Multi-Task Multi-Sensor Fusion for 3D Object Detection

Ming Liang[1*] Bin Yang[1,2*] Yun Chen[1†] Rui Hu[1] Raquel Urtasun[1,2]
[1]Uber Advanced Technologies Group [2]University of Toronto
{ming.liang, byang10, yun.chen, rui.hu, urtasun}@uber.com

Abstract

In this paper we propose to exploit multiple related tasks for accurate multi-sensor 3D object detection. Towards this goal we present an end-to-end learnable architecture that reasons about 2D and 3D object detection as well as ground estimation and depth completion. Our experiments show that all these tasks are complementary and help the network learn better representations by fusing information at various levels. Importantly, our approach leads the KITTI benchmark on 2D, 3D and BEV object detection, while being real time.

1. Introduction

Self driving vehicles have the potential to improve safety, provide mobility solutions for otherwise underserved sectors of the population and reduce pollution. Fundamental to its core is the ability to perceive the scene in real-time. Most autonomous driving systems rely on 3-dimensional perception, as it enables interpretable motion planning in bird's eye view.

Over the past few years we have seen a plethora of methods that tackle the problem of 3D object detection from monocular images [2, 26], stereo cameras [4] or LiDAR point clouds [31, 29, 15]. However, each sensor has its challenges: cameras have difficulty capturing fine-grained 3D information, while LiDAR provides very sparse observations at long range. Recently, several attempts [5, 16, 12, 13] have been developed to fuse information from multiple sensors. Methods like [16, 6] adopt a cascade approach by using cameras in the first stage and reasoning using point clouds from LiDAR-only at the second stage. However, such cascade approach suffers from the weakness of each single sensor. As a result, it is difficult to detect objects that are occluded or far away. Others [5, 12, 13] have proposed to fuse features instead. Single-stage detectors like [13] fuse multi-sensor feature maps using LiDAR point as pixel correspondence. Local nearest neighbor interpolation is used to densify the correspondence. However, the fusion is limited when LiDAR points become extremely sparse at long range. Two-stage detectors [5, 12] fuse multi-sensor features per object at Region-Of-Interest (ROI) level. However, the fusion process is slow (as it involves thousands of ROIs) and imprecise (either using fix-sized anchors or ignoring object orientation).

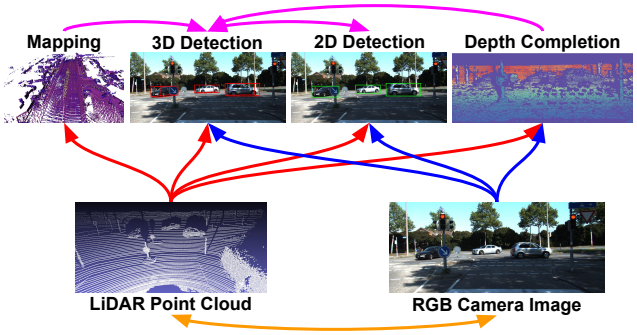

Figure 1. Different sensors (bottom) and tasks (top) are complementary to each other. We propose a joint model that reasons on two sensors and four tasks, and show that the target task - 3D object detection can benefit from multi-task learning and multi-sensor fusion.

In this paper we argue that by performing multiple perception tasks, we can learn better feature representations that result in better detection performance. Towards this goal, we developed a multi-sensor detector that reasons about 2D and 3D object detection, ground estimation and depth completion. Importantly, our model can be learned end-to-end and performs all these tasks at once. We refer the reader to Fig. 1 for an illustration of our approach.

We propose a new multi-sensor fusion architecture that leverages the advantages from both point-wise and ROI-wise feature fusion, resulting in fully fused feature representations. Knowledge about the location of the ground can provide useful cues for 3D object detection in the context of self driving, as the traffic participants stick out of it. Our detector estimates an accurate pointwise ground location online as one of its auxiliary tasks. This in turn is used by

*Equal contribution.
†Work done as part of Uber AI Residency program (https://careersinfo.uber.com/ai-residency).

the main bird's eye view (BEV) backbone to reason about relative location. We also exploit the task of depth completion to learn better cross-modality feature representation and more importantly, help achieve dense point-wise feature fusion.

We demonstrate the effectiveness of our approach on the KITTI object detection benchmark [8] as well as the more challenging TOR4D object detection benchmark [29]. On the KITTI benchmark, we show very significant performance improvement over other state-of-the-art approaches in 2D, 3D and Bird's Eye View (BEV) detection tasks. In particular, we surpass the second best 3D detector by over 3% in Average Precision (AP). Meanwhile, the proposed detector also runs over 10 frames per second, making it a practical solution for real-time applications. On the TOR4D benchmark, we show detection improvement from multi-task learning over previous state-of-the-art detector.

2. Related Work

We focus our literature review on works that exploit multi-sensor fusion and multi-task learning to improve 3D object detection.

3D detection from single modality: Early approaches to 3D object detection focus on camera based solutions, with monocular or stereo images [3, 2]. However, they suffer from the inherent difficulties of estimating depth from images and as a result perform poorly in 3D localization. More recent 3D object detectors rely on depth sensors such as LiDAR [29, 31]. However, although range sensors provide precise depth measurements, the observations are usually sparse (particularly at long range) and lack the information richness of images. It is thus difficult to distinguish classes such as pedestrian and bicyclist with LiDAR-only detectors.

Multi-sensor fusion for 3D detection: Recently, a variety of 3D detectors that exploit multiple sensors (e.g., LiDAR and camera) have been proposed. F-PointNet [16] uses a cascade approach to fuse multiple sensors. Specifically, 2D object detection is done first on images, 3D frustums are then generated by projecting 2D detections to 3D and PointNet [17, 18] is applied to regress the 3D position and shape of the bounding box. In this framework the overall performance is bounded by either stage which is still using single sensor. Furthermore, regressing positions from a frustum in LiDAR point cloud has difficulty dealing with occluded or far away objects as LiDAR observation can be very sparse (often containing a single point on the object). MV3D [5] generates 3D proposals from LiDAR features, and refines the detections by Region-Of-Interest (ROI) feature fusion from LiDAR and image features. AVOD [12] further adds ROI feature fusion to the proposal generation stage to improve the proposal quality. However, ROI feature fusion happens only at high-level feature maps. Furthermore, it only fuses features at selected object regions instead of densely over the feature map. To overcome this drawback, ContFuse [13] uses continuous convolutions to fuse multi-scale convolutional feature maps, where the correspondence between modalities is computed through projection of the LiDAR points. However, such fusion is limited when LiDAR points are very sparse. To resolve this issue, in this paper we propose to predict dense depth from LiDAR and image and use the predicted depth points to find dense correspondences between the feature maps from the two sensor modalities.

3D detection from multi-task learning: Various tasks have been exploited to help improve 3D object detection. HDNET [28] exploits geometric ground shape and semantic road masks to improve 3D object detection. Our model also reasons about a geometric map. The difference is that this module is part of our detector and thus end-to-end trainable, so that these two tasks can be optimized jointly. Wang et al. [25] exploit depth reconstruction and semantic segmentation to help 3D object detection. However, they rely on rendering, which is computationally expensive. Other contextual cues such as the room layout [21, 24], and support surface [22] have also been exploited to help 3D object reasoning in the context of indoor scenes. 3DOP [3] exploits monocular depth estimation to refine the 3D shape and position based on 2D proposals. Mono3D [2] proposes to use instance segmentation and semantic segmentation as evidence, along with other geometric priors to reason about 3D object detection from monocular images. In contrast to the aforementioned approaches, in this paper we also exploit depth completion which provides two benefits: it guides the network to learn better cross-modality feature representations and its prediction is exploited for dense pixel-wise feature fusion between the two-stream backbone networks.

3. Multi-Task Multi-Sensor Detector

One of the fundamental tasks in autonomous driving is to be able to perceive the scene in real-time. In this paper we propose a multi-task multi-sensor fusion model for the task of 3D object detection. We refer the reader to Fig. 2 for an illustration of the overall architecture. Our model has the following highlights. First, we design a multi-sensor architecture that combines point-wise and ROI-wise feature fusion. Second, our integrated ground estimation module reasons about the geometry of the scene. Third, we exploit the task of depth completion to learn better multi-sensor features and achieve dense point-wise feature fusion. As a result, the whole model can be learned end-to-end by exploiting a multi-task loss. Importantly, it achieves superior detection accuracy over the state of the art, with real-time

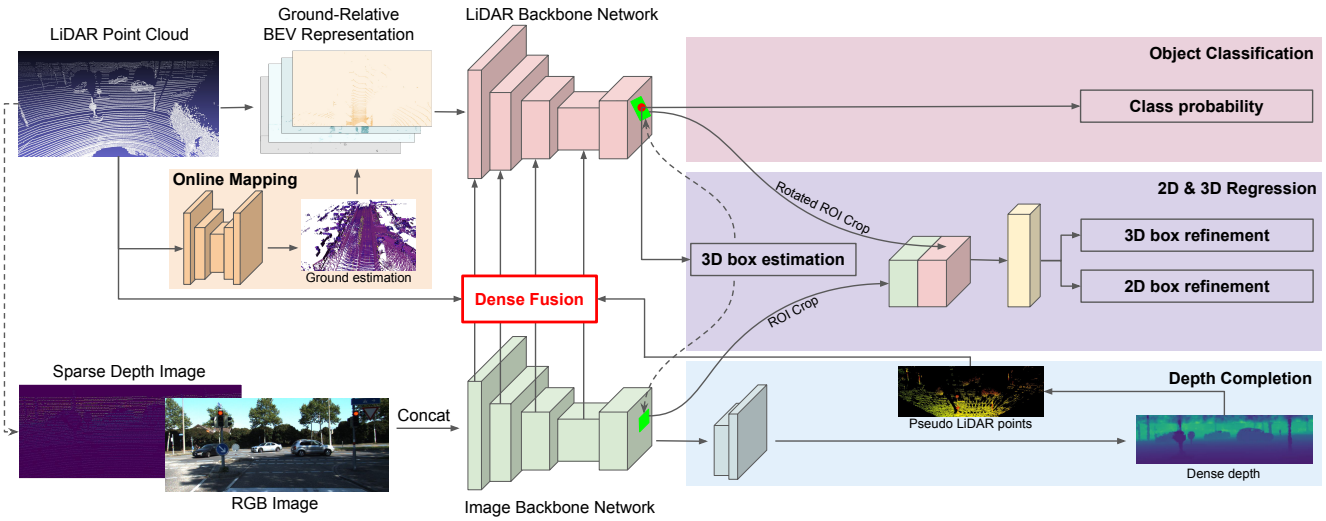

Figure 2. The architecture of the proposed multi-task multi-sensor fusion model for 2D and 3D object detection. Dashed arrows denote projection, while solid arrows denote data flow. Our model is a simplified two-stage detector with densely fused two-stream multi-sensor backbone networks. The first stage is a single-shot detector that outputs a small number of high-quality 3D detections. The second stage applies ROI feature fusion for more precise 2D and 3D box regression. Ground estimation is explored to incorporate geometric ground prior to the LiDAR point cloud. Depth completion is exploited to learn better cross-modality feature representation and achieve dense feature map fusion by transforming predicted dense depth image into dense pseudo LiDAR points. The whole model can be learned end-to-end.

efficiency.

In the following, we first introduce the single-task fully fused multi-sensor detector architecture with point-wise and ROI-wise feature fusion. We then show how we exploit the other two auxiliary tasks to further improve 3D detection. Finally we provide details of how to train our model end-to-end.

3.1. Fully Fused Multi-Sensor Detector

Our multi-sensor detector takes a LiDAR point cloud and an RGB image as input. It then applies a two-stream architecture as the backbone network with point-wise feature fusion at multiple layers. After the backbone network, the detector directly outputs high-quality 3D object detections via convolution thanks to multi-scale feature fusion. We then perform ROI-wise feature fusion via precise ROI feature extraction, and feed the fused ROI feature to a refinement module to produce very accurate 2D and 3D detections. Since the high-quality 3D detections are predicted via a fully convolutional network, the refinement network with ROI feature fusion only has to process a small number of detections (typical fewer than 20 on KITTI). This makes our two stage architecture very efficient.

Input representation: We use the voxel based LiDAR representation of [13] due to its efficiency. In particular, we voxelize the point cloud into a 3D occupancy grid, where the voxel feature is computed via 8-point linear interpolation on each LiDAR point. This LiDAR representation has the advantage of capturing fine-grained point density information efficiently. We consider the resulting 3D volume as Bird's-Eye-View (BEV) representation by treating the height slices as feature channels. This allow us to reason in 2D BEV space. This simplification brings significant efficiency gains with no performance drop. We simply use the RGB image as input for the camera stream. When we exploit the auxiliary task of depth completion, we additionaly add a sparse depth image generated by projecting the LiDAR to the image plane.

Network architecture: The backbone network follows a typical two-stream architecture to process multi-sensor data. We use a 2D fully convolutional residual network [10] as feature extractor. Specifically, for the image stream we use a ResNet-18 [10] architecture until the fourth residual block. Each block contains 2 residual layers with number of feature maps increasing from 64 to 512 linearly. For the LiDAR stream, we use a customized residual network which is deeper and thinner than ResNet-18 for a better trade-off between speed and accuracy. In particular, we have four residual blocks with 2, 4, 6, 6 residual layers in each, and the numbers of feature maps are 64, 128, 192 and 256. We also remove the max pooling layer before the first residual block to maintain more details in the point cloud feature. On the LiDAR stream we apply a Feature Pyramid Network (FPN) [14] with 1×1 convolution and bilinear

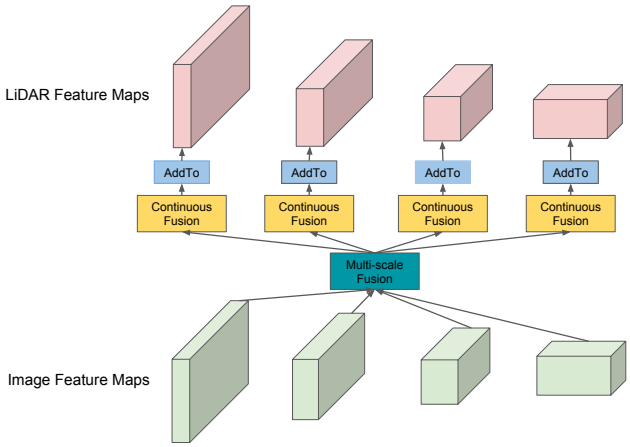

Figure 3. Point-wise feature fusion between multi-scale feature maps from LiDAR and image backbone networks.

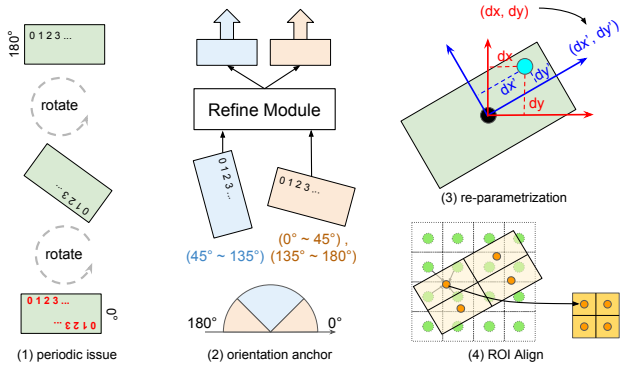

Figure 4. Precise rotated ROI feature extraction that takes orientation cycle into account. **(1)** The rotational periodicity causes abrupt change of order in feature extraction. **(2)** ROI refine module with two orientation anchors. An ROI is assigned to 0° or 90°. They share most refining layers except for the output. **(3)** The regression target of relative offsets are re-parametrized with respect to the object orientation axes. **(4)** A $n \times n$ sized feature is extracted using bilinear interpolation (we show an example with $n = 2$).

up-sampling to combine multi-scale features. Similarly we apply another FPN on the image stream to combine multi-scale image features. As a result, the final feature maps on the two streams have a down-sampling factor of 4 compared with the input. On top of the feature map output from the LiDAR stream, we simply add a 1×1 convolution to output the object classification and 3D box regression for 3D detections. After score thresholding and oriented Non-Maximum-Suppression (NMS), a small number of high-quality 3D detections are projected to both LiDAR BEV space and 2D image space, and their ROI features are cropped from each stream's backbone feature map via precise ROI feature extraction. The two-stream ROI features are fused together and fed into a refinement module with two 256-dimension Fully Connected (FC) layers to predict the 2D and 3D box refinements for each 3D detection.

Point-wise Feature Fusion: We apply point-wise feature fusion between the convolutional feature maps of LiDAR and image streams. The fusion is directed from image steam to LiDAR steam to augment BEV features with information richness of image features. We gather multi-scale features from all four blocks in the image backbone network by up-sampling the low resolution maps and element-wisely add them together. These multi-scale image features are then fused to each block of the LiDAR backbone network. Fig. 3 shows an example depicting fusion of multi-scale image features to the first block of LiDAR backbone network.

To fuse multi-sensor convolutional feature maps, we need to find the pixel-wise correspondence between the two sensors. Inspired by [13], we use continuous fusion to establish dense and accurate correspondences between the image and BEV feature maps. For each pixel in the BEV feature map, we find its nearest LiDAR point and project the point onto the image feature map to retrieve the corresponding image feature. We compute the distance between the BEV pixel and LiDAR point as the geometric feature. Both image feature and geometric feature are pass as input into a Multi-Layer Perceptron (MLP) and the output is fused to BEV feature maps by element-wise addition.

ROI-wise Feature Fusion: The motivation of the ROI-wise feature fusion is to further refine the localization precision of the high-quality 3D detections. Towards this goal, the ROI feature extraction needs to be precise so as to properly predict the relative box refinement. By projecting a 3D detection onto the image and BEV feature maps, we get an axis-aligned image ROI and an oriented BEV ROI. Feature extraction on axis-aligned image ROI is straightforward. However, there are two new issues arising from oriented BEV ROI (see Fig. 4). First, the periodicity of the ROI orientation causes the abrupt change of feature extraction order at the cycle boundary. To solve this issue, we propose an oriented ROI feature extraction module with anchors. Given an oriented ROI, we first assign it to one of the two orientation anchors, 0 or 90 degrees. All ROIs belonging to an anchor have a consistent feature extraction order. The two anchors share the refinement net except for the output layer. Second, when the ROI is rotated, its location offsets have to be represented in the rotated coordinates as well. To implement this, we first compute the location offset in the original coordinates, and then rotate them to be aligned with the ROI. Similar to ROIAlign[9], we extract bilinearly interpolated feature from a $n \times n$ regular grid in the ROI (in practice we use $n = 5$).

3.2. Multi-Task Learning for 3D Detection

In this paper we exploit two auxiliary tasks to improve 3D object detection, namely ground estimation and depth completion. They help in different ways: ground estimation provides geometric priors to enhance the LiDAR point clouds. Depth completion guides the image network to learn better cross-modality feature representations. Furthermore, it provides dense point-wise feature fusion.

3.2.1 Ground estimation

Mapping is an important task for autonomous driving, and in most cases the map building process is done offline. However, online mapping is appealing for that it decreases the system's dependency on offline built maps and increases the system's robustness. Here we focus on one basic subtask in mapping of ground estimation, which is to estimate the road geometry on-the-fly from a single LiDAR sweep. We formulate the task as a regression problem, where we estimate the ground height value for each voxel in the BEV space. This formulation is more accurate than plane based parametrization [3, 1], as in practice the road is often curved especially when we look far ahead.

Network architecture: We apply a small U-shaped Fully Convolutional Network (FCN) to estimate the normalized voxel-wise ground geometry at an inference time of 8 ms. We chose a U-Net architecture [23] since it outputs prediction at the same resolution as the input, and is good at maintaining low-level details.

Map fusion: Given a voxel-wise ground estimation, we first extract point-wise ground height by looking for the point index during voxelization. We then subtract it from each LiDAR point's Z axis value and generate a new LiDAR BEV representation (relative to ground), which is fed to the LiDAR backbone network. On the first stage regression output, we add the ground height back to the predicted Z term. The on-the-fly predicted ground geometry helps make 3D object localization easier because traffic participants, which are our objects of interest, all lay on the ground.

3.2.2 Depth completion

LiDAR provides long range 3D information for accurate 3D object detection. However, the observation is sparse especially at long range. Here, we propose to densify LiDAR observations by depth completion by exploiting both LiDAR and images. Specifically, given the projected (into the image plane) depth observation from the LiDAR and a camera image, the model outputs dense depth at the same resolution as the input image.

Sparse depth image from LiDAR projection: We first generate a three-channel sparse depth image from the LiDAR data, representing the sub-pixel offsets and the depth value. Specifically, we project each LiDAR point (x, y, z) to the camera space, denoted as $(x_{cam}, y_{cam}, z_{cam})$ (the Z axis points to the front of the camera), where z_{cam} is the depth of the LiDAR point in camera space. We then project the point from camera space to image space, denoted as (x_{im}, y_{im}). We find the pixel (u, v) closest to (x_{im}, y_{im}), and compute $(x_{im} - u, y_{im} - v, z_{cam}/10)$ as the value of pixel (u, v) on the sparse depth image [1]. For pixel locations with no LiDAR point, we set the pixel value to zero. After generating the sparse depth image, we concatenate it with the RGB image along the channel dimension and feed to the image backbone network.

Network architecture: The depth completion network shares the same backbone as the image backbone network, and applies four convolutional layers accompanied with two bilinear up-sampling layers to regress the dense pixel-wise depth at the same resolution with the input image.

Dense depth for dense point-wise feature fusion: As mentioned above, the point-wise feature fusion relies on LiDAR points to find the feature map correspondence. However, since LiDAR measurements are sparse by nature, the point-wise feature fusion can be sparse, especially when the image has a larger resolution than LiDAR (for example, images captured by a camera with long-focus lens). In contrast, the depth completion task provides dense depth information per image pixel, and therefore can be used as "pseudo" LiDAR points to find dense feature map correspondences between the two modalities. In practice, we use the dense depth prediction for point-wise fusion only on pixels where there's no true LiDAR point found.

3.3. Joint Training

We employ mutli-task loss to train our multi-sensor detector end-to-end.

The full model outputs object classification, 3D box estimation, 2D and 3D box refinement, ground estimation and dense depth. During training, we have detection labels and dense depth labels, while ground estimation is optimized indirectly by the detection loss. There are two paths of gradient transmission for ground estimation. One is from the output where ground height is added to predicted Z term. The other goes through the LiDAR backbone network to the LiDAR point cloud input where ground height is subtracted from the Z coordinate.

For object classification L_{cls}, we use binary cross entropy on positive and negative samples. For

[1] We divide the depth value by 10 for normalization purpose.

Detector	Input Data		Time	2D AP (%)			3D AP (%)			BEV AP (%)		
	LiDAR	IMG	(ms)	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
SHJU-HW [30, 7]		✓	850	90.81	90.08	79.98	-	-	-	-	-	-
RRC [19]		✓	3600	90.61	**90.23**	87.44	-	-	-	-	-	-
MV3D [5]	✓		240	89.80	79.76	78.61	66.77	52.73	51.31	85.82	77.00	68.94
VoxelNet [31]	✓		220	-	-	-	77.49	65.11	57.73	89.35	79.26	77.39
SECOND [27]	✓		50	90.40	88.40	80.21	83.13	73.66	66.20	88.07	79.37	77.95
PIXOR [29]	✓		**35**	-	-	-	-	-	-	87.25	81.92	76.01
PIXOR++ [28]	✓		**35**	-	-	-	-	-	-	89.38	83.70	77.97
HDNET [28]	✓		50	-	-	-	-	-	-	89.14	86.57	78.32
MV3D [5]	✓	✓	360	90.53	89.17	80.16	71.09	62.35	55.12	86.02	76.90	68.49
AVOD [12]	✓	✓	80	89.73	88.08	80.14	73.59	65.78	58.38	86.80	85.44	77.73
ContFuse [13]	✓	✓	60	-	-	-	82.54	66.22	64.04	88.81	85.83	77.33
F-PointNet [16]	✓	✓	170	90.78	90.00	80.80	81.20	70.39	62.19	88.70	84.00	75.33
AVOD-FPN [12]	✓	✓	100	89.99	87.44	80.05	81.94	71.88	66.38	88.53	83.79	77.90
Our MMF	✓	✓	80	**91.82**	90.17	**88.54**	**86.81**	**76.75**	**68.41**	**89.49**	**87.47**	**79.10**

Table 1. Evaluation results on the testing set of KITTI 2D, 3D and BEV object detection benchmark (car). We compare with previously published detectors on the leaderboard ranked by Average Precision (AP) in the moderate setting.

the 3D box estimation L_{box} and 3D box refinement losses L_{r3d}, we parametrize a 3D object as $(x, y, z, \log(w), \log(l), \log(h), \theta)$, and apply smooth $\ell 1$ loss on each dimension for positive samples only. For 2D box refinement loss L_{r2d}, we parametrize a 2D object as $(x, y, \log(w), \log(h))$, and also apply smooth $\ell 1$ loss on each dimension. For dense depth prediction loss L_{depth}, we sum $\ell 2$ loss over all pixels. The total loss for training the model is then defined as follows:

$$Loss = L_{cls} + \lambda(L_{box} + L_{r2d} + L_{r3d}) + \gamma L_{depth}$$

where λ, γ are the weights to balance different tasks during training.

A good initialization is important to train successfully. We therefore use the pre-trained ResNet-18 to initialize the image backbone network. For the additional channels added to the image input, we set their corresponding weights to zero. We also pre-train the ground estimation network on TOR4D dataset [29] with offline maps as labels and $\ell 2$ loss as objective function [28]. Other networks in the model are initialized randomly. We train the model with stochastic gradient descent using Adam optimizer [11].

4. Experiments

In this section, we first evaluate the proposed method on the KITTI 2D/3D/BEV object detection benchmarks [8]. We also provide a detailed ablation study to analyze the gains bring by multi-sensor fusion and multi-task learning. We then evaluate on the more challenging TOR4D multi-class BEV object detection benchmark [29].

4.1. Object Detection on KITTI

Dataset and metric: KITTI's object detection dataset has 7,481 frames for training and 7,518 frames for testing. We evaluate our approach on "Car" class. We apply the same data augmentation as [13] during training, which utlizes random translation, orientation and scaling on LiDAR point clouds and camera images. For multi-task training, we also leverage the dense depth labels from the intersection of KITTI's depth completion and object detection datasets. KITTI's detection metric is defined as Average Precision (AP) averaged over 11 points on the Precision-Recall (PR) curve. The evaluation criterion for cars is 0.7 Intersection-Over-Union (IoU) in 2D, 3D or BEV. KITTI also divides labels into three subsets (easy, moderate and hard) according to the object size, occlusion and truncation levels, and ranks methods by AP in the moderate setting.

Implementation details: We detect objects within 70 meters forward and 40 meters to the left and right of the ego-car, as most of the labeled objects are within this region. We voxelize the cropped point cloud into a volume of size $512 \times 448 \times 32$ as the LiDAR input representation. We also center-crop the images of different sizes into a uniform size of 370×1224. We train the model on a 4 GPU machine with a total batch size of 16 frames. We set the initial learning rate to 0.001 for Adam optimizer [11] and decay it after 30 and 45 epochs respectively. The training ends after 50 epochs.

Evaluation results: We compare our approach with previously published state-of-the-art detectors in Table 1, and show that our approach outperforms competitors by a large margin in all 2D, 3D and BEV detection tasks. In 2D detection, we surpass the best image detector RRC [19] by 1.1% AP in the hard setting, while being $45\times$ faster. Note that we only use a small ResNet-18 network as the image stream backbone network, which shows that 2D detection

Model	Multi-Sensor		Multi-Task			2D AP (%)			3D AP (%)			BEV AP (%)		
	pt	roi	map	dep	depf	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
LiDAR only						93.44	87.55	84.32	81.50	69.25	63.55	88.83	82.98	77.26
+image	✓					+2.95	+1.97	+2.76	+4.62	+5.21	+3.35	+0.70	+2.39	+1.25
+map	✓		✓			+3.06	+2.20	+3.33	+5.24	+7.14	+4.56	+0.36	+3.77	+1.59
+refine	✓	✓	✓			+3.94	**+2.71**	+4.66	**+6.43**	+8.62	+12.03	+7.00	+4.81	+2.12
+depth	✓	✓	✓	✓		**+4.69**	+2.65	+4.64	+6.34	**+8.64**	**+12.06**	+7.74	+5.16	+2.26
full model	✓	✓	✓	✓	✓	+4.61	+2.67	**+4.68**	+6.40	+8.61	+12.02	**+7.83**	**+5.27**	**+2.34**

Table 2. Ablation study on KITTI object detection benchmark (car) training set with four-fold cross validation. *pt*: point-wise feature fusion. *roi*: ROI-wise feature fusion. *map*: online mapping. *dep*: depth completion. *depf*: dense fusion with dense depth.

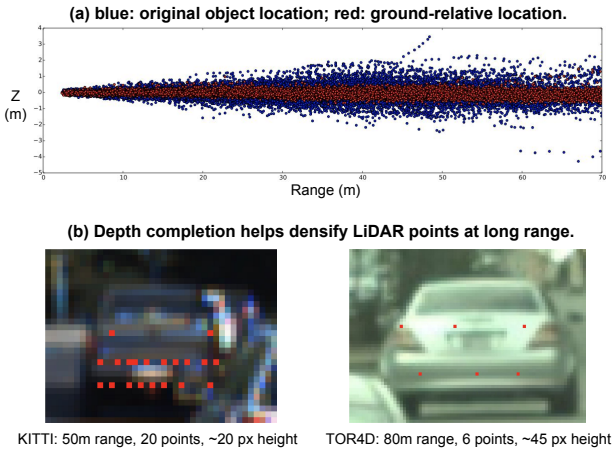

Figure 5. Object detection benefits from ground estimation and depth completion.

benefits a lot from exploiting the LiDAR sensor and reasoning in 3D detection. In BEV detection, we outperform the best detector HDNET [28], which also exploits ground estimation, by 0.9% AP. The improvement mainly comes from multi-sensor fusion. In the most challenging 3D detection task (as it requires 0.7 3D IoU), we show an even larger gain over competitors. We surpass the best detector SECOND [27] by 3.09% AP, and outperform the previously best multi-sensor detector AVOD-FPN [12] by 4.87% AP. We believe the large gain mainly comes from the fully fused feature representation and the proposed ROI feature extraction for precise object localization.

Ablation Study: To analyze the effects of multi-sensor fusion and multi-task learning, we conduct an ablation study on KITTI training set. We use four-fold cross validation and accumulate the evaluation results over the whole training set. This produces stable evaluation results for apple-to-apple comparison. We show the ablation study results in Table 2. Our baseline model is a single-shot LiDAR only detector. Adding image stream with point-wise feature fusion brings over 5% AP gain in 3D detection, possibly because image features provides complementary information on the Z axis in addition to the BEV representation of LiDAR. Ground estimation improves 3D and BEV detection by 1.9% and 1.4% AP respectively in moderate setting. This suggests that the geometric ground prior provided by online mapping is very helpful for detection at long range (Fig. 5), where we have very sparse 3D LiDAR measurements. Adding the refinement module with ROI-wise feature fusion brings consistent improvements on all three tasks, which purely comes from more precise localization. This proves the effectiveness of the proposed orientation aware ROI feature extraction. Lastly, the model further benefits in BEV detection from the depth completion task with better feature representations and dense fusion, which suggests that depth completion provides complementary information in BEV space. On KITTI we do not see much gain from dense point-wise fusion using estimated depth. We hypothesize this is because in KITTI the captured image is at equivalent resolution of LiDAR at long range (Fig. 5). Therefore, there is not much juice to squeeze from another modality. However, as we will see in next section, on TOR4D benchmark where we have higher resolution camera images, we show that depth completion helps not only by multi-task learning, but also dense feature fusion.

Model	Vehicle		Pedestrian		Bicyclist	
	$AP_{0.5}$	$AP_{0.7}$	$AP_{0.3}$	$AP_{0.5}$	$AP_{0.3}$	$AP_{0.5}$
ContFuse [13]	95.1	83.7	88.9	80.7	72.8	58.0
+dep	95.6	84.5	88.9	81.2	74.3	62.2
+dep+depf	**95.7**	**85.4**	**89.4**	**81.8**	**76.3**	**63.1**

Table 3. Ablation study of BEV object detection with multi-task learning on TOR4D benchmark. *dep*: depth completion. *depf*: dense fusion using estimated dense depth.

4.2. BEV Object Detection on TOR4D

Dataset and metric: The TOR4D BEV object detection benchmark [29] contains over 5,000 video snippets with a duration of around 20 seconds each. To generate the training and testing dataset, we sample from different snippets at 1 Hz and 0.5Hz respectively, leading to around 100,000 training frames and around 6,000 testing frames. To validate the effectiveness of depth completion in improving object detection, we use images captured by camera with long-focus lens which provide richer information at long range

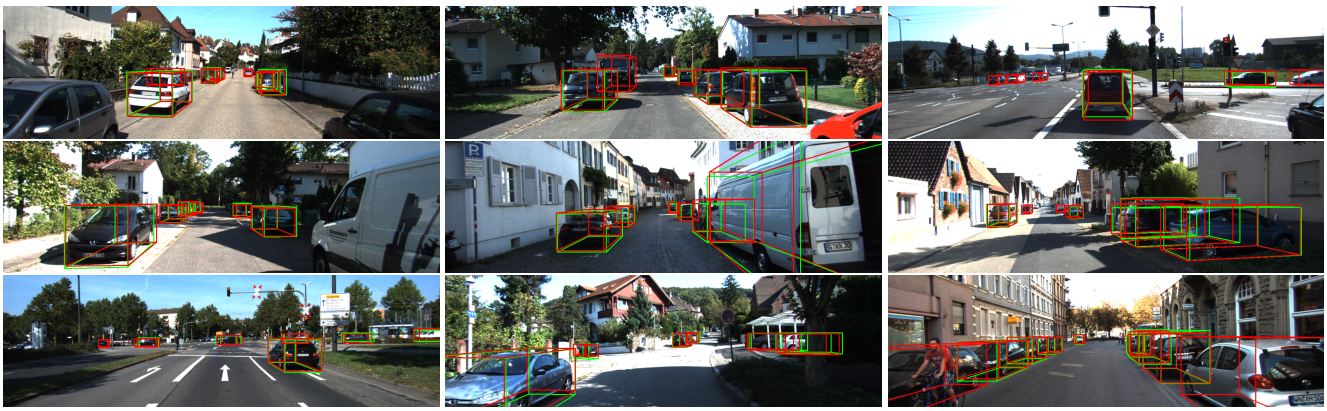

Figure 6. Qualitative results of 3D object detection (car) on KITTI benchmark. We draw object labels in green and our detections in red.

(Fig. 5). We evaluate on multi-class BEV object detection (i.e., vehicle, pedestrian and bicyclist) with a range of 100 meters distance from the ego-car. We use AP at different IoU thresholds as the metric for multi-class object detection. Specifically, we look at 0.5 and 0.7 IoU for vehicles, 0.3 and 0.5 IoU for the pedestrians and cyclists.

Evaluation results: We re-produce the previously state-of-the-art detector ContFuse [13] on TOR4D under our current setting. Two modifications are made to further improve the detection performance. First, we follow FAF [15] to fuse multi-frame of LiDAR point clouds together. Second, following HDNET [28] we incorporate semantic and geometric High-Definition map priors to the detector. We use the new ContFuse detector as the baseline, and apply the proposed depth completion with dense fusion on top of it. As shown in Table 3, the depth completion task helps in two ways: multi-task learning and dense feature fusion. The former increases the bicyclist AP by an absolute 4.2%. Since bicyclists have the fewest number of labels in the dataset, having additional multi-task supervision is particularly helpful. In terms of dense fusion with estimated depth, the performance on vehicles improves by over 5% in terms of relative error reduction (1-AP). The reason may be that vehicles receive more additional feature fusion compared to the other two classes (Fig. 5).

4.3. Qualitative Results and Discussion

We show qualitative 3D object detection results of the proposed detector on KITTI benchmark in Fig. 6. The proposed detector is able to produce high-quality 3D detections of objects that are highly occluded or far away from the ego-car. Some of our detections are unannotated cars in KITTI. Previous works [5, 12] often follow state-of-the-art 2D detection framework (like two-stage Faster RCNN [20]) to solve 3D detection. However, we argue that it may not be the optimal solution. With thousands of pre-defined anchors, the feature extraction is both slow and inaccurate. Instead we show that by detecting 3D objects in BEV space, we can produce high-quality 3D detections via a single pass of FCN (as shown in ablation study), given that we fully fuse the multi-sensor feature maps via dense fusion.

Cascade approaches [16, 6] suggest that 2D detection is solved better than 3D detection, and therefore use 2D detector to generate 3D proposals. However, we argue that 3D detection is actually easier than 2D. Because we detect objects in 3D metric space, we do not have to handle the problems of scale variance and occlusion reasoning that arise in 2D. Our model, using a pre-trained ResNet-18 as image network and trained from thousands of object labels, surpasses F-PointNet [16], which exploits two orders of magnitude more training data, by over 7% AP in hard setting of KITTI 2D detection. Multi-sensor fusion and multi-task learning are highly interleaved. In this paper we provide a way to combine them together under the same hood. In the proposed framework, multi-sensor fusion helps learn better feature representations to solve multiple tasks, while different tasks in turn provide different types of cues to make feature fusion deeper and richer.

5. Conclusion

We have proposed a multi-task multi-sensor detection model that jointly reasons about 2D and 3D object detection, ground estimation and depth completion. Point-wise and ROI-wise feature fusion are applied to achieve full multi-sensor fusion, while multi-task learning provides additional map prior and geometric cues enabling better representation learning and denser feature fusion. We validate the proposed method on KITTI [8] and TOR4D [29] benchmarks, and surpass the state-of-the-art in all detection tasks by a large margin. In the future, we plan to expand our multi-sensor fusion approach to exploit other sensors such as radar as well as temporal information.

References

[1] J. Beltran, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. de la Escalera. Birdnet: a 3d object detection framework from lidar information. *IEEE International Conference on Intelligent Transportation Systems*, 2018. 5

[2] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1, 2

[3] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015. 2, 5

[4] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *TPAMI*, 2017. 1

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 1, 2, 6, 8

[6] X. Du, M. H. Ang Jr, S. Karaman, and D. Rus. A general pipeline for 3d detection of vehicles. In *ICRA*, 2018. 1, 8

[7] L. Fang, X. Zhao, and S. Zhang. Small-objectness sensitive detection based on shifted single shot detector. *Multimedia Tools and Applications*, pages 1–19, 2018. 6

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 6, 8, 9

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[12] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018. 1, 2, 6, 7, 8

[13] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 1, 2, 3, 4, 6, 7, 8

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[15] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 1, 8

[16] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 1, 2, 6, 8

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[19] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017. 6

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 8

[21] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, 2016. 2

[22] Z. Ren and E. B. Sudderth. 3d object detection with latent support surfaces. In *CVPR*, 2018. 2

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 5

[24] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV*, 2013. 2

[25] S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015. 2

[26] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2018. 1

[27] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 6, 7

[28] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *2nd Conference on Robot Learning (CoRL)*, 2018. 2, 6, 7, 8

[29] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 1, 2, 6, 7, 8, 9

[30] S. Zhang, X. Zhao, L. Fang, F. Haiping, and S. Haitao. Led: Localization-quality estimation embedded detector. In *IEEE International Conference on Image Processing*, 2018. 6

[31] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1, 2, 6

Supplementary Materials

We provide more quantitative and qualitative results on KITTI [8] and TOR4D [29] benchmarks.

Fig. 7 shows the PR curves of the proposed detector as well as other state-of-the-art approaches in 2D/3D/BEV car detection on KITTI test set for a more comprehensive comparison. In all detection settings, the proposed detector shows consistent advantage in terms of precision rate, which proves the effectiveness of the proposed joint model in producing high-quality detections.

Fig. 8 shows the fine-grained evaluation results of the proposed detector on TOR4D multi-class BEV object detection at different ranges and IoU thresholds. Note that by using depth completion for dense fusion, our approach achieves larger AP gains at long range.

Fig. 9 shows the qualitative results of depth completion on KITTI and TOR4D. Note that the camera on TOR4D has longer focal length, therefore the input depth image is more sparse. But the objects with predicted depth are also farther away, leading to more gain in long range detection.

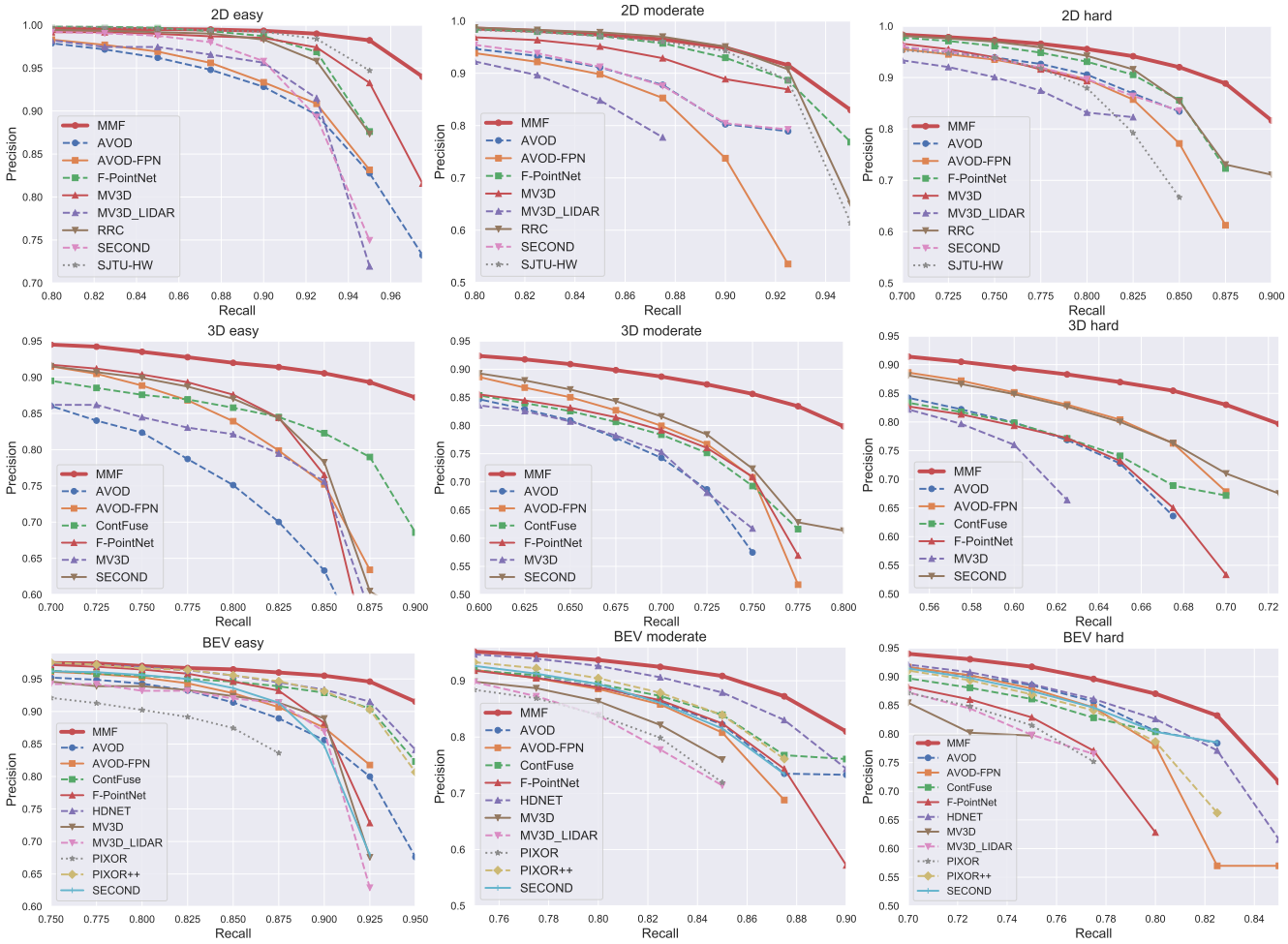

Figure 7. PR curve comparison between the proposed **MMF** and other state-of-the-art in 2D/3D/BEV car detection on KITTI testing set.

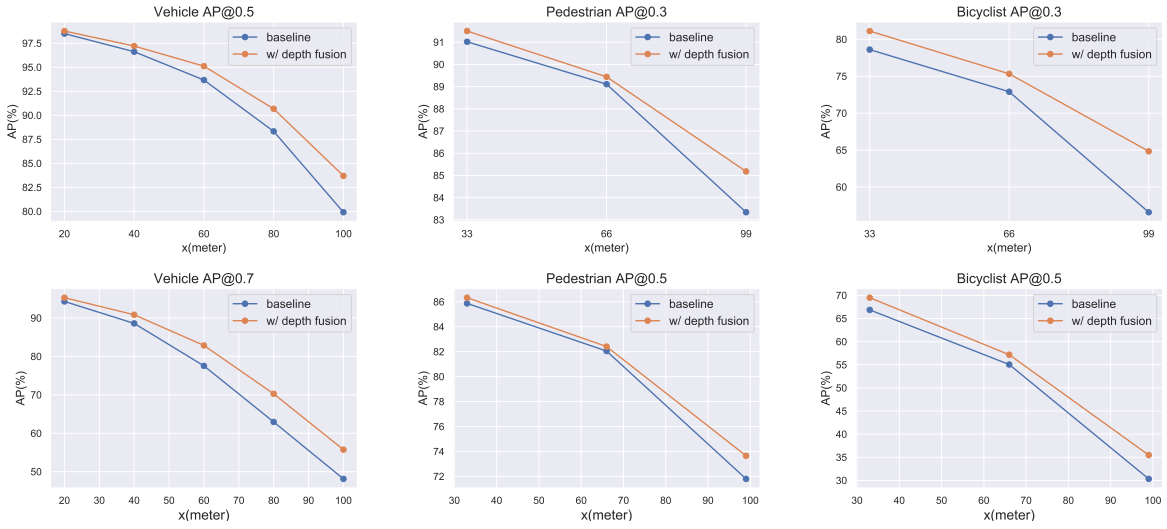

Figure 8. Range-wise evaluation on TOR4D BEV detection.

Figure 9. Qualitative results of depth completion on KITTI (first 2 examples) and TOR4D (last example).