Robust 3D Object Detection for Autonomous Vehicles with Cross-Modal Hallucination

Jianning Deng, Cabriel Chan, Hantao Zhong, Chris Xiaoxuan Lu

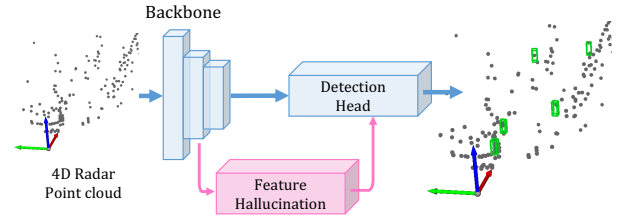
***Abstract*—This paper presents a novel framework for robust 3D object detection from point clouds via cross-modal hallucination. Our proposed approach is agnostic to either hallucination direction between LiDAR and 4D radar. We introduce instance feature aggregation and feature alignment to achieve simultaneous backbone refinement and hallucination generation. Specifically, instance feature aggregation is proposed to deal with the geometry discrepancy for better instance matching between LiDAR and radar. The feature alignment step further bridges the intrinsic attribute gap between the sensing modalities and stabilizes the training. The trained object detection models can deal with difficult detection cases better, even though only single-modal data is used as the input during the inference stage. Extensive experiments on the View-of-Delft (VoD) dataset show that our proposed method outperforms the state-of-the-art (SOTA) methods for radar and LiDAR object detection while maintaining competitive efficiency in runtime.**

I. Introduction

Robust recognition and localization of objects in 3D space is a fundamental computer vision task and an essential capability for intelligent systems. In the context of autonomous driving, accurate 3D object detection is vital for safe motion planning, especially in a complex urban environment. Due to accurate depth measurement in long-range and robustness to illumination conditions, ranging sensors such as LiDAR and radar have attracted increasing attention recently and in turn, make the point clouds from them one of the most commonly used data representations for 3D object detection.

While advances are witnessed in LiDAR- [13, 29, 37, 17, 30, 31, 42, 38, 39, 20] and radar-based 3D object detection [1, 22], each of them comes with intrinsic limitations.

LiDAR sensors can provide dense point clouds with informative geometry, but they lack per-point semantic information. Even though some LiDAR sensors can output point-level intensity values [9, 3], they are often inconsistent across semantic labels as lidar intensity is complexly determined by the distances between objects and LiDAR, rather than the object class alone [4]. Due to the lack of semantic information, far objects or occluded objects with incomplete shapes or the insufficient number of points will be difficult for LiDAR to detect [35]. On the other side, 4D radars recently emerge as a promising automotive sensor against bad weather [22, 18]. While their point clouds suffer from point sparsity and low data fidelity, these radars can provide rich semantic features like Radar Cross Section (RCS) and the (Doppler) velocity. Unlike LiDAR intensity, RCS is a distance-independent measurement uniquely determined by the object material and reflection angle. The Doppler velocity measures the moving speed of a detected point relative to the ego vehicle. Benefiting from the rich semantic information provided by RCS and velocity measurements, fairly reasonable 3D object performance can be achieved by less than 400 points per scan even in complex urban environments [22], while most of the objects contain less than 10 points per scan.



(a) Proposed method for 4D radar object detection

(b) Centerpoint (c) Pointpillar

(d) PV-RCNN (e) Ours

Fig. 1: Fig. 1a illustrates the proposed method with the 4D radar input as an example. Fig. 1b - Fig. 1e are the visualization of radar detection in the same scene of different methods. Ground truth boxes are denoted in red, the wrong detections are denoted in yellow and the correct detections are denoted in green. RGB images are **only used for visualization**.

The complementariness between these two sensors motivates us to explore the following question - *is it possible for one ranging sensor to learn to 'imagine the lacking bits' from another heterogeneous ranging sensor so that it can later individually detect objects beyond its comfort zone?* For example, since both model inputs are point clouds, can we let a radar detection model learn from a LiDAR detection model to improve object detection robustness and vice versa? Indeed, due to the cost consideration, many low-end autonomous vehicles are only equipped with either radars or LiDARs. It is thus valuable to train a *single-modal* detection model from the *multimodal* data collected by pilot vehicles equipped with both sensors, yet dispatch the trained models on low-end vehicles equipped with only one of them.

This essentially requires us to address a cross-modal learning problem. Prior arts achieve cross-modal learning via

hallucination for feature augmentation [34, 28, 12, 15, 5] or knowledge distillation [47, 40, 10, 27] for backbone refinement. Traditionally, the hallucination network is trained separately with raw input data to provide extra information for specific tasks. Consequently, the performance improvement of the framework depends entirely on the hallucination quality.

More recently, [47] leverages knowledge distillation to learn multi-modal information to improve a single-modal network. This method guides the backbone for better feature extraction but limits extra performance gain from hallucination features. Besides, this guidance only applies from the more robust modal-modal network to the weaker single-modal network.

Unlike prior arts, in this work, we aim to realize simultaneous feature augmentation and backbone refinement by a new cross-modal hallucination framework. Besides, to avoid information loss on sparse radar points and maintain runtime efficiency, we follow the point-based single shot detector architecture in our design. Notably, we target a framework agnostic to learning directions so that it can be applied to either hallucination direction between LiDAR and radar point clouds. To this end, we introduce multi-level alignment in this framework. The first one is space-level instance alignment which exploits instance co-location across different sensory data. Due to the discrepancy between sensory data and measurement noise from radar, it is difficult to establish fine-grained point-to-point matches between sensors. Thus, we employ an instance feature aggregation module that moves points to their corresponding object center to generate more comprehensive instance-level features. By doing so, points belonging to the same instance in different modalities will be positioned near the instance center, which eases the effort for cross-modal matching. The second one is feature-level alignment which is carried out by two learned non-linear mapping functions to project features from both modalities to a shared latent space respectively, which optimizes the training by closing the domain gap between modalities. With both spatial and feature alignment, our backbone can extract better intra-modality features via cross-modal learning and hallucinate inter-modality features to improve detection robustness, especially on hard samples (see Fig. 1). In summary, our contributions are as follows:

- Our method is the first object detection work from point clouds by utilizing cross-modal hallucination. Our method distinguishes itself in simultaneous backbone refinement and feature-level hallucination.
- We introduce spatial and feature-level alignment to resolve the intrinsic measurement differences between LiDAR and radar point clouds to establish correct cross-modal supervision signals.
- Our method outperforms previous state-of-the-art (SOTA) methods in 3D object detection, which we demonstrate on the public View-of-Delft dataset in both LiDAR and radar object detection tasks.

II. Related work

LiDAR object detection. Previous works on LiDAR 3D object detection can be categorized into three types: voxel-based, point-based, and point-voxel methods. The main idea of voxel-based methods is to group irregular point cloud data into compact grid cells, which can be effectively handled with convolution operators [48, 36, 39, 13, 6, 46, 8, 43, 45, 17, 31]. In particular, [13] converts the point clouds to 2D pseudo images and adopts a single-stage framework for better inference speed. [39] employs a two-stage pipeline with a 3D voxel backbone and object heatmap to further boost detection performance at the cost of a larger memory footprint. Another approach in LiDAR object detection is to process the unstructured point cloud data directly without quantization or information loss [29, 38, 42, 32]. These models [29, 42, 38] extract features from point clouds using operations proposed in [23] and [25]. Furthermore, [38, 42] explore different point sampling strategies for a better trade-off between memory efficiency and accuracy. Recent research [30, 37, 20, 11] started to design point-voxel networks that utilize both representations for better detection accuracy. As reported in [42], point-voxel methods like [30] achieve slightly better performance on the KITTI dataset [9] at the cost of much lower computation efficiency.

Radar object detection. Early efforts in radar-based object detection focused on 2D object detection [33, 41, 26]. It is only with the recent availability of 4D radar sensors that radar 3D object detection began receiving attention from researchers. [1] is an anchor-based 3D detection framework with a PointNet style backbone. It was only evaluated on short-range radar point clouds within $\sim$ 10 meters in front of the ego-vehicle, which is far too close for real-world autonomous scenarios. On the other hand, authors of the recently published dataset [22] successfully repurposed PointPillars [13] on radar point clouds. Although the 3D object detection architecture was designed for LiDAR point clouds, it achieved SOTA performance on their radar 3D object detection benchmark [22].

Cross-modal feature augmentation. Learning with side information means introducing additional information to the network during training, which yields a stronger single-modality network [28, 15, 5, 34, 12, 40, 10, 47]. [12] was the first to explore the effects of cross-modal hallucination for object detection. This work incorporates features from depth images to improve the detection performance of RGB images. Additional thermal image information is also used in [34] for 2D RGB pedestrian detection. Similar to hallucination, one can train a better-informed network via transfer learning from different modalities. Most recent work like [47] introduces multi-modal features at the training phase via knowledge distillation to a single modal network for better 3D detection results. Yet, the methods mentioned above fail to make full use of the cross-modal information. In fact, these works [34, 12] need an extra backbone branch to fulfill the hallucination goal, rather than reusing the original backbone to simultaneously extract detection features and hallucinate the features of the other modality. On the other side, knowledge distillation methods [40, 10, 47] only provide a more robust backbone but fail to take advantage of hallucinated information.

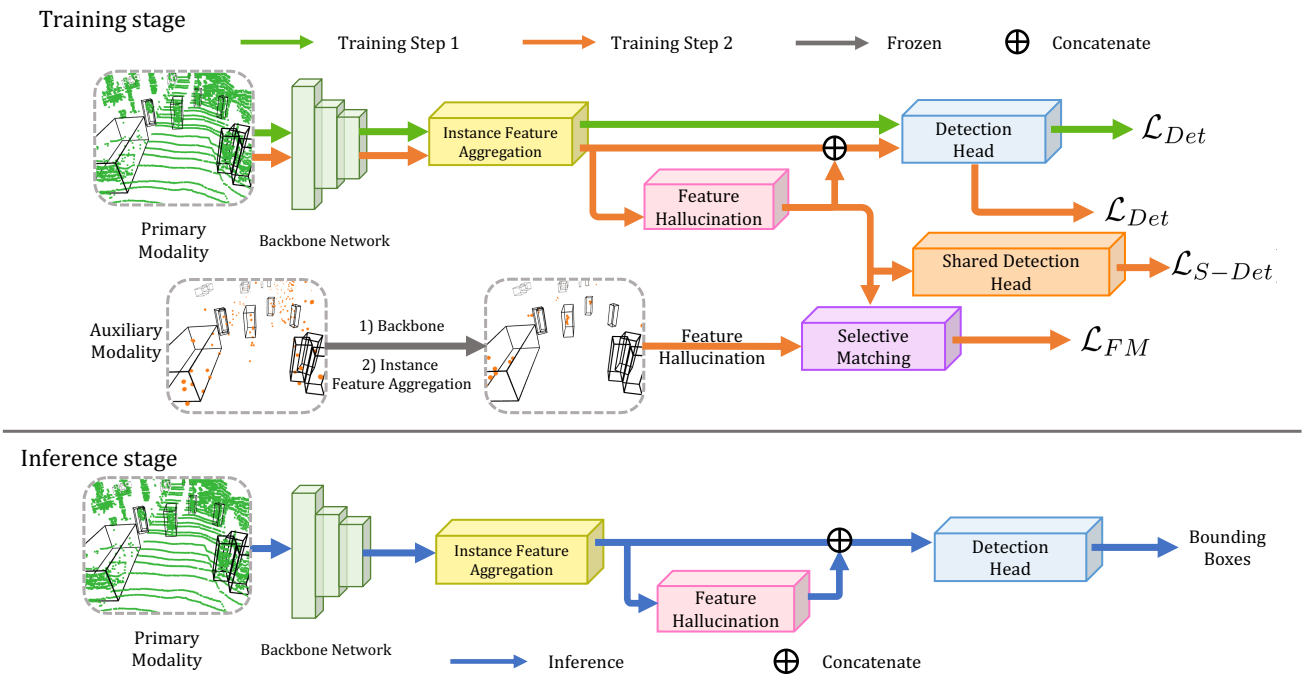


Fig. 2: Overview of the proposed framework. The upper figure illustrates the 2-step training strategy, blocks used in the first step training are connected with green line and those for second step are connected with orange line. Note the primary and auxiliary data can be interchangeable among two sensor modalities (radar and LiDAR) depending on the end goal. **Only single modal data (primary modal) will be used during inference** as shown in the lower figure connected with blue line. Best viewed in color.

III. Method

A. Overview

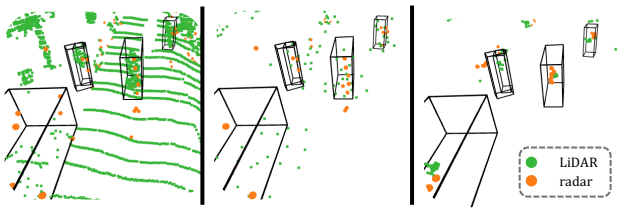
The overview of the proposed method is shown in Fig. 2. Our framework consists of four modules, coherent to the generic network architectures used by both lidar and radar-based object detection: (i) point-based backbone for feature extraction, (ii) instance feature aggregation module to resolve instance co-locality between input modalities, especially for noisy radar point clouds, (iii) hallucination branch for cross-modal feature alignment, and (iv) detection head for bounding box prediction. Besides, the selective matching module and shared detection head will only be used during training. As our framework is agnostic to modalities, for readability, we will use the term 'primary' and 'auxiliary' to interchangeably represent the two sensor modalities of our interest.

Notation and Context. The input to the backbone network is a point set $\mathbf{P} = \{p_i\} \in \mathbb{R}^{N\times(3+A)}$ of N points, where $p_i = [t_i, f_i^A]$, with $t_i \in \mathbb{R}^3$ as position and $f_i^A \in \mathbb{R}^A$ as the input point attribute vector. Here we use the superscript for the feature vector to denote its dimension. Notation for the auxiliary modal data is denoted with the same letter but with a **hat** on top of it, e.g., the input point set of the auxiliary modal is denoted as $\hat{\mathbf{P}}$. The backbone extracts features and samples foreground points. The output of the primary modal backbone is a subset of $\mathbf{P}$ with extracted features, denoted as $\mathbf{P}_f = \{p_i\} \in \mathbb{R}^{N_f\times(3+D)}$ with N_f points and $p_i = [t_i, f_i^D]$. Afterward, point set $\mathbf{P}_f$ is sent to the instance feature aggregation module to obtain 'centered points' and instance-level feature representation. The point set after centroid generation in instance feature aggregation is denoted as $C = \{c_i\} \in \mathbb{R}^{N_f\times(3+D)}$, with new position $t_i' = t_i + \tilde{o}_i$, where $\tilde{o}_i$ is a spatial offset to shift the point towards its corresponding object center. Once we obtain 'centered points' $\mathbf{C}$ and $\hat{\mathbf{C}}$ from both modalities, we adopt two non-linear mapping functions to perform the feature-level alignment for hallucination generation. The alignment process is constrained based on the instance location provided by the instance feature aggregation module. The hallucination features generated from the primary modality are denoted as $\mathbf{H} = \{h_i\} \in \mathbb{R}^{N_f\times F}$. Then we concatenate these features with the point set $\mathbf{C}$ and send them to the detection head for classification and bounding box regression. All mathematical notation is summarized in the supplementary.

B. Backbone Network

We construct our backbone network based on the Set-Abstraction (SA) layer proposed in PointNet++ [25] for better efficiency and to avoid information loss [38, 42]. Additionally, the farthest-point sampling (FPS) operation in SA layer is replaced with center-aware sampling proposed in [42] for better performance, which selects top-k points based on the predicted centeredness. This predicted centeredness is constrained during training as:

$$\mathcal{L}_{Ctr} = -\sum_{k=1}^{N}(Mask_k \cdot y_k log(\tilde{y}_k) + (1-y_k)log(1-\tilde{y}_k)) \quad (1)$$



(a) Inputs (b) Backbone results (c) Centered

Fig. 3: Point-level matches are difficult to obtain without centroid generation in the instance feature aggregation module due to sparsity. The black bounding boxes are the object ground truth labels. (Best viewed in color and zooming in).

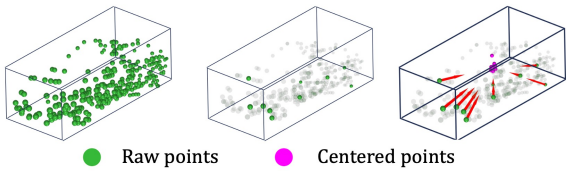


Fig. 4: Illustration of the centroid generation. The left figure is the input point clouds. The middle one shows the sampling process of the backbone network. Foreground points with high confidence (points in green) survived in the final layers. The right figure demonstrates the centroid generation process, where foreground points are moved toward the object center.

with y_k as the ground truth centeredness and $\tilde{y}_k$ as network estimated value. $Mask_k$ is the corresponding mask value for each point proposed in [38], which assigns higher weights to points closer to the centroid of objects, and no weight at all for background points.

C. Instance Feature Aggregation

Unlike traditional two-stage detectors that can establish instance matches based on accurate RoI (Region of Interest) estimations, a key challenge for cross-modal hallucination in single-stage detectors lies in establishing the point matching across two sensor modalities. As shown in Fig. 3a, due to the radar sparsity[22, 7], there exists an obvious discrepancy between the two point clouds captured by co-located LiDAR and radar even in the same scene. Such discrepancy is caused by the fundamental difference (e.g., the point density and positions) between the two sensors and cannot be eradicated, which can not be solved via calibration. Moreover, due to the introduced sampling randomness, this cross-modal discrepancy will be exacerbated by the foreground sampling step in the backbone (see Fig. 3b) where more point sparsity is exhibited and makes it more difficult to establish fine-grained point matches between LiDAR and radar. Realizing the difficulty of point-level matching, we decide to use instance feature aggregation for better spatially-aligned point sets before cross-modal point matching. This process is illustrated in Fig. 3c.

First, we follow [24] to create canonical 'centered points' for context aggregation from different parts of an instance. In particular, an offset $\tilde{o}_i \in \mathbb{R}^3$ for point position is estimated for each foreground point $p_i \in \mathbf{P}_f$ to shift the point towards the corresponding object center. This process is illustrated in Fig. 4. We constrain this regression process with a smooth-L1 loss as:

$$\mathcal{L}_{O-Reg} = \frac{1}{\sum_i \mathbf{I}(p_i)} \sum_{i=1}^{m} smooth_{L1}(o_i - \tilde{o}_i) \cdot \mathbf{I}(p_i) \quad (2)$$

where o_i is the ground truth offset for the foreground point to its corresponding object center and $\mathbf{I}(p_i)$ is an indicator function, it is 1 when p_i is inside an object and 0 otherwise. Unlike the original p_i, the new coordinates t'_i for the same object gather closely. We call this step as centroid generation.

Second, we utilize a set-abstraction(SA) layer on those 'centered points' with coordinates t'_i to acquire a robust instance-level feature representation. Points after this instance-aggregation step is denoted as $C = \{c_i\} \in \mathbb{R}^{N_f \times (3+D)}$, with new position $t'_i = t_i + \tilde{o}_i$. Since those points are closely gathered near the instance centroid, the ball query operation in the SA layer will group the same set of points most of the time, leading to the same instance feature for each point after max pooling. Thus, we can quickly establish instance matches between modalities by matching points in close spatial locations. These clustered points with shifted coordinates towards their centroids will be fed to the hallucination branch and detection head subsequently.

D. Cross-Modal Feature Hallucination

After centroid generation in Sec. III-C, we are now ready to deal with the domain misalignment on features. Due to the different sensing principles between LiDAR and radar, the extracted instance feature exhibit different information specific to the modality domain. For example, radar can measure the relative Doppler velocity [19, 18, 3] from a single scan, while LiDAR can give more accurate object geometry information. To close this domain gap and improve learning efficiency, feature-level alignment is needed.

Considering the aforementioned attribute difference in intrinsic, it is unreasonable to brute-force match the *entire* features across two modalities. At the minimum, one cannot expect to extract sufficient velocity-related features from a *single* LiDAR scan, even with a *'magical'* backbone. A feasible workaround is to use a common *subspace* to ground the cross-modal hallucination learning only on a subset of their features (the magenta block in Fig. 2). This essentially requires us to have a feature projection module in place so that the embedded per-point features can be projected onto a latent subspace shared by both modalities. Specifically, given the clustered point set $\mathbf{C} \in \mathbb{R}^{N_f \times (3+D)}$ of the primary modality in domain X and clustered point set $\hat{\mathbf{C}} \in \mathbb{R}^{\hat{N}_f \times (3+\hat{D})}$ of the auxiliary modality in domain $\hat{X}$, we have two mapping functions acting as the feature projection: $F_{pri} : X \rightarrow T$ and $F_{aux} : \hat{X} \rightarrow T$ to project both modalities to a shared common space T. We adpot a MLP module for each mapping function. The dimension of the shared common subspace is set empirically. Detailed experiments about the subspace dimension can be found in the supplementary.

E. Selective Matching

Given the projected hallucination features, we now need to find matching pairs between modalities to constrain the training process (the purple block in Fig. 2). Thanks to the instance feature aggregation module, we can establish correct matching pairs between modalities as points with instance features are now aligned closely near the object center (see Fig. 3c). We observe that noisy and misclassified points are clustered in random locations. Therefore, we simply apply a cross-modal Nearest Neighbor (1-NN) search with a predefined radius to find the correct matching pairs. This process is illustrated in Fig. 5a. Notice that only foreground points successfully shifted towards object centers can reach neighbor points from the other modality, as they are close in space. This leads to fast cross-modal representation learning due to fewer distractions from noisy points. The cross-modal hallucination loss is as follows:

$$\mathcal{L}_{FM} = \frac{1}{N_p} \sum_{i=0}^{N_f} \sum_{j=0}^{\hat{N}_f} ||F_{pri}(h_i) - F_{aux}(\hat{h}_i)||_2 \cdot \mathbf{PM}_{ij} \quad (3)$$

where $\mathbf{PM} \in \mathbb{R}^{N_f \times \hat{N}_f}$ is a binary matrix, and $\mathbf{PM}_{ij} = 1$ when c_i and $\hat{c}_j$ are a selected matching pair, otherwise $\mathbf{PM}_{ij} = 0$. And N_p denotes the total number of matching pairs. Note that for the cross-modal feature, there is a risk of falling into a trivial solution if the projected features from both modalities become zero vectors. We thus use an extra detection as a curator, which will be discussed in Sec. III-F.

F. Detection Head

Similar to [38, 42], we encode the bounding box into a multidimensional vector containing locations, scale, and bin-residual orientation. The detection head (the blue block in Fig. 2) adopts two branches for confidence prediction and box refinement, respectively. Thus, the loss function for the detection head can be written as:

$$\mathcal{L}_{Det} = \mathcal{L}_{ref} + \mathcal{L}_{cls} \quad (4)$$

To avoid trival solution for feature matching in Eq. 3, we use an extra detection head specifically for the shared space features, called shared detection head (the orange block in Fig. 2). This loss term is denoted as $\mathcal{L}_{S-Det}$.

G. Training and Inference

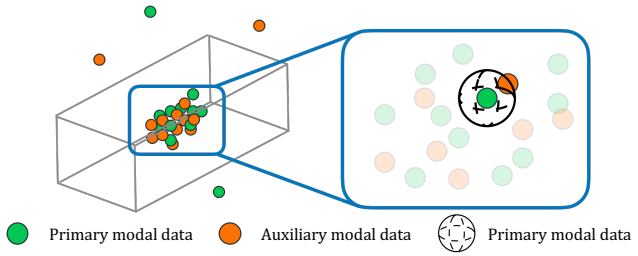
We use a two-step training strategy to stabilize the training process. It is worth noting that unlike other two-stage approaches [28, 34], the hallucination branch in our approach is jointly optimized with the backbone during the second step of training. In the first step, we train a simplified object detection network (data path connected with green line in Fig. 2). The loss function during this step is:

$$\mathcal{L}_{s1} = \mathcal{L}_{Ctr} + \mathcal{L}_{O-Reg} + \mathcal{L}_{Det} \quad (5)$$

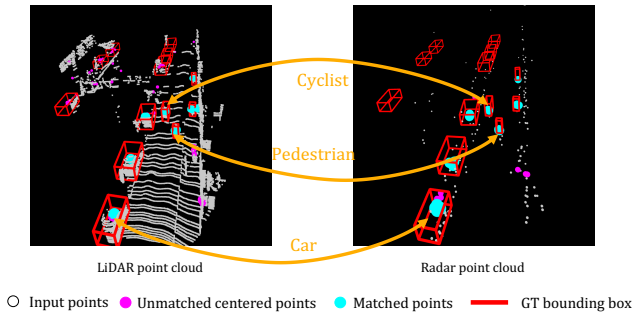
Later, we use the parameters trained in the first step for backbone and instance feature aggregation module initialization in the second step of training. The auxiliary backbone is frozen to provide stable cross-modal features, while the primary modal backbone is jointly optimized with the hallucination module. The optimization of the primary backbone is named as backbone refinement. A new detection head is also trained to handle the larger feature space better (data path connected with orange line in the upper figure of Fig. 2). The loss function we used during the second step of training is as follows:

$$\mathcal{L}_{s2} = \mathcal{L}_{s1} + \lambda_1 \mathcal{L}_{FM} + \lambda_2 \mathcal{L}_{S-Det} \quad (6)$$

Only the primary modality data is used during inference. Blocks for inference are illustrated in the lower figure of Fig. 2, which are connected in blue line.



(a) Illustration of selective matching. Points with lighter colors on the right are ignored in the match-searching process.



(b) Visualization of the training process for selective matching.

Fig. 5: Here is the illustration and visualization of the selective matching during training. In Fig. 5b, we can see that all matched points (cyan points) are positioned near the bounding box center for co-visible objects, which demonstrates the effectiveness of the instance feature aggregation module. For wrongly sampled centered points, they will be moved to random positions and will not interfere with the training process (magenta points).

IV. Evaluation

We next move to the evaluation and performance comparison. In particular, we leverage the recent availability of a co-located LiDAR and 4D radar dataset [22] and comprehensively evaluate the effectiveness of our method on both sensor modalities. *Note that our code and models will be made public based upon acceptance.*

A. Experimental Setup

Dataset. The View-of-Delft (VoD) dataset [22] provides calibrated and synchronized LiDAR, RGB camera, and 4D radar

data for 3D object detection. This dataset comprises 8693 frames of aligned multi-sensor data recorded in the complex urban environment in the city of Delft. VoD dataset features a considerable proportion of vulnerable road users such as pedestrians and cyclists. To our best knowledge, VoD is the only dataset with 4D radar recordings and LiDAR data simultaneously and can be accessed by the public.[1] Other datasets like nuScenes [3] also provide radar and LiDAR point clouds at the same time. But the automotive radar in [3] only provides *2D* spatial measurements with azimuth and range (x and y), which is not designed for *3D* object detection. Besides, unlike other predominant dataset used for 3D object detection[9, 3], VoD focuses on the cluttered urban traffic environment with crowded pedestrians and cyclists. Additionally, truncated and highly occluded objects are also included in the evaluation, making object detection on this dataset more realistic but also more challenging. We follow the protocol of the VoD evaluation system and report our test results here.

Implementation details. We follow the design of the backbone network in [42], which comprises several *SA layers* with center-aware sampling to remove noisy points. Next, the *Vote Layer* predicts an offset for each point to concentrate them on the corresponding object center for spatial alignment in centroid generation. After that, another *SA layer* is employed to aggregate instance-level features. Once we obtain point clusters for different objects, we project instance-level features to a shared subspace using a 4-layer MLP. These projected features are later concatenated back to the instance-level features for bounding box classification and regression. We use $\lambda_1 = \frac{1}{3}$ and $\lambda_2 = \frac{2}{3}$ in Eqn. 6 for all our experiments. Only the primary modal data will be used during inference. The best LiDAR and radar detection models are selected based on the best validation result for their respective detection tasks. Results for baseline models are retrained on VoD based on official implementation. More details about the training configuration and model implementation can be found in the supplementary.

B. Overall Results

Tab. I and Tab. II show that our proposed framework gives rise to the best overall performance for both modalities, achieving 41.77 mAP and 69.62 mAP for radar and LiDAR object detection, respectively. The overall mAP of our method outperforms the second-best model by 8.2% and 1.1%, respectively, on radar and LiDAR. Fig. 6 illustrates the qualitative results of our method and more qualitative comparisons can be found in the supplementary file.

Radar object detection. Given the sparse and noisy radar point clouds, the detection results of our method surpass the SOTA methods by a large margin. PV-RCNN [30] achieves the best result for cars detection because of less size ambiguity problem [16] with voxel representation on extremely sparse radar point clouds. However, our methods are much better on small objects such as cyclists and pedestrians. In particular, our model achieves 42.49 mAP for pedestrians, surpassing the PV-RCNN result with a large margin $\sim 21.7\%$. This indicates that better instance representation for small objects possessing sparse point clouds can be learned from cross-modal information - LiDAR in our case. In fact, the available points of an instance in LiDAR can easily surpass the radar points by 100 times. It is thus unsurprising that LiDAR detection models can extract better geometry representations for each instance, especially for small objects. Through our hallucination branch introduced in Sec. III-D, we enforce the radar detection network to have a branch mimicking the instance representation of LiDAR, and the results here confirm its effectiveness.

LiDAR object detection. On the LiDAR side, our method performs the best for cars and cyclists, overtaking the second-best by 2.6% and 2.2% mAP, respectively. Such improvement can be attributed to the semantic cues hallucinated from the Radar Cross Section (RCS) features, which provide highly discriminative features between metal materials and human skin[2, 14]. While two-stage detection methods generally [39, 30, 29] excel in pedestrian detection due to refinement

Method	Type	Car (IoU = 0.5)	Pedestrian (IoU = 0.25)	Cyclist (IoU = 0.25)	mAP
PointPillars†[13]	V	35.90	34.90	43.10	38.00
SECOND[36]		35.07	25.47	33.82	31.45
CenterPoint[39]		32.32	17.37	40.25	29.98
PV-RCNN[30]	PV	**38.30**	30.79	46.58	38.56
PointRCNN[29]	P	15.99	34.01	26.50	25.50
3DSSD[38]		23.86	9.09	32.20	21.71
IASSD[42]		31.33	23.61	49.58	34.84
Ours		32.32	**42.49**	**50.49**	**41.77**

TABLE I: Test results for **radar object detection** on VoD dataset. Note that results for PointPillars with symbol † are reported in [22]. The *'Type'* column denotes the data representation used in the method: *'V'* denotes voxel, *'P'* denotes point, *'PV'* denotes point-voxel. Methods underlined are all two-stage detectors.

Method	Type	Car (IoU = 0.5)	Pedestrian (IoU = 0.25)	Cyclist (IoU = 0.25)	mAP
PointPillars†[13]	V	75.60	55.10	55.40	62.10
SECOND[36]		77.69	59.95	65.50	67.71
CenterPoint[39]		68.29	66.90	64.42	66.54
PV-RCNN[30]	PV	75.16	65.24	66.09	68.83
PointRCNN[29]	P	61.51	**67.36**	67.03	65.30
3DSSD[38]		77.34	12.64	37.68	42.55
IASSD[42]		77.29	32.18	57.11	55.53
Ours		**79.74**	60.58	**68.52**	**69.62**

TABLE II: Test results for **LiDAR object detection** on VoD dataset. Note that results for PointPillars with symbol † are reported in [22]. The *'Type'* column denotes the data representation used in the method: *'V'* denotes voxel, *'P'* denotes point, and *'PV'* denotes point-voxel. Methods underlined are all two-stage detectors.

[1]Recent works [44, 21] also have similar sensor configurations, but the authors cannot give us access before the submission deadline.

Auxiliary radar point attributes	Car (IoU=0.7)	Pedestrian (IoU=0.5)	Cyclist (IoU=0.5)	mAP
+ none (baseline)	48.86	48.91	67.88	55.22
+ x, y, z	58.98	48.81	78.04	61.94
+ x, y, z, RCS	58.82	**56.29**	**78.17**	**64.42**
+ x, y, z, v	**59.27**	50.14	77.91	62.44
+ x, y, z, RCS, v	59.18	55.30	77.56	64.01

TABLE III: LiDAR detection results in *validation set* supervised by different radar attributes during training. x, y, z for point position, RCS for radar cross section, v for velocity.

Auxiliary LiDAR point attributes	Car (IoU=0.5)	Pedestrian (IoU=0.25)	Cyclist (IoU=0.25)	mAP
+ none (baseline)	31.24	32.50	60.69	41.48
+ x, y, z	31.32	40.04	67.78	46.38
+ x, y, z, I	**32.20**	**40.42**	**68.67**	**47.03**

TABLE IV: Radar detection results in *validation set* supervised by different LiDAR attributes during training. x, y, z for point position, I for intensity.

on the region-of-interest (ROI) in the second stage, this gain is at the cost of larger memory footprints and slower inference speed. It is noteworthy that our method demonstrates the best performance of pedestrians among single-stage detectors, which is much more efficient than two-stage architectures as discussed in Sec. IV-D.

C. Ablation Study

We now conduct extensive ablation studies to analyze the contributions of individual components of our proposed method. To be consistent with LiDAR detection work, here we use the more demanding 3D IoU in KITTI [9] with 0.7, 0.5, and 0.5 for LiDAR evaluation. and 0.5, 0.25, 0.25 for radar evaluation on the VoD [22] *validation* set.

Effect of auxiliary-modal attributes. The main research problem in this paper is to explore whether we can utilize cross-modal supervision and hallucination for more robust single-modal object detection. We thus conduct experiments with different auxiliary modal data input attributes.

(a). LiDAR object detection. The results in Tab. III indicate that the LiDAR detection results are significantly improved when radar supervision is available, confirming the effectiveness of exploiting cross-modal information to facilitate better object detection for single-modal inference. When all coordinate information (i.e., x, y, and z) is available in the radar input, the LiDAR detection performance is boosted on objects of cars and cyclists. We hypothesize that radar supervision guides the network to better handle objects with limited points, similar to the radar side. Tab. III also shows that the RCS attribute in radar inputs leads to the most notable performance gain overall, especially for pedestrian detection. We attribute this to the reliable semantic cues from the RCS measurements to differentiate pedestrians and vehicles, which are very different in terms of object materials. Due to the sparsity and noise of radar data, the proportion of detected points on metal objects (bikes) and people could be completely different between cyclist objects. Thus, the RCS measurements on cyclists are less stable than that on pedestrians and eventually lead to marginal performance improvement for cyclist objects.

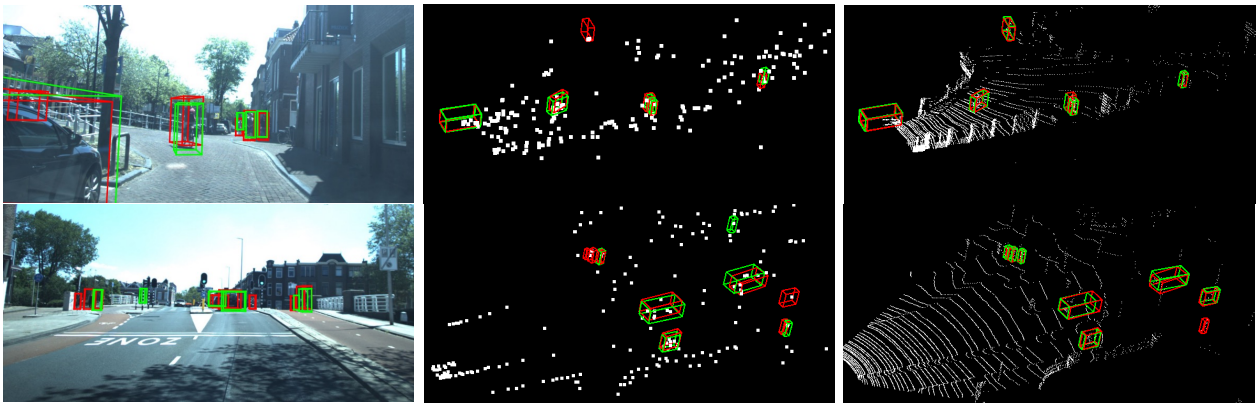
The LiDAR detection network can, therefore, better differentiate objects with the cross-modal supervision provided by RCS. Besides, the largest car performance gains are found when radar velocity measurements are used. This boost indicates that the velocity difference between cars and low-speed objects (e.g., pedestrians and cyclists) can also be hallucinated and used for better vehicle detection results. However, combining radar RCS and velocity does not lead to further gains. The reason is the velocity ambiguity between pedestrians and cyclists travelling at similar speeds, which undermines the overall detection accuracy.

(b). Radar object detection. Similar effectiveness of the proposed cross-modal learning is also found for radar object detection. Tab. IV shows that the radar object detection model enjoys a performance boost by hallucinating LiDAR point cloud (with only x, y, z), which contains only geometric information. This result supports our hypothesis that radar detection can benefit from instance features provided by a more robust modality. Meanwhile, adding the intensity attribute from LiDAR only marginally contributes to the performance since radar already has more stable semantic cues from the RCS measurements. We also notice that the performance improvement for the Car category is less significant than the other two categories. There are two reasons for this result. First, about 25% of annotated cars have no radar points, which means they are impossible to be detected. Second, the size of cars is much larger than that of cyclists and pedestrians, making it more challenging to estimate the bounding boxes with limited points (about 50% of the annotated cars contain less than 3 points). More details about these statistics will be provided in the supplementary material.

Effect of joint optimization. We study the effectiveness of our joint optimization strategy.

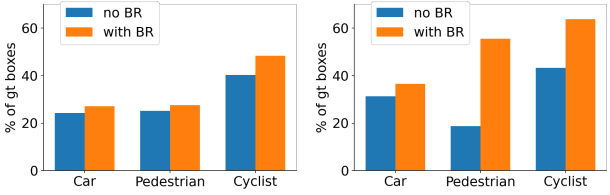
(a). Setup. We use the base network trained without cross-modal supervision and the hallucination branch as the ablation baseline. Later, we remove the hallucination branch of our framework shown in Fig. 2, freeze the backbone weights and train a new detection head for performance comparison.

(b). Results. Comparing the first and second rows in both modalities in Tab. V, it is clear the mAP is improved in both modalities, even without the hallucination branch. The most noticeable improvement is the Cyclist on the radar and the Pedestrian on the LiDAR side. For LiDAR, we hypothesize that fewer matched instances for cars cause a slight drop in performance with **BR** (Backbone Refinement) only, but improves performance for other categories. To further investigate why performance is improved with the refined backbone only, we gather some statistics on the backbone output. We can see from Fig. 7 that the percentage of instances containing more than five points gets improved. We thus believe that cross-modal representation learning can help the backbone network extract more robust features. Those robust features can be partially propagated to nearby points through max pooling, leaving more effective points of an instance to be kept in the



(a) Radar detection in perspective (b) Radar detection (c) LiDAR detection

Fig. 6: Qualitative result of our method. Bounding boxes for GTs are denoted in red, and the predictions are denoted in green. The left images and the middle figures are the radar detection results. Notice that the RGB images here are **only for visualization purposes but not used in model training/inference**. The right figures visualize the LiDAR point clouds and the prediction results.



(a) Radar backbone statistics (b) LiDAR backbone statistics

Fig. 7: Percentage of instance containing 5-20 points in the backbone output. Left for **radar**, right for **LiDAR**, The **BR** denotes backbone refinement.

Modality	BR	HA	Car (IoU=0.5)	Pedestrian (IoU = 0.25)	Cyclist (IoU = 0.25)	mAP
radar			31.24	32.50	60.69	41.48
	✓		31.51	33.48	67.19	43.94
	✓	✓	**32.20**	**40.42**	**68.67**	**47.03**
			(IoU=0.7)	(IoU = 0.5)	(IoU = 0.5)	mAP
LiDAR			57.94	29.40	68.26	51.87
	✓		55.72	48.64	76.22	60.20
	✓	✓	**58.82**	**56.29**	**78.17**	**64.42**

TABLE V: Evaluation for different modalities on the validation set. **BR** means network with backbone refinement which optimizes the backbone in the second step training with cross-modal supervision, **HA** means network with hallucination branch.

last sampling layer, eventually yielding a more robust instance feature representation for better detection. More detailed statistics can be found in the supplementary. Furthermore, the last row in each modality shows that the entire network with the hallucination branch leads to the best performance for all three categories, indicating its effectiveness and that extra cross-modal features can be encoded in our model for better detection inference.

Effect of two-level alignment. We next study the LiDAR

Modality	Spatial	Feature	Car (IoU=0.7)	Pedestrian (IoU=0.5)	Cyclist (IoU=0.5)	mAP
LiDAR			57.94	29.40	68.26	51.87
		✓	28.88	30.22	64.33	41.15
	✓		57.29	48.66	77.70	61.22
	✓	✓	**58.82**	**56.29**	**78.17**	**64.42**

TABLE VI: Ablation on the effect of spatial alignment in centroid generation and feature level alignment to the cross-modal hallucination. Notice that we use the basic model trained without any cross-modal information in Tab. V as the baseline performance.

object detection results to better understand the influence of our alignment strategies introduced in Sec. III-C and Sec. III-D. Tab. VI shows that spatial alignment in centroid generation is the key to cross-modal learning. When removed, the discrepancy between modalities results in mismatching pairs. Misleading supervision signals caused by mismatched pairs degenerate the network performance, especially for large objects like Cars. When the spatial alignment in centroid generation is added, the model establishes correct cross-modal instance matches and improves mAP, as shown in the third row of Tab. VI. We attribute the slight performance drops on cars to fewer matched instances in this category. As expected, the feature alignment will only function and positively contribute to the model when the cross-modal points are spatially aligned first. The last row shows that combining two alignment operations yields the best performance. Similar conclusions are observed for radar detection, and we omit its discussion to avoid repetition.

D. Runtime Efficiency

We evaluate the memory consumption and inference speed compared with SOTA methods with LiDAR as the primary modal input. As shown in Tab. VII, our method has the second lowest memory footprint among all methods and the fastest

Type	Method	Memory	Parallel	Speed
Voxel-based	PointPillars[13]	354MB	69	123
	SECOND[36]	710MB	34	34
Point-Voxel	PV-RCNN[30]	1223MB	17	13
Point-based	pointRCNN[29]	560MB	43	14
	3DSSD[38]	502MB	48	20
	IA-SSD[42]	120MB	202	23
	Ours	**133MB**	**183**	**23**

TABLE VII: Comparisons of memory usage and runtime efficiency with LiDAR input. The memory footprint for each method is measured by feeding the same 16384 points per scan. Parallel denotes the maximum batch size that can be parallelized in one RTX 3090. The speed is measured with single scan input and reported in frames per second. Results of our method are shown in **bold**.

inference speed in the point-based methods. Together with Tab. II and Tab. I, our framework has the best balance between efficiency and accuracy.

V. Conclusions and Future Work

This work introduced a novel framework to improve the robustness of single-modal 3D object detection via cross-modal supervision. Our method is able to effectively exploit the side information from auxiliary modality data for a better-informed backbone network and a robust hallucination branch. Bespoken spatial and domain alignment strategies are also proposed to address the fundamental discrepancy across modalities and showed a significant performance improvement. Experimental results on VoD[22] demonstrate that our method achieves the best overall results in both radar and LiDAR object detection. It outperforms SOTA methods in terms of accuracy while maintaining a competitive memory footprint and runtime efficiency.

References

[1] Kshitiz Bansal, Keshav Rungta, Siyuan Zhu, and Dinesh Bharadia. Pointillism: Accurate 3d bounding box estimation with multi-radars. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 340–353, 2020.

[2] Emna Bel Kamel, Alain Peden, and Patrice Pajusco. Rcs modeling and measurements for automotive radar applications in the w band. In *2017 11th European Conference on Antennas and Propagation (EUCAP)*, pages 2445–2449, 2017. doi: 10.23919/EuCAP.2017.7928266.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Patrick Chazette, Julien Totems, Laurent Hespel, and Jean-Stéphane Bailly. 5 - principle and physics of the lidar measurement. In Nicolas Baghdadi and Mehrez Zribi, editors, *Optical Remote Sensing of Land Surface*, pages 201–247. Elsevier, 2016. ISBN 978-1-78548-102-4. doi: https://doi.org/10.1016/B978-1-78548-102-4.50005-3. URL https://www.sciencedirect.com/science/article/pii/B9781785481024500053.

[5] Chiho Choi, Sangpil Kim, and Karthik Ramani. Learning hand articulations by hallucinating heat distribution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3104–3113, 2017.

[6] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1201–1209, May 2021. doi: 10.1609/aaai.v35i2.16207. URL https://ojs.aaai.org/index.php/AAAI/article/view/16207.

[7] Fangqiang Ding, Zhijun Pan, Yimin Deng, Jianning Deng, and Chris Xiaoxuan Lu. Self-supervised scene flow estimation with 4-d automotive radar. *IEEE Robotics and Automation Letters*, 7(3):8233–8240, 2022. doi: 10.1109/LRA.2022.3187248.

[8] Liang Du, Xiaoqing Ye, Xiao Tan, Jianfeng Feng, Zhenbo Xu, Errui Ding, and Shilei Wen. Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13329–13338, 2020.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

[10] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

[11] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11873–11882, 2020.

[12] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 826–834, 2016.

[13] 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/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[14] Seongwook Lee, Seokhyun Kang, Seong-Cheol Kim, and Jae-Eun Lee. Radar cross section measurement with 77 ghz automotive fmcw radar. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[15] José Lezama, Qiang Qiu, and Guillermo Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6628–6637, 2017.

[16] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7546–7555, 2021.

[17] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (07):11677–11684, Apr. 2020. doi: 10.1609/aaai.v34i07.6837. URL https://ojs.aaai.org/index.php/AAAI/article/view/6837.

[18] Michael Meyer and Georg Kuschk. Automotive radar dataset for deep learning based 3d object detection. In *2019 16th European Radar Conference (EuRAD)*, pages 129–132, 2019.

[19] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. In *2019 16th European Radar Conference (EuRAD)*, pages 133–136. IEEE, 2019.

[20] Jongyoun Noh, Sanghoon Lee, and Bumsub Ham. Hvpr: Hybrid voxel-point representation for single-stage 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14605–14614, 2021.

[21] Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. K-radar: 4d radar object detection dataset and benchmark for autonomous driving in various weather conditions. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, December 2022. URL https://github.com/kaist-avelab/K-Radar.

[22] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian FP Kooij, and Dariu M Gavrila. Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2): 4961–4968, 2022.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[24] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[26] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 444–453, 2021.

[27] Sucheng Ren, Yong Du, Jianming Lv, Guoqiang Han, and Shengfeng He. Learning from the master: Distilling cross-modal advanced knowledge for lip reading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13325–13333, 2021.

[28] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Chris Xiaoxuan Lu, Yasin Almalioglu, Stefano Rosa, Changhao Chen, Johan Wahlström, Wei Wang, Andrew Markham, and Niki Trigoni. Deeptio: A deep thermal-inertial odometry with visual hallucination. *IEEE Robotics and Automation Letters*, 5(2):1672–1679, 2020.

[29] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[30] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[31] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020.

[32] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020.

[33] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Radar object detection using cross-modal supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 504–513, 2021.

[34] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5363–5371, 2017.

[35] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021.

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

[37] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for

point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019.

[38] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020.

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

[40] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8573, 2022.

[41] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganiere. Raddet: Range-azimuth-doppler based radar object detection for dynamic road users. In *2021 18th Conference on Robots and Vision (CRV)*, pages 95–102. IEEE, 2021.

[42] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022.

[43] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11079–11087, 2020.

[44] Lianqing Zheng, Zhixiong Ma, Xichan Zhu, Bin Tan, Sen Li, Kai Long, Weiqi Sun, Sihan Chen, Lu Zhang, Mengyue Wan, Libo Huang, and Jie Bai. Tj4dradset: A 4d radar dataset for autonomous driving. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 493–498, 2022. doi: 10.1109/ITSC55140.2022.9922539.

[45] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3555–3562, May 2021. doi: 10.1609/aaai.v35i4.16470. URL https://ojs.aaai.org/index.php/AAAI/article/view/16470.

[46] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021.

[47] Wu Zheng, Mingxuan Hong, Li Jiang, and Chi-Wing Fu. Boosting 3d object detection by simulating multimodality on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13638–13647, 2022.

[48] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.