEV object detection metrics for vehicles on the WOD validation set and 3D detection metrics using the public evaluation server for the test set.

4.2 Baseline Methods

The full RCD model is compared to an equivalent two-stage detector baseline without SRG and RCD layers replaced with a fixed dilated convolution (set to a dilation rate of 3 which is shown to have best RPN performance). See Table 6 in Appendix. Furthermore, a selection of state of art methods from each mainstream category in 3D object detection algorithms are compared on WOD.

LaserNet [21]: LaserNet is a 2D CNN-based singleshot 3D object detector operating on LiDAR range-images. It showed improvements on a private dataset which we don't have access to. While there is no publicly available implementation for this method, we use a variant of our first-stage sub-network with normal 2D convolutions and ResNet blocks following Figure 5, trained using multi-model box regression loss and adaptive NMS as described in [21].

Point Pillars (P.Pillars) [21]: Another single stage detector which utilizes PointNets [24] to encode a pointcloud scene representation organized in vertical columns in the BEV. Metrics on the WOD validation set come from [44].

Multi-View Fusion (MVF) [44]: This method fuses Cartesian view features and spherical view features. It shows significant improvements on long

	BEV AP (IoU=0.7)				BEV APH (IoU=0.7)			
	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$
P.Pillars [18]	75.57	92.1	74.06	55.47	-	-	-	-
DynVox [44]	77.18	93.04	76.07	57.67	-	-	-	-
MVF [44]	80.40	**93.59**	79.21	63.09	-	-	-	-
PV-RCNN [28]	**82.96**	**97.35**	**82.99**	**64.97**	**82.06**	**96.71**	**80.01**	**63.15**
LaserNet*	71.19	83.94	71.42	54.49	67.66	80.69	68.38	51.44
Baseline (Ours)	78.46	91.93	76.88	61.8	78.56	91.49	76.19	60.34
RCD (Ours)	**82.09**	93.27	**80.94**	**67.23**	**81.39**	**92.84**	**80.21**	**66.20**

Table 1. Comparison using metrics in BEV for methods for vehicle detection on the WOD validation set. The best and second best results are highlighted in **blue** and **red** respectively. (*) Note that the results for LaserNet are our best attempt at reimplementing the framework described in [21] with our backbone network to provide the performance comparisons provided in this paper. The results of P.Pillars on the WOD validation set was reproduced and reported by [44]. The baseline row shows the performance of our two stage pipeline with normal convolutions replacing our RCD layer. Columns with r show breakdown of metrics by range (in meters).

range detection because of the spherical view features. We share the same findings in our method as our method is perspective only.

Point-Voxel (PV-RCNN) [28]: is a very recently proposed method that combines both PointNets [25] and sparse convolution architectures into its RPN backbone. PV-RCNN also includes a second stage RCNN refinement network.

4.3 Training and Inference

Unless otherwise stated all experiments use the Adam optimizer [16] for 350k iterations of batch-size 8 (or 17.5 epochs) with a learning rate starting from 6e-3 with a cosine decay end to end from scratch without any data augmentation. For submission to the WOD leader-board our RCD model is trained for a total of 1 million iterations.

4.4 Discussion of Results

Our main experiment is the comparison of the proposed RCD network, using RCD_I, in the two stage architecture with (a) the same network without RCD, and (b) to the baseline methods on WOD. Among the baseline methods, only LaserNet [21] uses the range imagery directly, this makes LaserNet our direct comparison. The results of this are shown in Table 1 and Table 2.

We show significant improvements compared with the range-image detector LaserNet [21], even without RCNN stage as is shown in Table 4. This is mainly because of the way RCD and SRG handle scale variance and occlusion in the range image view. We show improvements when compared with all the best voxel

	3D AP (IoU=0.7)				3D APH (IoU=0.7)			
	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$	All	$r_{\leq 30}$	r_{30-50}	$r_{\geq 50}$
StarNet [22]	53.70	-	-	-	-	-	-	-
P.Pillars [18]	56.62	81.01	51.75	27.94	-	-	-	-
DynVox [44]	59.29	84.9	56.08	31.07	-	-	-	-
MVF [44]	62.93	86.30	60.02	36.02	-	-	-	-
PV-RCNN [28]	**70.30**	**91.92**	**69.21**	**42.17**	**69.69**	**91.32**	**68.53**	**41.31**
LaserNet*	52.11	70.94	52.91	29.62	50.05	68.73	51.37	28.55
Baseline (Ours)	63.60	86.34	62.30	36.24	63.06	85.96	61.99	36.62
RCD (Ours)	66.39	86.59	65.64	40.00	66.13	86.29	65.35	39.88
RCD 1M (Ours)	**68.95**	**87.22**	**66.53**	**44.53**	**68.52**	**86.82**	**66.07**	**43.97**

Table 2. Comparison of methods for vehicle detection on the Waymo Open Dataset (WOD) validation set for 3D detection with 7DOF boxes. The best and second best results are highlighted in **blue** and **red** respectively. (*) Our implementation of LaserNet [21]. Columns with r show breakdown of metrics by range (in meters). RCD 1M shows the performance with the training schedule increased to a million iterations.

	Level 1		Level 2	
	AP	APH	AP	APH
Second [38]	50.11	49.63	42.88	42.48
P.Pillars [18]	54.94	54.47	48.61	48.18
StarNet [22]	**61.68**	**61.23**	**55.17**	**54.76**
RCD 1M (Ours)	**71.97**	**71.59**	**65.06**	**64.70**

Table 3. Comparison of methods for 3D vehicle detection on the Waymo Open Dataset (WOD) test set. The values are provided by the test server and divided into two levels of difficulty where Level 1 has at least five points per ground truth object and Level 2 can have as few as a single point.

based methods reported on WOD with greatest improvements in long range. This is due to the issue around voxel sparsity for distant objects corroborating with the findings of [44]. As our method processes the range image in the perspective view it has fundamentally different characteristics compared to voxel or BEV projection based methods. As a result of this, we see it having complementary performance to the state-of-the-art PV-RCNN method.

WOD Leaderboard: Recently, an unlabelled test set was released to the public by Waymo for benchmarking 3D object detectors via an online evaluation service. Table 3 shows the performance of our best model trained for 1 million iterations compared to other published single LiDAR frame vehicle detectors on the leaderboard[3]. The public leaderboard divides the detection results into two difficulty levels based on the number of points within the annotated boxes. Level 1 has at least five points per ground truth object and Level 2 boxes may only

[3] https://waymo.com/open/challenges/3d-detection/

Method	AP	APH	R@50P
Standard 2D Convolution	52.25	50.28	59.22
Range conditioned (Ours)	54.87	52.47	**61.03**
Range conditioned with SRG (Ours)	**55.01**	**52.89**	60.83

Table 4. Performance comparison for first stage network with a single RCD or standard convolution applied to the input tensor. Average precision (AP), average precision (APH) and Recall at 50% precision (R@50P) is measured on WOD validation set after 350k iterations of training. RCD improves over using only standard convolutional layers, and adding soft range gating (SRG), improves AP and APH even further.

have a single point. Our range based RCD model significantly outperforms other methods for both difficulty levels on this public benchmark[4].

4.5 RCD Study

Effect of Nominal Width: For assessing the behaviour of the nominal width in all experiments we set it as a learnable parameter initialized to the value of 1m. Over the course of training, this parameter generally increases and then settles close to 3m which is comparable to the average dimensions of a vehicle. See Figure 7 in Appendix.

Effect of SRG: We found that the addition of a range gate improved training stability leading to faster convergence with a slightly improved performance. Table 4 shows the benefits provided to the first stage detector compared to adding a standard square kernel with different dilation rates. For a fair comparison we substitute the RCD block with a standard 2D convolutional layer with 7×7 kernel and 64 channels matching the RCD output.

Range Dilation Architecture: In this final set of experiments the placing of the location of the RCD layer(s) in the network is studied. Several variants are evaluated, at different levels of the network. The results are shown in Table 5.

5 Conclusions

We have proposed a novel end-to-end 3D object detection network, operating on native range images (e.g. LiDAR), which dynamically adjust dilation rates as a function of the locally measured range, for scale invariance. The proposed model further relies on soft range gating to mitigate occlusion and integrates an end-to-end trained box-refinement network for additional performance gains for bounding box prediction in occluded areas. An improved system relying on our new RCD block representation, and based on the two stage RCNN method [29], is the top performing range image based detection method, over all ranges, on

[4] Public results of comparison methods were gathered on May 18^{th}, 2020

	RCD_I	$RCD_{b=1}$	$RCD_{b=2,3}$	$RCD_{b=1,2,3}$	RCD_O	No-RCD
AP	**54.90**	53.29	**54.10**	52.56	53.21	53.08
APH	**52.88**	51.08	**51.97**	50.72	50.85	50.75
R@50P	**60.59**	59.33	**59.89**	59.09	59.22	59.22

Table 5. Architecture search for the location of the RCD layer. Average precision (AP), average precision weighted by heading (APH) and Recall at 50% precision (R@50P) is measured on WOD validation set after 350k iterations of training. Interestingly either having an RCD layer at the input (RCD_I), or at the intermediate layers $RCD_{b=2,3}$ is most beneficial.

the Waymo Open Dataset [33] benchmark and achieves competitive results in other tests. Specifically, our approach is state-of-the-art for detecting objects at long distances as it benefits from the dense nature of the range image. While this work has focused on single frame detection from a roof mounted LiDAR sensor, fusing data from multiple sensors, multiple frames, and multiple views are all promising directions of future research.

References

1. Bewley, A., Upcroft, B.: Advantages of exploiting projection structure for segmenting dense 3d point clouds. In: Australian Conference on Robotics and Automation (2013) 2
2. Beyer, L., Hermans, A., Leibe, B.: DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data. IEEE Robotics and Automation Letters (RA-L) (2016) 4
3. Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. The Annals of Mathematical Statistics **29**(2), 610–611 (06 1958) 19
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019) 11
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017) 4, 5, 6, 20
6. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1, 3
7. Chen, Y., Mensink, T., Gavves, E.: 3d neighborhood convolution: Learning depth-aware features for rgb-d and rgb semantic segmentation. In: 2019 International Conference on 3D Vision (3DV). pp. 173–182. IEEE (2019) 4
8. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017) 4

9. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020) 4
10. Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I.: 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) 1
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 11
12. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017) 3
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 8, 9
14. He, Y., Keuper, M., Schiele, B., Fritz, M.: Learning dilation factors for semantic segmentation of street scenes. In: German Conference on Pattern Recognition. pp. 41–51. Springer (2017) 3
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015) 3, 5, 6
16. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference for Learning Representations (ICLR) (2015) 12
17. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: 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) 3
18. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019) 1, 3, 12, 13
19. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013) 4
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017) 10
21. Meyer, G.P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., Wellington, C.K.: Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12677–12686 (2019) 2, 3, 8, 10, 11, 12, 13
22. Ngiam, J., Caine, B., Han, W., Yang, B., Chai, Y., Sun, P., Zhou, Y., Yi, X., Alsharif, O., Nguyen, P., et al.: Starnet: Targeted computation for object detection in point clouds. arXiv preprint arXiv:1908.11069 (2019) 13
23. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927 (2018) 3
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017) 2, 3, 11

25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017) 2, 3, 12
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015) 8, 9, 10
27. Shand, M.A.: Origins and mitigations of some automotive pulsed lidar artifacts. In: Jalali, B., ichi Kitayama, K. (eds.) AI and Optical Data Sciences. vol. 11299, pp. 49 – 55. International Society for Optics and Photonics, SPIE (2020) 4
28. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. arXiv preprint arXiv:1912.13192 (2019) 3, 12, 13
29. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019) 3, 4, 9, 10, 14
30. Shi, S., Wang, Z., Wang, X., Li, H.: Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. arXiv preprint arXiv:1907.03670 (2019) 4, 9
31. Simony, M., Milzy, S., Amendey, K., Gross, H.M.: Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In: The European Conference on Computer Vision (ECCV) Workshops (September 2018) 3
32. Strub, F., Seurin, M., Perez, E., De Vries, H., Mary, J., Preux, P., CourvilleOlivier Pietquin, A.: Visual reasoning with multi-hop feature modulation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 784–800 (2018) 4
33. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. arXiv pp. arXiv–1912 (2019) 1, 3, 4, 11, 15
34. Wang, D.Z., Posner, I.: Voting for voting in online point cloud object detection. In: Robotics: Science and Systems. pp. 10–15607 (2015) 1
35. Wang, W., Neumann, U.: Depth-aware cnn for rgb-d segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018) 4
36. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In: IROS. IEEE (2019) 3
37. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 1887–1893. IEEE (2018) 2
38. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018) 13
39. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018) 3
40. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1951–1960 (2019) 3
41. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR) (2016) 5, 20
42. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2403–2412 (2018) 8

43. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017) 6
44. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. arXiv preprint arXiv:1910.06528 (2019) 11, 12, 13
45. Zhou, Y., Tuzel, O.: 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) 1, 3
46. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9308–9316 (2019) 4

A Learnable Sampling Pattern Analysis

To further assess the effect of using a learnable, continuous dilated kernel sampling, Figure 6 shows the relative locations of the learnt spatial samples with their initial positions marked. Compared to the uniform sampling at initialization, inner sample points tend to increase the local concentration around the center with the outer points showing negligible or slight spread away from the central location. This indicates that the RCD model attempts to over-sample in the central region with sparser sampling in the extremities, similar to the Fovea in the human eye. This calls for further investigation into non-uniform sampling patterns as an initialization scheme (e.g. with a Box-Muller transform [3] as proposed in the main paper). Figure 7 plots the evolution of the learnable nominal width parameter λ from Equation 2. At the end of training this parameter settles at approximately 3 meters or the size of a small vehicle.

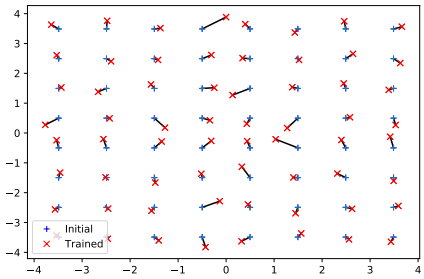


Fig. 6. Movement of a 8×8 kernel sampling pattern over the course of training. With blue '+' showing the uniform grid used to initialize offsets, and red '×' showing the pattern after training.

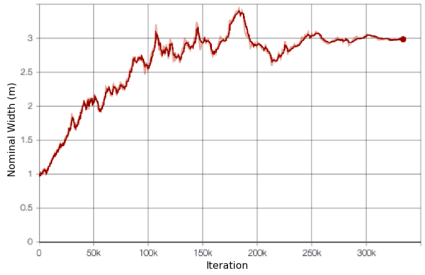


Fig. 7. Nominal width free parameter λ over the course of training. λ is used to transform the observed range to a nominal width (in meters).

Method	Dilation	AP	APH	R@50P
Standard Convolution	Fixed rate=1	52.25	50.28	59.22
	Fixed rate=3	53.75	51.76	60.05
	Fixed rate=5	53.61	51.3	59.93
	Fixed rate=7	51.58	48.06	60.23
ASPP* [5]	(6, 12, 18, G)	32.07	31.53	-
Ours	Range conditioned	54.87	52.47	**61.03**
Ours with SRG	Range conditioned	**55.01**	**52.89**	60.83

Table 6. Performance comparison for first stage network with a single RCD or standard convolution applied to the input tensor with fixed dilation rates. Average precision (AP), average precision weighted by heading (APH) and Recall at 50% precision (R@50P) is measured on WOD validation set after 350k iterations of training. (*) ASPP peaked near 53k iterations with values shown in table. RCD improves over using only standard convolutional layers, and adding soft range gating (SRG), improves AP and APH even further.

B Fixed Dilation Rates

Table 6, expands the experiments in the main paper with a more exhaustive set of dilation rates. Here we explore the effect of increasing the dilation rate of the kernel for a standard convolutional operation [5,41] in the first region proposal network (RPN). We also compare to the atrous spatial pyramid pooling (ASPP) [5] framework by replacing our RCD layer with the ASPP module. Here we used fixed strides of 6, 12, 18 and global average features denoted by 'G'[5].

With dilation rates of 3 and 5, a small improvement over the standard convolution (with default dilation rate of 1) is observed. However, further increasing the dilation results in a sharp drop in performance as seen with the dilation rate of 7. This drop in performance could be due to: insufficient sampling of small distant objects; or an increase in the amount of padding needed given the height of the range image is limited to 64 pixels. ASPP with various dilations significantly under-performed compared to the standard dilated convolutions. Having fixed and overly large dilation rates increases the susceptibility of ASPP to be distracted by sporadic activations caused by high frequency detail. The RCD models, with their ability to appropriately adjust their rate of dilation for both near and far, leads to the best overall performance.

[5] Following implementation from: github.com/rishizek/tensorflow-deeplab-v3

C Qualitative Results

Figures 8 and 9 provide a qualitative view of our detections on WOD.

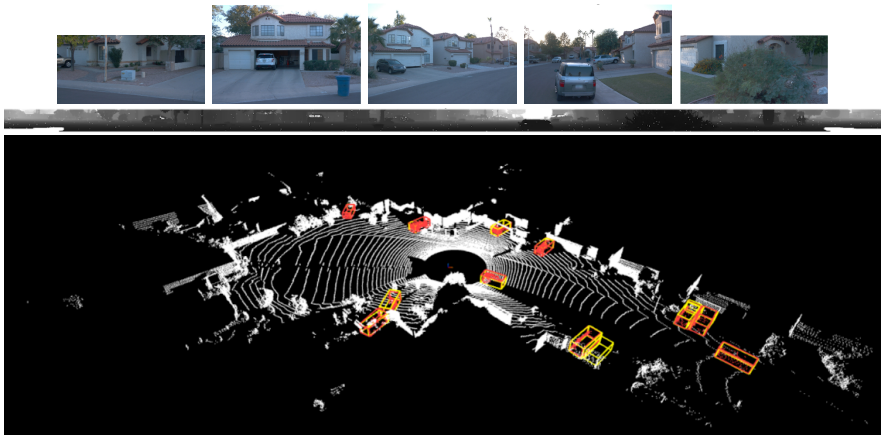
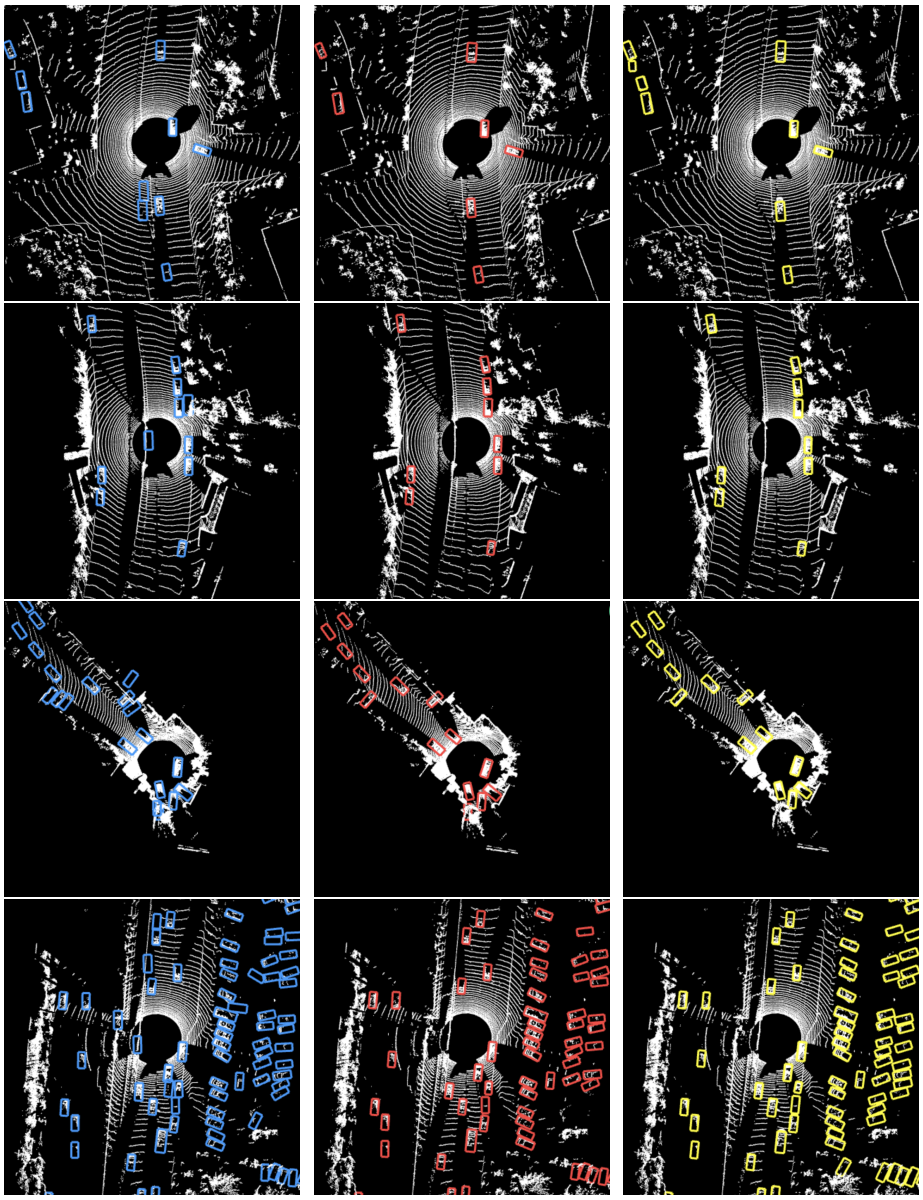


Fig. 8. Example visualization of 3D alignment. Top row shows colour images of Left, Front-Left, Front, Front-Right and Right view respectively. Middle row shows the corresponding range image for the scene. Bottom row shows the corresponding 3D point-cloud with our detections in red against the ground truth boxes in yellow. *Note the RGB images are only for illustration purposes.*



(a) Proposals (b) Detections (c) Ground Truth

Fig. 9. Birds-eye-view visualization of four different scenes from the WOD validation sequences. Each row shows a different scene and each column from left to right shows: output of the RPN in blue, output of RCNN in red, and ground truth boxes in yellow respectively. All predictions are filtered with minimum confidence of 0.5. Best viewed digitally with zoom.