Supplementary Material: Robust Multimodal Vehicle Detection in Foggy Weather Using Complementary Lidar and Radar Signals

Kun Qian[1], Shilin Zhu[1], Xinyu Zhang[1], Li Erran Li[2]
[1]University of California San Diego [2]Columbia University
{kuq002,xyzhang,shz338}@eng.ucsd.edu erranli@gmail.com

1. Overview

In the main paper, we have described MVDNet, a deep late fusion model for vehicle detection. MVDNet exploits lidar and radar's complementary advantages and achieves robust vehicle detection even in adverse foggy weather condition. Experimental results on a procedurally generated dataset show that MVDNet achieves notably better performance on vehicle detection in foggy weather condition compared with the state-of-the-art detectors [11, 5, 3, 2]. To better understand the performance of MVDNet, in this supplementary material, we provide additional evaluation results, including comparison with existing temporal fusion methods (Sec. 2), results of clear-only training (Sec. 3), effect of temporal and sensor fusion order (Sec. 5), ablation study on MVDNet-Fast (Sec. 6), additional visualization of sensor contribution (Sec. 7) and vehicle detection (Sec. 8). We further provide details of the dataset annotation (Sec. 9) and MVDNet network architectures (Sec. 10) to ease reproduction, and discuss MVDNet's limitations along with future work (Sec. 11).

2. Comparison with Existing Temporal Fusion Methods

MVDNet uses a temporal fusion network to merge historical data of the lidar and radar. As a comparison, we further report the performance of Fast and Furious (F&F) [4], a single-stage detector that fuses historical frames of the lidar. F&F develops two variants of the VGG-16 network [6] that fuses five lidar frames. The first model (Early-F&F) concatenates all lidar frames along the temporal dimension and uses a 1D convolution to reduce the temporal dimension to 1 at the early stage. The second model (Late-F&F) modifies two convolution layers of VGG-16 to perform 3D convolution and reduce the temporal dimension to 1. Tab. 1 shows the overall performance of F&F and MVDNet. MVDNet consistently outperforms F&F thanks to the use of additional radar signals. When trained with both clear and foggy data, Late-F&F statistically outperforms Early-F&F, meaning that the late fusion captures the useful temporal information better than the early fusion. When no foggy data is provided in the training sets, we observe that the Early-F&F and Late-F&F have comparable performance.

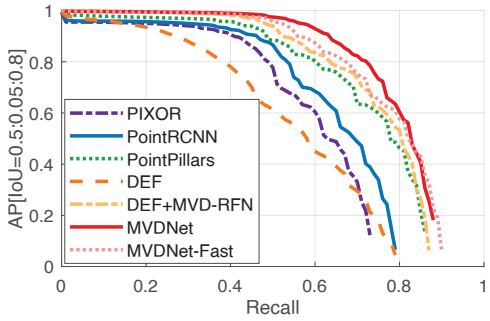


Figure 1. Precision-recall curves on clear-only training set.

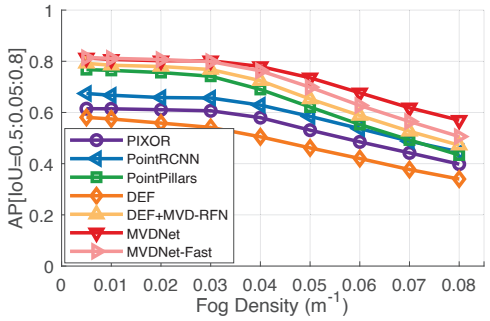


Figure 2. Impact of fog density on clear-only training set.

3. Results of Clear-Only Training

This section provides the additional evaluation results where models are trained with only clear lidar point clouds. Fig. 1 shows the fine-grained precision-recall curves of all detectors with IoU averaged over [0.5, 0.8]. When trained with only clear data, MVDNet still shows significant advantages over other detectors, verifying the effectiveness of MVDNet's late fusion design. Comparing with training sing both clear and foggy lidar point clouds (Fig. 6 in the main paper), training with only clear lidar point clouds has lower performance, which is consistent with the numeric results in Tab. 1 in the main paper.

Fig. 2 shows the AP of all detectors under typical fog densities from 0.005 m^{-1} to 0.08 m^{-1}, when the detectors are trained with only clear lidar point clouds. The results are consistent with Fig. 7 in the main paper, where the performance of all detectors drops as the fog density increases. However, without the foggy lidar point clouds in the train-

Method	Train	Clear+Foggy						Clear-only						#Params
	Test	Clear			Foggy			Clear			Foggy			
	IoU	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	
Early-F&F [4]		77.81	71.61	39.90	70.37	63.66	35.08	77.62	70.81	43.21	61.99	54.96	28.77	2,519K
Late-F&F [4]		80.64	73.73	44.48	71.63	65.00	38.09	79.50	72.16	40.17	61.47	53.04	24.90	2,607K
MVDNet-Fast (**Ours**)		88.99	86.20	68.30	85.58	82.25	62.76	**88.91**	85.96	68.15	76.30	73.97	56.96	**977K**
MVDNet (**Ours**)		**90.89**	**88.82**	**74.63**	**87.40**	**84.61**	**68.88**	87.22	**86.06**	**72.63**	**77.98**	**75.89**	**61.55**	8,591K

Table 1. Comparison with existing temporal fusion methods: AP of oriented bounding boxes in bird's eye view. Bold numbers represent the best score among all the methods.

Method	Test	Clear			Foggy		
	IoU	0.5	0.65	0.8	0.5	0.65	0.8
Radar-Only		73.04	68.27	43.25	-	-	-
Lidar-Only		86.96	85.78	72.46	77.60	75.66	63.20
MVDNet (**Ours**)		**90.89**	**88.82**	**74.63**	**87.40**	**84.61**	**68.88**
MVDNet-Reverse		88.75	85.94	71.66	85.51	82.85	66.91

Table 2. Additional ablation study on MVDNet: AP of oriented bounding boxes in bird's eye view.

ing set, all detectors experience similar dropping rates. It indicates that augmenting the clear data with foggification is crucial for robust all-weather detection.

4. Performance with Single Sensor Modality

To better understand the benefit from each sensor, we compare the performance of MVDNet and the models with single sensors on the separate clear and foggy test sets. Tab. 2 shows the AP of the models on the two test sets. First, the radar-only model achieves poorer performance than the lidar-only model, mainly because that the radar has coarser granularity than the lidar, as discussed in the main paper. Second, with the help of the radar, MVDNet outperforms the lidar-only model for both test sets. Specifically, the AP is improved by 3% for the clear test set and 8.1% for the foggy test set. It means that the benefits of radar are two folds. First, in clear weather, the radar can capture RF reflections from vehicles that are naturally occluded in the visible spectrum (e.g., vehicles behind bushes and walls) and cannot be seen by the lidar. Second, in foggy weather, the radar can see through fogs and further help the lidar detect vehicles occluded by fogs.

5. Effect of Temporal and Sensor Fusion Order

As shown in Fig. 3 in the main paper, MVDNet first fuses feature vectors of the two sensors and then fuses feature tensors along the temporal dimension. To evaluate the impact of the order between temporal and sensor fusion, we switch the sensor fusion and temporal fusion networks. Specifically, we use two temporal fusion networks to fuse region-wise feature tensors of the two sensors along the temporal dimension and then apply sensor fusion to the outputs of the temporal fusion networks. This variant of MVDNet is named as MVDNet-Reverse. Tab. 2 compares the performance of MVDNet and MVDNet-Reverse. Though MVDNet-Reverse has one more temporal fusion network, its AP drops by 2.3% on average, compared with MVDNet. The result indicates that the temporal fusion benefits from the sensor fusion, which extracts more detailed information from the two sensors via the attention-based networks for further temporal fusion. In contrast, MVDNet-Reverse compresses the information of multiple frames by early temporal fusion, which can negatively affect the granularity and capability of the attention-based sensor fusion.

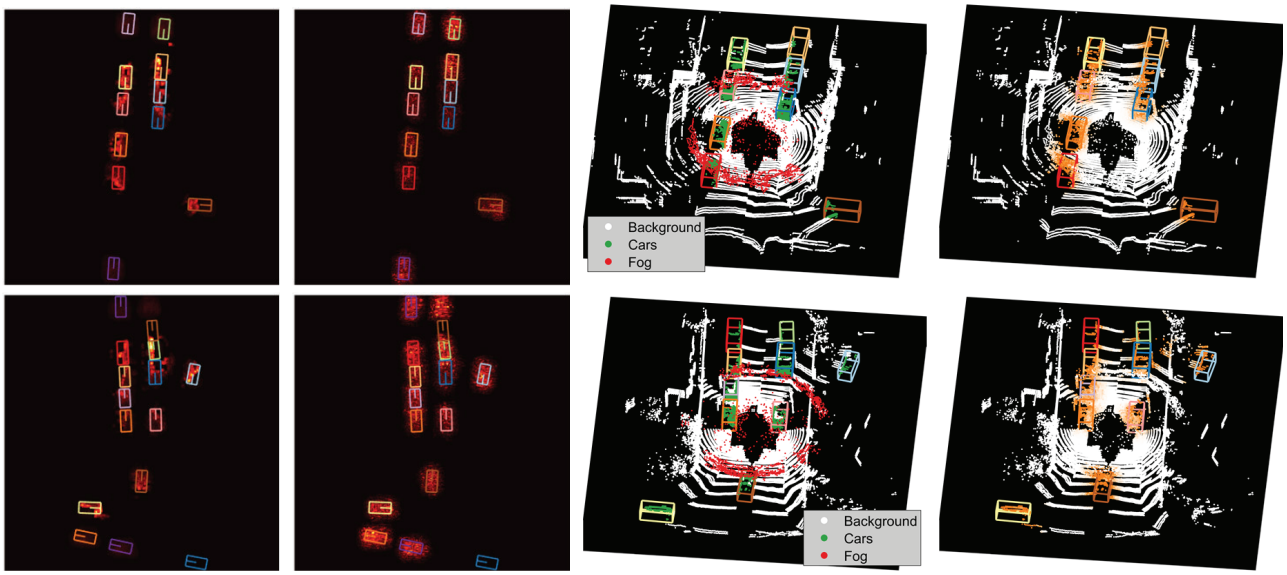
Method	IoU	0.5	0.65	0.8
Radar-Only-Fast		70.82	65.38	35.52
Lidar-Only-Fast		82.69	79.71	62.43
Lidar Reconstruction-Fast		85.28	81.47	63.45
No History-Fast		85.91	82.58	62.26
No History/Fusion-Fast		83.37	80.01	61.86
MVDNet-Fast (**Ours**)		**87.29**	**84.23**	**65.53**

Table 3. Ablation study on MVDNet-Fast: AP of oriented bounding boxes in bird's eye view (averaged over both clear and fog testing sets).

6. Ablation Study on MVDNet-Fast

This section provides additional ablation study of MVDNet-Fast. As shown in Tab. 3, we first evaluate the individual contribution of lidar and radar. Similar to the ablation study of MVDNet (Tab. 2 in the main paper), the radar-only model has a significant performance drop due to the coarse granularity and lack of height information of the radar. The performance of the lidar-only model drops by 4.1% on average, mainly due to the adverse impact of fog.

In contrast, the lidar reconstruction model takes advantage of both lidar and radar data and achieves better performance than the models using either individual sensor. The lidar reconstruction model fuses the lidar and radar data at the early feature extraction stage to reconstruct the incomplete lidar point clouds due to fog blockage. However, due to the low data quality of radar compared with lidar, the reconstruction with the early fusion is ineffective. Specifically, radar has a lower angular resolution than lidar, resulting in a severer blurry effect at far distances. For example, the high-end NavTech CTS350-X radar in our dataset uses a directional mechanic antenna to achieve a resolution of 0.9^o, while the Velodyne HDL-32E lidar has a much higher resolution of 0.33^o. Moreover, radar cannot provide accurate height information due to its limited antennas along the vertical direction. In contrast, vehicle lidar usually employs tens of vertical laser channels. Last but not least, radar suf-



(a) BEV gradient of lidar (b) BEV gradient of radar (c) Lidar 3d point cloud (d) Point-wise gradient

Figure 3. Illustration of the contribution of the lidar and radar inputs. (a) and (b) show the bird's eye view (BEV) gradients of the detected vehicles' features with respect to the lidar and radar inputs (brighter color means a larger gradient). (c) shows the foggy point clouds within the reduced visible range of the lidar. (d) shows the gradients (orange colored) of the detected vehicles' features with respect to the visible lidar points.

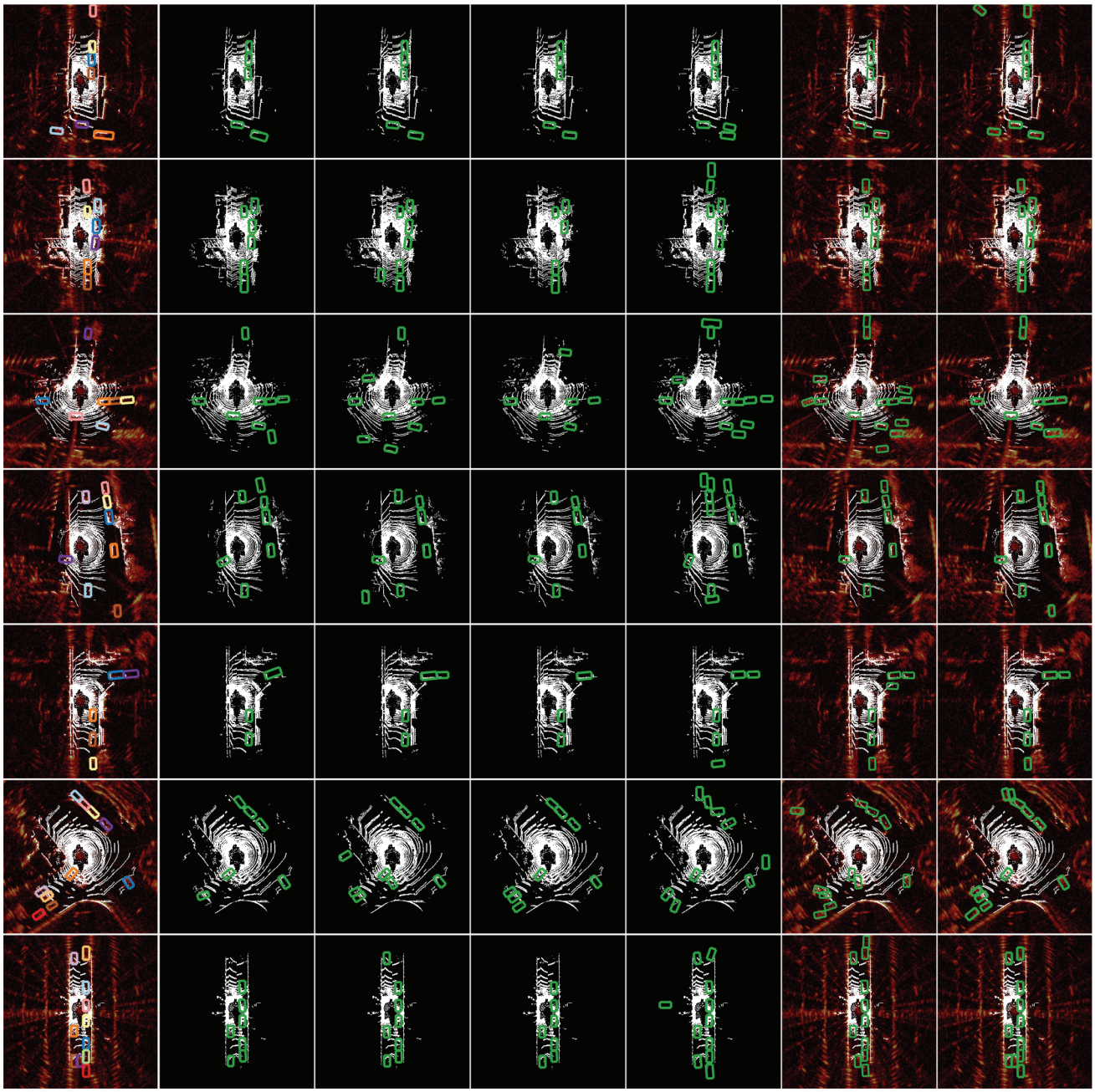
fers from noise artifacts and ghost images due to saturation and multipath effect [8]. In contrast, MVDNet bypasses the ineffective reconstruction step and fuses the region-wise lidar and radar features at the late stage, which concentrates more on the detection task and achieves a higher AP.

Besides, we further evaluate the impact of fusing historical information. First, we only use the current lidar and radar frames and implement the no-history model. The no-history model has a higher AP than the lidar-only model, which uses the historical information of the lidar. It shows that the fusion of the two sensors is critical for vehicle detection in adverse foggy weather. However, the AP of the no-history model is still 2.1% lower than that of MVDNet-Fast. It is due to the lack of historical information, which may "extend" the present visible range of lidar with the area visible in the past. Second, we further replace the fusion networks with a single convolution layer with the same input and output dimensions and implement the no-history/fusion model. Compared with the no-history model, the no-history/fusion model has a 1.8% lower AP, demonstrating the necessity of attention-based sensor fusion.

7. Visualization of Sensor Contribution

In Fig. 9 and 10 of the main paper, we illustrate the gradients of the detected vehicles' features with respect to the local inputs of the two sensors. In this section, we further provide the gradients with respect to the whole inputs to better understand the sensors' contributions. Fig. 3 shows the gradients of two scenes. Fig. 3a and 3b compare the bird's eye view gradients with respect to the lidar and radar inputs. In the first scene (i.e., the first row), the top two and bottom one vehicles are out of the lidar's visible range, while in the second scene (i.e., the second row), the top two and bottom three vehicles are out of the lidar's visible range. As a result, the gradients of these vehicles with respect to the lidar input are close to zero, indicating that the lidar has little contribution to the vehicles out of its reduced visible range. In contrast, since the radar is immune to fog, the gradients of all detected vehicles with respect to the radar input are prominent. However, there are two exceptional cases in the second scene. First, at the top of the figure, MVDNet mistakenly recognizes a vehicle, resulting in prominent gradients with respect to the radar input. Second, MVDNet fails to detect the top brown vehicle and the bottom blue vehicle, and their gradients with respect to both lidar and radar inputs are close to zero. The exceptional cases indicate that fusing lidar and radar still cannot cover all cases, mainly due to the reduced visible range of the lidar in foggy weather condition and low radar resolution at far range. Integrating a more diverse set of sensors, e.g., Doppler radar and RGBD camera, may further improve the robustness of the detector, which we leave as future work.

Fig. 3c further shows the point clouds within the reduced visible range of the lidar and Fig. 3d shows the point-wise gradients. On the one hand, most fog points contribute nearly zero gradients to the features of the detected vehicles, indicating that MVDNet is capable of denoising lidar point clouds. On the other hand, some points around the ground-truth boxes of vehicles also have prominent gradients, meaning that MVDNet additionally relies on the rep-



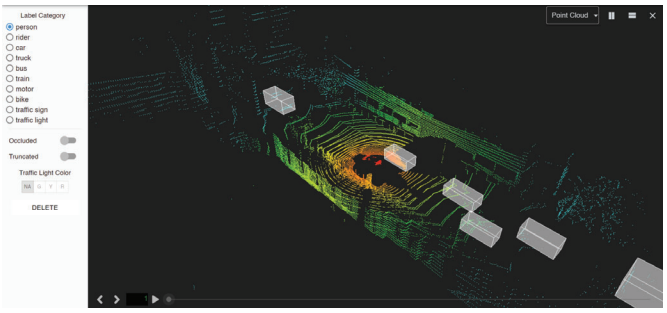
(a) Ground-truth (b) PIXOR [11] (c) PointRCNN [5] (d) PointPillars [3] (e) F&F [4] (f) DEF [2] (g) MVDNet (**Ours**)

Figure 4. Examples of 360° detection results of different detectors. The ground-truth is in various colors while the detection is in green.

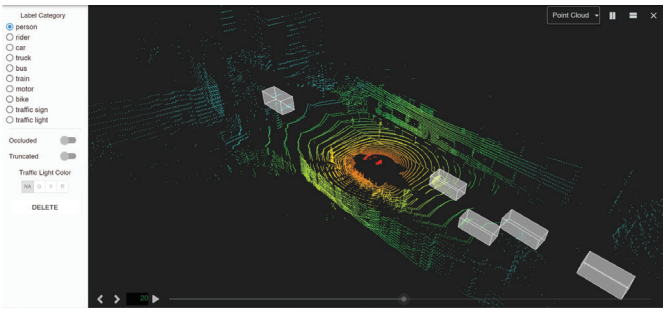
resentative surrounding background of vehicles for detection, similar to existing detectors [5] that explicitly exploit background information.

8. Visualization of Vehicle Detection

We further present additional results of different detectors in Fig. 4. Similar to the results in Fig. 11 of the main paper, all lidar-only detectors miss some vehicles and mistakenly recognize some background areas as vehicles. Among the four lidar-only detectors, F&F detects more vehicles thanks to the use of historical information. However, F&F also generates more false alarms, especially around moving vehicles, e.g., the top vehicles in the fourth scene. We think that F&F detects successive locations of moving vehicles in different lidar frames due to the symmetrical fusion of the lidar frames at the early feature extraction stage. While using both lidar and radar signals and thus detecting vehicles beyond the reduced visible range of the lidar, DEF has more



(a) Annotated keyframe #1



(b) Annotated keyframe #2

Figure 5. Illustration of two successive keyframes (separated by 20 frames) of ground-truth oriented bounding boxes (white) labeled with Scalabel [12]. For the intermediate frames, bounding boxes of vehicles are first linearly interpolated from their bounding boxes in the keyframes and then manually adjusted and verified by humans to ensure the quality of the ground-truth.

false alarms and missing targets than MVDNet, mainly due to its specialized early fusion design for front view images. In contrast, thanks to late fusion design, MVDNet consistently outperforms its counterparts in terms of detection and localization accuracy.

9. Dataset Annotation Details

We create a procedurally-generated dataset from the Oxford Robot Car [1]. To train models for foggy weather, we use the fog model in DEF [2] to randomly foggify the lidar point clouds. To generate ground-truth oriented bounding boxes, we use an open-sourced labeling tool Scalabel [12]. The interface of Scalabel and two example keyframes are shown in Fig. 5. To ensure the quality of the ground-truth labeling, we hire human annotators to generate 3D bounding boxes for keyframes first by asking them to provide reshaped, shifted, and rotated boxes from an initial cube that can best fit the vehicles. Then, labels of the intermediate frames are automatically initialized by linear interpolation between two successive keyframes. Since vehicles move at unknown speed and acceleration, the interpolation results can be inaccurate. To resolve this problem, we further request annotators to manually adjust each interpolated box using the same interface and verify the annotation results for all the frames. Overall, we follow the best practices of manual labeling to obtain clean and accurate training dataset.

10. Network Architecture and Training Details

To enable broader research community to reproduce MVDNet, we present architecture details of MVDNet-Fast and MVDNet in Fig. 6. Following the architecture overview in Fig 3 in the main paper, the model consists of 4 parts, i.e., feature extractor (blue), proposal generator (green), fusion network (red), and detection head (brown). MVDNet generates three types of outputs. Precisely, the class score consists of two values, indicating the likelihood of the prediction being a vehicle or the background. The bounding box consists of five values, i.e., the 2d locations, width, height, and orientation angle of the detected vehicle. The box direction consists of two values, indicating the direction of the detected vehicle. The difference between MVDNet and MVDNet-Fast is that the latter reduces the number of channels of the fusion network and the detection head by $8\times$.

Fig. 7 shows the detailed architecture of the lidar reconstruction model. Compared with MVDNet, a U-Net is prepended to fuse the lidar and radar input at the early stage, and the attention-based sensor fusion is removed. The lidar reconstruction model is trained with two steps. First, the U-Net is trained with a binary cross-entropy loss between the occupancy maps and a smooth l_1 loss between the intensity map of the reconstructed and clear lidar data, i.e., $L_{rec} = L_{BCE,occ} + L_{l_1,int}$. To train the U-Net, we use the Adam optimizer with an initial learning rate of 0.01, decay the learning rate by a factor of 0.1 every 20K iterations, and train the model for 80K iterations from scratch. Each iteration takes the input with a batch size of 4. Second, the whole lidar reconstruction model, including the U-Net, is trained with the reconstruction loss L_{rec} and the task loss of MVDNet (Eq. 3 in the main paper). The training setting is the same as MVDNet (Sec. 4.1 in the main paper).

11. Limitations and Future Work

Fusion of extra sensors. MVDNet fuses lidar and radar signals to improve the performance of vehicle detection in foggy weather condition. However, the fusion of lidar and radar still cannot cover all vehicles in practice (e.g., false alarms and missing cases in the second scene in Fig. 3), mainly due to the low resolution and noises of radar signals at far distances. We plan to exploit more diverse sensors in future work, e.g., Doppler radar, sonar, RGBD camera, infrared camera, to achieve more robust vehicle detection, especially in critical adverse weather conditions.

Real-time vehicle detection for autonomous driving. As shown in Fig. 8 in the main paper, the runtime of MVDNet with four historical frames is 110.7 ms (approximately 9 FPS). When processing only a single frame, MVDNet's runtime can be reduced to 54.9 ms (18 FPS) with some loss of detection accuracy. Overall, MVDNet achieves subsecond level processing speed. However, it still does not

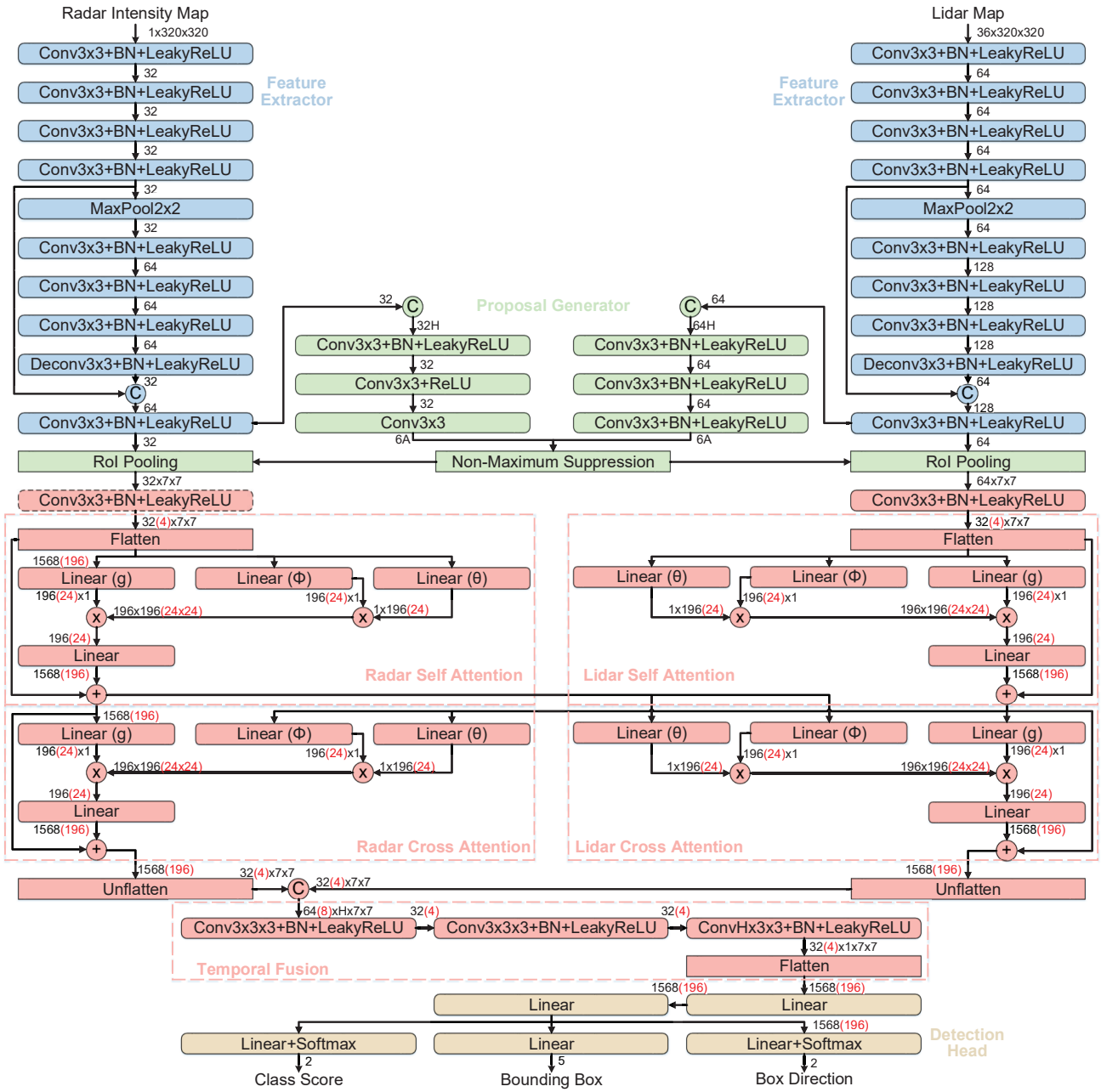


Figure 6. The detailed architecture of MVDNet and MVDNet-Fast. The red numbers in parentheses are the channels of MVDNet-Fast. $+$ means element-wise addition, $\times$ means matrix multiplication, and C means concatenation of two tensors.

support real-time vehicle detection compared to the speed of high-rate sensors, such as cameras (≥ 30 FPS). As a large portion of time cost comes from the processing of historical frames, it is possible to reduce the processing time by identifying reusable parts of historical results [10]. Besides, we plan to explore network compression methods [7, 9] to improve the detection efficiency in future work.

References

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *Proceedings of the IEEE ICRA*, 2020. 5

[2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE CVPR*, 2020. 1, 4, 5

[3] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE CVPR*, 2019. 1, 4

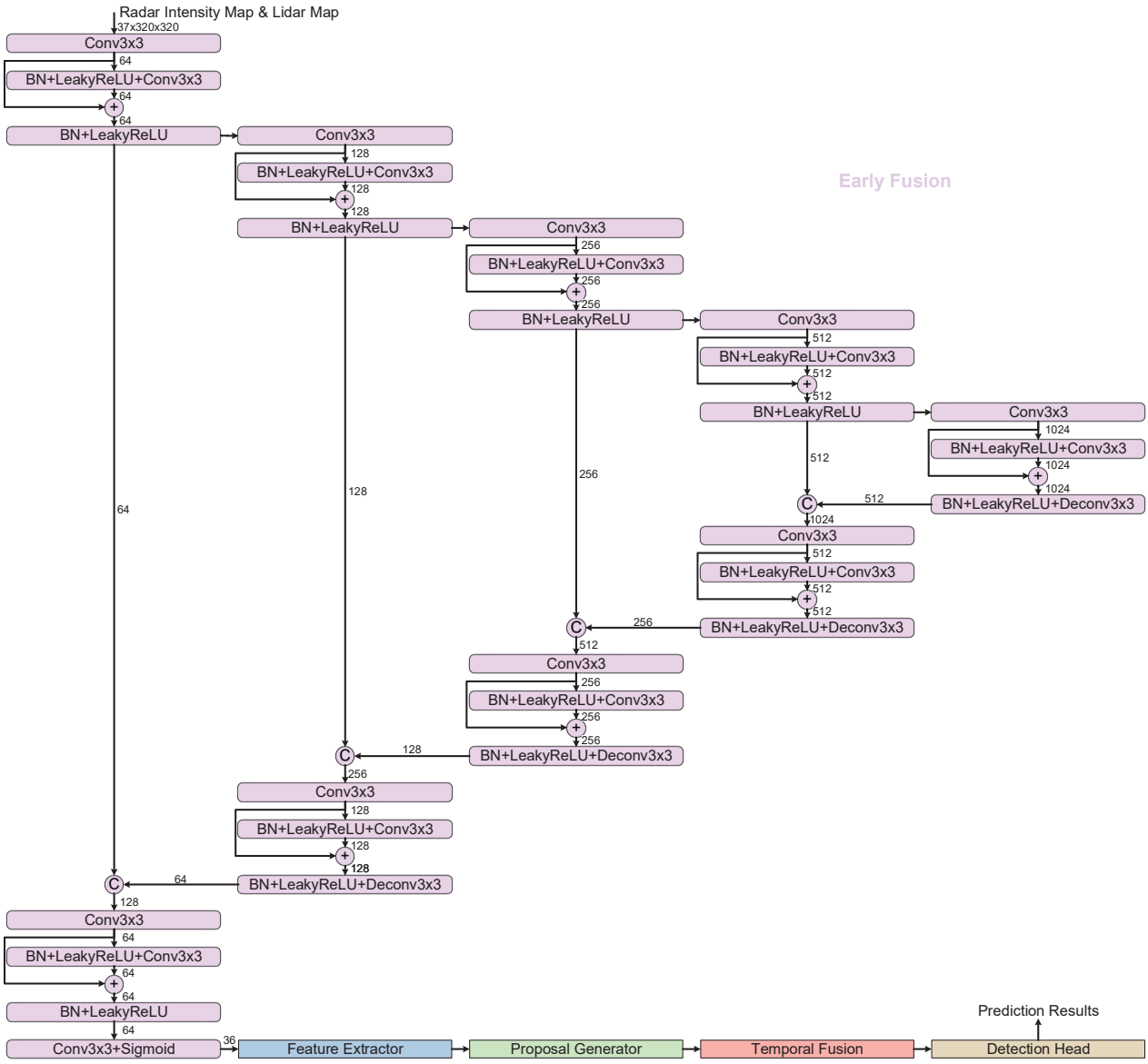


Figure 7. The detailed architecture of the lidar reconstruction model. The purple part is the U-Net that fuses the lidar and radar input. The rest parts as in Fig. 6 assemble the single branch of MVDNet without sensor fusion.

[4] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE CVPR*, 2018. 1, 2, 4

[5] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE CVPR*, 2019. 1, 4

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the IEEE ICLR*, 2015. 1

[7] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE CVPR*, 2019. 6

[8] Rob Weston, Sarah Cen, Paul Newman, and Ingmar Posner. Probably unknown: Deep inverse sensor modelling radar. In *Proceedings of the IEEE ICRA*, 2019. 3

[9] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE CVPR*, 2016. 6

[10] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. Deepcache: Principled cache for mobile deep vision. In *Proceedings of the ACM MobiCom*, 2018. 6

[11] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE CVPR*, 2018. 1, 4

[12] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 5