ImLiDAR: Cross-Sensor Dynamic Message Propagation Network for 3D Object Detection

Yiyang Shen, Rongwei Yu, Peng Wu, Haoran Xie, *Senior Member, IEEE*, Lina Gong, Jing Qin, and Mingqiang Wei, *Senior Member, IEEE*

Abstract—LiDAR and camera, as two different sensors, supply geometric (point clouds) and semantic (RGB images) information of 3D scenes. However, it is still challenging for existing methods to fuse data from the two cross sensors, making them complementary for quality 3D object detection (3OD). We propose ImLiDAR, a new 3OD paradigm to narrow the cross-sensor discrepancies by progressively fusing the multi-scale features of camera Images and LiDAR point clouds. ImLiDAR enables to provide the detection head with cross-sensor yet robustly fused features. To achieve this, two core designs exist in ImLiDAR. First, we propose a cross-sensor dynamic message propagation module to combine the best of the multi-scale image and point features. Second, we raise a direct set prediction problem that allows designing an effective set-based detector to tackle the inconsistency of the classification and localization confidences, and the sensitivity of hand-tuned hyperparameters. Besides, the novel set-based detector can be detachable and easily integrated into various detection networks. Comparisons on both the KITTI and SUN-RGBD datasets show clear visual and numerical improvements of our ImLiDAR over twenty-three state-of-the-art 3OD methods.

Index Terms—ImLiDAR, 3D object detection, Cross sensors, Dynamic message propagation, Set-based detector.

I. INTRODUCTION

With the rapid development of autonomous driving, profound progress has been made in 3D object detection from monocular images [1]–[3], stereo cameras [4]–[6] and LiDAR point clouds [7]–[9]. Among these sensors, LiDAR provides depth and geometric structure information, but its sparsity is causing degraded performance on small-object and long-range perception. Camera images usually possess richer color and semantic information to perceive objects while they lack the depth information for accurate 3D localization. This provides an intriguing and practical question of how to present effective fusion of camera images and LiDAR point clouds for quality 3D object detection.

Recent years have witnessed considerable efforts of information fusion from cross sensors. However, it is still non-trivial to fuse the representations of camera images and LiDAR

Y. Shen and R. Yu are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, WuHan, China (e-mail: shenyiyang114@gmail.com, roewe.yu@whu.edu.cn).

P. Wu, L. Gong and M. Wei are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: wupengupon1@gmail.com; gonglina@nuaa.edu.cn; mingqiang.wei@gmail.com).

H. Xie is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, China (e-mail: hrxie@ln.edu.hk).

J. Qin is with the School of Nursing, The Polytechnic University of Hong Kong, Hong Kong, China (e-mail: harry.qin@polyu.edu.hk).

point clouds, due to their extremely different data characteristics. According to different ways of fusion, existing methods are divided into three categories. (1) Image-level methods adopt a cascade strategy by exploiting image annotations to fuse camera images and point clouds in different stages [10], [13], [14]. (2) BEV-level methods jointly reason over camera images and the generated BEV data from point clouds [11], [15]–[17]. (3) Feature-level methods attempt to directly fuse camera images and point clouds by sharing the extracted features between 2D and 3D networks [12], [18]–[22]. Among the three categories, the image-level methods require image annotations, i.e., 2D bounding boxes, and their performance is easily restricted by each single stage. And the BEV-level methods require generating the BEV data via perspective projection and voxelization; as a result, they usually establish a relatively coarse correspondence between the image and voxel features, and suffer from the loss of 3D information when converting point clouds into the BEV data.

A recent trend in cross-sensor 3D object detection, which we call feature-level fusion, is to directly fuse the image and point features extracted by 2D and 3D networks. However, when encountering challenging (hard) cases such as under-exposed and occluded instances, the performance of existing feature-level fusion methods is still far from satisfactory, due to two major problems. First, their straightforward fusion strategies assign no weights or coarse weights learned within limited receptive fields to different features. During fusion, there are no crucial clues to keep the original geometric structure without the information loss and avoid introducing new interfering information. Second, they heavily rely on the post-processing step of non-maximum suppression (NMS) to remove redundant and near-duplicate results, leading to the inconsistency of the classification confidence and localization confidence. Moreover, NMS requires multiple hand-tuned hyperparameters. For example, a lower threshold of the predefined IoU misses highly overlapped objects while a higher one introduces more false positives.

To address the above issues, we present a novel cross-sensor dynamic message propagation network, dubbed ImLiDAR, which contains two novel designs for quality 3D object detection. First, we propose a cross-sensor dynamic message propagation (CDMP) module. CDMP targets effective and efficient fusion of camera images and LiDAR point clouds with two key dynamic properties. They are dynamically sampling feature nodes for capturing rich geometric information and filtering harmful semantic information from data of two cross sensors, and dynamically predicting filter weights and affinity

| (a) Image | (b) F-Pointnet [10] | (c) MMF [11] | (d) EPNet [12] | (e) Ours |

Fig. 1. Visualization results by different categories of fusion methods, i.e., the image-level, BEV-level and feature-level methods. For the real outdoor scenes where the underexposed instance (row 1) and the occluded instance (row 2) exist, ImLiDAR shows clear visual improvements over the three prevailing cross-sensor methods. From (a) to (f): (a) the camera image, and (b-e) the 3OD results of F-Pointnet, MMF, EPNet and our ImLiDAR. Red box and green box denote GT and the predicted bounding box, respectively.

matrices as clues for propagating useful image features to enrich the original point features. Second, we formulate a direct set prediction problem and accordingly design a set-based detector to select high-quality 3D bounding boxes with both high classification and localization confidence. Such a set-based detector can avoid the post-processing of NMS, and it can be easily implemented in various detection networks.

Although ImLiDAR does not need additional image annotations, the complex BEV data, and the commonly used NMS post-processing step, it usually exhibits better performance over all prevailing cross-sensor methods. For example, when encountering real outdoor scenes (see Fig. 1), cutting-edge models suffer from the condition of poor illuminations and heavy occlusions, while ImLiDAR will not. More results, in terms of visual quality and quantitative accuracy, will be found in Section V. In summary, our main contributions are three-fold:

- We propose a novel cross-sensor 3D object detection paradigm, namely ImLiDAR, with two core designs, i.e., a cross-sensor dynamic message propagation module and a set-based detector. Extensive experiments in both the outdoor and indoor scenes show clear improvements of ImLiDAR over both the LiDAR-based and cross-sensor methods.
- We propose a cross-sensor dynamic message propagation module, combining the best of image and point features without the BEV data or 2D bounding boxes.
- We propose a set-based detector to guarantee the consistency between the classification confidence and localization confidence and select high-quality proposals without non-maximum suppression.

II. RELATED WORK

We first review image-based, point cloud-based and cross-sensor methods for 3D object detection. Subsequently, we introduce the recent advance of graph neural networks.

A. Image-based 3D Object Detector

Many methods focus on camera images, e.g., monocular [23]–[25] and stereo images [2], [5], [26]. They take RGB images as input to generate 2D bounding boxes and and estimate the corresponding 3D bounding boxes [1], [3], [27]. Another way is to conduct depth estimation and design multi-level fusion methods to fuse image features with the depth maps [2], [28], [29]. Particularly, DSGN [30] provides a simple and effective one-stage stereo-based 3D detection pipeline that jointly estimates the depth and detects 3D objects. However, the performance of the image-based methods is bounded due to the absence of depth information.

B. Point Cloud-based 3D Object Detector

3D object detection methods usually exploit LiDAR point clouds, which provide spatial geometry information to locate the objects. They can be divided into voxel-based [9], [31], [32], and point-based methods [12], [33]–[37]. Voxel-based models [9], [38]–[40] group point clouds into regular voxels and employ 3D CNNs to learn voxel features for the generation of 3D bounding boxes. To remove 3D CNN layers, PointPillars [31] elongates voxels into pillars that are arrayed in a BEV perspective. Point-based approaches [12], [33], [41]–[45] sample a fixed number of points as key points via point set abstraction, and aggregate point features around key points with ball query. Recently, most of point-based methods [12], [33], [43]–[45] formulate a two-stage detection framework, which consists of a region proposal network (RPN) to predict the foreground points and generate 3D proposals, and a refinement network to refine the coarse bounding boxes from RPN. However, all point cloud-based methods suffer from the sparsity of points.

C. Cross-sensor 3D Object Detector

In realistic self-driving situations, it is insufficient to perform object detection through single types of sensors. Thus, many cross-sensor techniques are proposed to alleviate the shortcomings of single-sensor data. Current studies are categorized into three groups based on different ways of fusion.

Image-level fusion. Image-level approaches usually exploit camera images in the first stage and reason in LiDAR point clouds only at the second stage [10], [13], [14], [46]. F-PointNet [10] projects 2D detection results to 3D space to generate 3D frustums and then adopt PointNet [41] to regress

corresponding 3D boxes from the frustums. V2-SENet [13] focuses on utilizing the front view images and frustum point clouds to generate 3D detection results. For these wisdom, the overall performance is bounded by each stage, since they still depend on single sensors.

BEV-level fusion. MV3D [16] is the pioneering attempt to fuse the bird's eye view (BEV) and front view (FV) representations of cross-sensor data. The follow-up BEV-level methods [11], [15]–[17] remove the FV branch and only reason over the BEV data and camera images. Confuse [15] designs a continuous fusion layer to achieve the voxel-wise alignment between the BEV and image feature maps. However, the complex BEV data generation inevitably causes computation costs and the information loss.

Feature-level fusion. A new fashion trend is to fuse each point with the corresponding image pixel instead of fusing the BEV data and camera images [12], [18]–[22]. For example, EPNet [12] designs an end-to-end framework with LiDAR-guided image fusion modules, which assign coarse weights to image features to guide the feature fusion. Similarly, attention fusion modules [18] and gated fusion modules [22] are developed to produce fused features. These methods do not require the generation of 2D bounding boxes and complex BEV, but their fusion manners cannot fully exploit the complementary information of LiDAR point clouds and camera images.

Please note that the proposed ImLiDAR is different from all the above cross-sensor approaches largely, since it combines the best of multi-scale features of camera images and LiDAR point clouds, and does not require any post-processing step of NMS.

D. Graph Neural Networks

Graph neural network [47] have exhibited its powerful ability in many vision tasks, because of their robust capacity of non-local feature aggregation. However, these local-connected graphs can only capture partial long-range contextual information needed for complex vision tasks such as segmentation [48]–[50] and detection [35], [51]–[53]. Differently, Zhang et al. [54] propose an efficient dynamic graph learning model based on the message propagation mechanism to solve this problem. In this work, we also design a cross-sensor dynamic message propagation (CDMP) module to effectively fuse the LiDAR point features with the corresponding image features, resulting in more comprehensive and discriminative feature representations.

III. OVERVIEW

Cross-sensor fusion has shown its superiority in various applications. Primarily, point clouds provide geometric structure information of 3D scenes, and camera images further enrich the point clouds by fulfilling semantic information of the 3D scenes. To effectively fuse the image and point features in multiple scales for quality 3D object detection, we propose a new 3D object detection paradigm, called ImLiDAR. The top level of ImLiDAR, consisting of a two-stream region proposal network (RPN), a set-based detector, and a refinement network, is outlined in Fig. 2.

Two-stream region proposal network (RPN). The two-stream RPN consists of a point stream, an image stream, and cross-sensor dynamic message propagation (CDMP) modules. The two-stream RPN combines the best of image and point features in multiple scales for 3D object detection, as discussed in Section IV-A.

Set-based detector. Considering that NMS will degrade the detection performance, we newly design a set-based detector to filter out redundant and near-duplicate results to avoid such an NMS step in Section IV-B.

Refinement network. The proposals produced by the set-based detector are fed into the refinement network for further box refinement, leading to more precise 3D object detection results, as discussed in Section IV-C.

A. Preliminary

Despite the success of graph networks in 2D/3D single-sensor object detection tasks, the attempt to combine advantages from both point clouds and camera images remains scarce. In Section IV-A, we introduce a cross-sensor dynamic message propagation (CDMP) module to fuse multi-scale features of camera images and LiDAR point clouds. Before going into the details, we will give some basic knowledge of graph message passing used in CDMP.

Graph message passing. Given an input feature map interpreted as the latent feature vectors $H = \{h_i\}_{i=1}^{N}$, where N denotes the number of pixels, the goal of the message passing mechanism is to refine the latent feature vectors H by extracting hidden structured information among the feature vectors at different pixel locations. Therefore, the common message passing network usually converts such feature map into a graph domain by constructing a feature graph $G = \{V, E, A\}$, where V denotes the node set represented by the above latent feature vectors, i.e., $V = \{h_i\}_{i=1}^{N}$, E is the edge set, and $A \in R^{N \times N}$ is a binary or learnable matrix with self-loops describing the connections between nodes. The common message passing phase, composed of a message calculation step M^t and a message updating step U^t, takes T iterations. For the latent feature vector $h_i^{(t)}$ at the iteration t, it dynamically samples K nodes to connect and form a local field $v_i \subset V, v_i \in R^{K \times C}, K \ll N$, where C denotes the dimension of the vector. The message calculation step for the node i is defined as

$$m_i^{(t+1)} = M^t(A_{i,j}, \{h_1^{(t)}, ..., h_K^{(t)}\}, w_j)$$
$$= \sum_{j \in \mathcal{N}(i)} A_{i,j} h_j^{(t)} w_j \qquad (1)$$

where $A_{i,j}$ denotes the connection relationship between latent nodes $h_i^{(t)}$ and $h_j^{(t)}$, $\mathcal{N}(i)$ represents a self-included neighborhood of the node $h_i^{(t)}$, and $w_j \in R^{C \times C}$ is a transformation matrix for message calculation on the hidden node $h_j^{(t)}$. Then the message updating step U^t obtains the updated latent feature vector $h_i^{(t+1)}$ with a linear combination of the calculated message $m_i^{(t+1)}$ and the original feature vector $h_i^{(t)}$ at the node position i:

$$h_i^{(t+1)} = U^t(h_i^{(t)}, m_i^{(t+1)}) = \sigma(h_i^{(t)} + \alpha_i^m m_i^{(t+1)}) \qquad (2)$$

Fig. 2. The pipeline of our ImLiDAR. ImLiDAR consists of three cascaded branches, i.e., the two-stream RPN, the set-based detector, and the refinement network. Concretely, the two-stream RPN contains an image stream for extracting image features, a point stream for extracting point features, and well-designed CDMP modules to fuse the geometric point features and semantic image features for enhancing feature representations. Then the set-based detector attempts to select high-quality 3D proposals without the NMS post-processing, and feeds them into the refinement network for further box refinement, leading to more precise 3D object detection results.

Fig. 3. Illustration of the cross-sensor dynamic message propagation (CDMP) module. CDMP first dynamically samples context-aware nodes in the LiDAR point features (a) and point-wise image features (c), which are extracted by projecting the source LiDAR (a) onto the image plane (b). Then it predicts hybrid image-dependent filter weights and affinity matrices as clues for propagating semantic information to enrich the point features. (d) demonstrates that our ImLiDAR can effectively fuse the image and point features, leading to significant improvement of the 3D object detection performance. Note that white, red and green boxes represent the ground truth, predicted bounding boxes of the LiDAR-based detector [33] and ours (LiDAR+Image), respectively.

where α_i^m denotes a learnable parameter to scale the message, and $\sigma(.)$ is a non-linearity function, e.g., ReLU. By propagating the message on each node with T steps, the module finally obtains the refined features. Especially, we fuse the image and point features via the graph message passing mechanism as:

$$h_i^{(t+1)} = \sigma(h_i^{(t)} || \alpha_i^m m_i^{(t+1)}) \qquad (3)$$

where $||$ denotes the channel-wise concatenation operation.

IV. IMLIDAR

ImLiDAR consists of a two-stream RPN, a set-based detector, a refinement network, and the defined loss function.

A. Two-stream RPN

Our two-stream RPN consists of a point stream, an image stream, and cross-sensor dynamic message propagation (CDMP) modules in Fig. 2. The point stream and image stream are designed for extracting multi-scale geometric point features and semantic image features, respectively. The CDMP modules are employed to fuse the image and point features in different scales, resulting in more robust and discriminative representations.

Image stream. The image stream, depicted in Fig. 2, takes camera images as input to extract multi-scale semantic image features. Concretely, the architecture of the image stream consists of four feature extract blocks (FEBs), which both include two 3×3 convolution layers followed by a batch normalization layer and a ReLU activation function. F_i^k $(k = 1, 2, 3, 4)$ denotes the multi-scale features extracted from four FEBs, which provide adequate semantic information to enrich the point features in different scales. At the end of the image stream, we feed these multi-scale image features into four parallel transposed convolution layers to obtain image features with the same size as the original image, which are used to enrich the final point features, resulting in the generation of more high-quality proposals.

Point stream. For the point stream, we employ PointNet++ [42] as our backbone network. The point stream takes LiDAR point clouds as input and utilizes four set-abstraction (SA) modules with multi-scale grouping to subsample points into groups with the sizes of 4096, 1024, 256, 64, and four feature propagation (FP) modules to recover the point resolution. Especially, F_s^k $(k = 1, 2, 3, 4)$ and F_p^k $(k = 1, 2, 3, 4)$ represent the outputs of SA and FP layers in different scales, respectively. With the aid of CDMP (1×1) modules, we can

Fig. 4. Illustration of CDMP in a single scale pattern. We first project the LiDAR points onto the 2D camera image to obtain the corresponding point-wise image features. Then we sample the dynamic nodes from the image and point feature graphs, and predict the filter weights and affinity matrices from image features to propagate the semantic message.

effectively fuse the point features F_s^k with the image features F_i^k at different levels. Further, we apply the CDMP (1×4) module at the end of the point stream, which enables the point feature vectors F_p^4 to possess different level semantic information from the image features F_i^k $(k = 1, 2, 3, 4)$. Similar to PointRCNN [33], given the final point features, we first append a box regression head for 3D proposal generation and a segmentation head for foreground point segmentation. Moreover, we append an additional match head to estimate the match score for the set-based detector. The match scores mean that the probability of each predicted 3D bounding box is retained by the set-based detector.

Cross-sensor dynamic message propagation (CDMP). To combine the best of multi-scale image and point features, we design a novel CDMP module. The more detailed scheme of CDMP is further depicted in Fig. 3. In particular, it includes three steps: (1) generating the fine-grained point-wise correspondence and point-wise image features; (2) dynamically sampling on image and point feature graphs to select the most object-relevant nodes; and (3) dynamically predicting hybrid filter weights and affinity matrices for message propagation.

The CDMP module in a single scale pattern is shown in Fig. 4. First, we project the LiDAR points onto the 2D camera image based on the calibration matrix, which is usually provided by the benchmark datasets, to generate a finer point-wise correspondence between LiDAR points and camera images. Concretely, for a particular point $p(x, y, z)$ in the point cloud, we obtain its corresponding position $p'(x, y)$ in the camera image. Then we input both the image feature map and sampling position p' into the bilinear interpolation to produce point-wise image features at the continuous coordinates.

Based on the fine-grained point-wise image features, CDMP possesses two novel dynamic properties, i.e., dynamically sampling feature nodes and dynamically predicting filter weights and affinity matrices. We regard the image feature map and point feature map as two graphs. For each node v_i in the point feature node set $V = \{v_i\}_1^N$, where N is the total number of pixels, the sampling number K determines its receptive field. To adaptively sample relevant nodes for v_i, we denote $\Delta d_{i,j} \in R^D$ as the predicted walk, which makes the module walk around to sample the relevant node $v_{i,j}$ with $j \in \mathcal{N}(i)$, where $\mathcal{N}(i)$ contains K number of sampled nodes for v_i and $D = 2$ represents the space dimension along the height and width. Especially, we describe such node walk as a matrix

transformation:

$$\Delta d_{i,j} = W_{i,j} h_i + b_{i,j} \tag{4}$$

where $W_{i,j}$ and $b_{i,j}$ both are matrix transformation parameters learned on point graph nodes, and h_i denotes the latent vector for v_i. Due to the difference between point and image modalities, we generate another walk $\Delta \bar{d}_{i,j}$ for uniformly neighboring nodes on the image feature graph. The dynamic walk $\Delta d_{i,j}$ for each point feature node, as well as $\Delta \bar{d}_{i,j}$ for each image feature node, is generated by applying 3×3 convolution layers according to Equation 4. Based on the above random walks, we adopt the deformable convolution [55] to obtain dynamic sample nodes $\hat{v}_{i,j}^l$ and $\bar{v}_{i,j}^l$ from point and image feature graphs at the level l, respectively. This dynamically sampling operation enables to efficiently gather long-range context information and only select a subset of the most important feature nodes in the two graphs.

Based on the sampled image feature nodes $\bar{v}_{i,j}^l$, we apply 3×3 convolution layers to generate the affinity matrix $A_{i,j}^l$ and transformation matrix $w_{i,j}^l$, which are used as clues for propagating the useful image features to enhance the point features, which can be formulated as:

$$\{A_{i,j}^l; w_{i,j}^l\} = \bar{W}_{i,j}^l \bar{v}_{i,j}^l + \bar{b}_{i,j}^l \tag{5}$$

where $\bar{W}_{i,j}^l$ and $\bar{b}_{i,j}^l$ are matrix transformation parameters generated by dynamic sample image nodes. Then the calculated message is summarized as:

$$\begin{aligned} m_i^{t+1} &= \sum_{l \in L} \sum_{j \in \mathcal{N}(i)} \beta_l A_{i,j}^l \hat{h}_j^{l,(t)} w_{i,j}^l \\ &= \sum_{l \in L} \sum_{j \in \mathcal{N}(i)} \beta_l A_{i,j}^l \delta(\hat{h}_j^{l,(t)} | V; j; \Delta d_{i,j}) w_{i,j}^l \end{aligned} \tag{6}$$

where L denotes the layer from different level stages, $\hat{h}_j^{l,(t)}$ is the latent vector for dynamic point feature nodes $\hat{v}_{i,j}^l$ calculated by $\Delta d_{i,j}$ over the whole nodes V of the point graph. β_l and $\delta(.)$ represent the balance weight and bilinear sampler, respectively. In CDMP (1×4) module, all messages are calculated as Equation 6 using group convolution layers and concatenated into a 1×1 convolution layer. Then the result is concatenated again with the original point features to obtain the final refined point features with semantic image information, as described in Equation 3.

B. Set-based Detector

Most of existing 3D object detectors predict a larger number of bounding boxes than the number of the real objects in the scene. In view of such fact, non-maximum suppression (NMS) is often a necessary post-processing step.

NMS includes two parts: (1) Selecting 3D bounding boxes with the maximum scores after ranking the proposals according to the classification scores. However, NMS may filter out the bounding boxes with low classification scores but large overlaps; this leads to the inconsistency of the classification confidence and localization confidence. (2) Removing any 3D bounding box which possesses an overlap greater than a predefined IoU threshold. It makes the current detectors with a

dilemma: a lower threshold leads to missing highly overlapped objects while a higher one introduces more false positives in crowded scenes.

Motivated by DETR [56], we design a set-based detector to address the above problems caused by NMS. We design three sets of bounding boxes: the set of predicted 3D bounding boxes $B_{pre} = \{\bar{b}\}_1^M$ from RPN, the set of GT bounding boxes $B_{gt} = \{b\}_1^N$, and the set of output bounding boxes $B_{out} = \{\tilde{b}\}_1^N$, where M and N represent the number of bounding boxes, and $M \gg N$. The number N is larger than or equal to the number of the real objects in the scene. To obtain the high-quality bounding box set B_{out}, we perform a bipartite graph matching, which is simpler and more effective than NMS. We compute the match cost for each predicted box $\bar{b}$ with each ground truth box b.

Bipartite matching. We define a match cost for a pair of predicted box $\bar{b}$ and GT box b, which is formulated as:

$$C_{match}(\bar{b}, b) = -\log(c \times \frac{Area(\bar{b} \cap b)}{Area(\bar{b} \cup b)}) \tag{7}$$

where c denotes the classification confidence for $\bar{b}$. We compute the optimal bipartite matching between all the predicted boxes $\bar{b}$ and GT boxes b using the Hungarian algorithm [57]. Therefore, each GT box b is successfully matched with a predicted 3D bounding box $\bar{b}$ with both large overlaps and high classification possibilities. We regard these matched predicted 3D bounding boxes as positive samples and the others as negative samples. During the training step, these positive samples constitute the output high-quality bounding boxes B_{out}, which are fed into the refinement network. We also generate a match label vector $\tilde{M} \in R^{M \times 1}$ according to the match results. Particularly, if the predicted 3D bounding box is a positive sample, its corresponding value of $\tilde{M}$ will be set to 1, and the others are set to 0. Then we adopt the focal loss [58] between the match label and match score from the match head of RPN:

$$L_{SD} = -\alpha(1 - c_m)^\gamma \log c_m \tag{8}$$

where c_m represents the probability of the predicted box $\bar{b}$ is the positive sample, and $\alpha = 0.25, \gamma = 2$ are kept as in [58]. During the testing step, we directly select 3D bounding boxes with the highest match score to constitute B_{out} for the refinement network.

C. Refinement Network

We feed the set of proposals B_{out} from the set-based detector into the refinement network to refine the box locations and orientations for final predictions. Similar to PointRCNN [33], for each input proposal, we randomly select 512 points as its 3D RoI feature descriptor. For those proposals with less than 512 points, the descriptor is padded with zeros. The refinement network is composed of three SA layers for extracting a compact global descriptor for each 3D ROI, and two 1×1 convolution layers as two detection heads for classifying and regressing the final 3D objects.

D. Total Loss Function

We present the loss functions. We adopt a multi-task loss function for jointly optimizing the two-stream RPN, the set-based detector and the refinement network, which can be defined as:

$$L_{total} = L_{rpn} + L_{rcnn} + \lambda L_{SD} \tag{9}$$

$$\{L_{rpn}; L_{rcnn}\} = L_{cls} + L_{reg} \tag{10}$$

where L_{rpn} and L_{rcnn} represent the training objective for the two-stream RPN and the refinement network. They both contain a classification loss and a regression loss. Concretely, we adopt the focal loss [58] as the classification loss to balance the positive and negative samples as:

$$L_{cls} = -\alpha(1 - c_t)^\gamma \log c_t \tag{11}$$

where c_t represents the probability of the point in consideration belonging to the ground truth category. And we keep the default settings $\alpha = 0.25, \gamma = 2$ as suggested by [58].

In the LiDAR coordinate system, a 3D bounding box is represented as $(x, y, z, h, w, l, \theta)$, where (x, y, z) is the object center location, (h, w, l) is the object size, and θ denotes the object orientation. Following PointRCNN [33], we adopt the bin-based regression loss as our regression loss function to estimate 3D bounding boxes of objects. Concretely, we split the neighboring area of each foreground point into several bins. The bin-based loss first predicts which bin $\tilde{b}_u$ the center point belongs to, and regresses the residual offset $\tilde{r}_u$ within the bin. Thus, the regression loss is formulated as:

$$L_{reg} = \sum_{u \in x,z,\theta} E(\tilde{b}_u, b_u) + \sum_{u \in x,y,z,h,w,l,\theta} S(\tilde{r}_u, r_u) \tag{12}$$

where E and S denote the cross entropy loss and the smooth L1 loss, respectively. b_u and r_u denote the ground truth of the bins and the residual offsets.

V. EXPERIMENTS

Dataset and metric. We conduct experiments on the KITTI dataset [79] and the SUN-RGBD dataset [80]. KITTI is an outdoor standard benchmark dataset, which consists of 7,481 frames for training and 7,518 frames for testing. Following the protocol of [10], [33], we split the 7,481 frames into 3,712 frames for training and 3,769 frames for validation. Three levels of difficulty are defined in the benchmark according to size, occlusion, and truncation, i.e., Easy, Moderate, and Hard. Besides, our results are reported for the car, pedestrian and cyclist categories and the IoU thresholds are set to 0.7, 0.5, and 0.5, respectively. SUN-RGBD is an indoor benchmark dataset, which includes 10,335 images with 700 annotated object categories, including 5,285 images for training and 5,050 images for testing. The IoU thresholds for all ten categories are set to 0.25. The common Average Precision (AP) is used as our evaluation metric following the official evaluation protocol of the KITTI dataset and the SUN-RGBD dataset. Especially, the 40 recall positions-based metric $AP|R40$ has been utilized by the KITTI dataset instead of $AP|R11$ as before.

Implementation details. Each LiDAR point cloud is cropped to the range of [-40, 40], [-1, 3], [0, 70.4] meters

TABLE I
QUANTITATIVE COMPARISONS WITH STATE-OF-THE-ART METHODS ON THE TEST SET OF THE KITTI DATASET.

Method	Modality	Car(IoU=0.7)				Pedestrian(IoU=0.5)				Cyclist(IoU=0.5)			
		Easy	Moderate	Hard	mAP	Easy	Moderate	Hard	mAP	Easy	Moderate	Hard	mAP
SECOND [39]	LiDAR	87.44	79.46	73.97	80.29	-	-	-	-	-	-	-	-
PointPillars [31]	LiDAR	82.58	74.31	68.99	75.29	51.45	41.92	38.89	44.09	77.10	58.65	51.92	62.56
PointRCNN [33]	LiDAR	86.96	75.64	70.70	77.76	47.98	39.37	36.01	41.12	74.96	58.82	52.53	62.10
STD [59]	LiDAR	87.95	79.71	75.09	80.91	53.29	42.47	38.35	44.70	78.69	61.59	55.30	65.19
3DSSD [60]	LiDAR	88.36	79.57	74.55	80.82	54.64	44.27	40.23	46.38	82.48	64.10	56.90	67.82
SA-SSD [61]	LiDAR	88.75	79.79	74.16	81.03	-	-	-	-	-	-	-	-
PV-RCNN [43]	LiDAR	90.25	81.43	76.82	82.83	52.17	43.29	40.29	45.25	78.60	63.71	57.65	66.65
MGAF-3DSSD [62]	LiDAR	88.16	79.68	72.39	80.07	50.65	43.09	39.65	44.46	80.64	63.43	55.15	66.40
HVPR [63]	LiDAR	86.38	77.92	73.04	79.11	-	-	-	-	-	-	-	-
CIA-SSD [64]	LiDAR	89.59	80.28	72.87	80.91	-	-	-	-	-	-	-	-
CT3D [65]	LiDAR	87.83	81.77	77.16	82.25	-	-	-	-	-	-	-	-
SASA [66]	LiDAR	88.76	82.16	77.16	82.69	-	-	-	-	-	-	-	-
SVGA-Net [67]	LiDAR	87.33	80.47	75.91	81.23	48.48	40.39	37.92	42.26	78.58	62.28	54.88	65.24
MV3D [16]	LiDAR + RGB	74.97	63.63	54.00	64.20	-	-	-	-	-	-	-	-
Confuse [15]	LiDAR + RGB	83.68	68.78	61.67	71.38	-	-	-	-	-	-	-	-
F-Pointnet [10]	LiDAR + RGB	82.19	69.79	60.59	70.86	50.53	42.15	38.08	43.59	72.27	56.12	49.01	59.13
MMF [11]	LiDAR + RGB	88.40	77.43	70.22	78.68	-	-	-	-	-	-	-	-
3D-CVF [22]	LiDAR + RGB	89.20	80.05	73.11	80.79	-	-	-	-	-	-	-	-
PointPainting [68]	LiDAR + RGB	82.11	71.70	67.08	73.63	50.32	40.97	37.84	43.05	77.63	63.78	55.89	65.77
EPNet [12]	LiDAR + RGB	89.81	79.28	74.59	81.23	-	-	-	-	-	-	-	-
Fast-CLOCs [69]	LiDAR + RGB	89.10	80.35	76.99	82.14	52.10	42.72	39.08	44.63	82.83	65.31	57.43	68.53
Focals Conv [70]	LiDAR + RGB	90.55	82.28	77.59	83.47	-	-	-	-	-	-	-	-
CAT-Det [71]	LiDAR + RGB	89.87	81.32	76.68	82.62	54.26	45.44	41.94	47.21	83.68	68.81	61.45	71.31
ImLiDAR	LiDAR + RGB	90.98	83.23	77.67	83.96	55.38	46.26	42.38	48.01	84.22	68.89	61.80	71.63

TABLE II
QUANTITATIVE COMPARISONS WITH THE STATE-OF-THE-ART METHODS ON THE TEST SET OF THE SUN-RGBD DATASET.

Method	Modality	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
VoteNet [34]	LiDAR	74.4	83.0	28.8	75.3	22.0	29.8	62.2	64.0	47.3	90.1	57.7
MLCVNet [72]	LiDAR	79.2	85.8	31.9	75.8	26.5	31.3	61.5	66.3	50.4	89.1	59.8
H3DNet [73]	LiDAR	73.8	85.6	31.0	76.7	29.6	33.4	65.5	66.5	50.8	88.2	60.1
HGNet [74]	LiDAR	78.0	84.5	35.7	75.2	34.3	37.6	61.7	65.7	51.6	91.1	61.6
MLCVNet++ [75]	LiDAR	79.3	85.3	36.5	77.1	28.7	31.6	61.4	68.3	50.7	90.0	60.9
Group-Free-3D [76]	LiDAR	80.0	**87.8**	32.5	**79.4**	32.6	36.0	**66.7**	**70.0**	53.8	91.1	63.0
Pointformer [36]	LiDAR	80.1	84.3	32.0	76.2	27.0	37.4	64.0	64.9	51.5	**92.2**	61.1
DSS [38]	LiDAR + RGB	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
2D-driven [77]	LiDAR + RGB	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
COG [78]	LiDAR + RGB	58.3	63.7	31.8	62.2	45.2	15.5	27.4	51.0	51.3	70.1	47.6
PointFusion [14]	LiDAR + RGB	37.3	68.6	**37.7**	55.1	17.2	24.0	32.3	53.8	31.0	83.8	44.1
F-PointNet [10]	LiDAR + RGB	43.3	81.1	33.3	64.2	24.7	32.0	58.1	61.1	51.1	90.9	54.0
EPNet [12]	LiDAR + RGB	75.4	85.2	35.4	75.0	26.1	31.3	62.0	67.2	52.1	88.2	59.8
MBDF-Net [18]	LiDAR + RGB	**81.5**	84.7	33.0	77.3	31.2	29.0	57.7	65.6	49.9	85.5	59.5
ImLiDAR	LiDAR + RGB	80.3	85.3	35.7	**79.4**	**35.4**	**38.4**	65.9	69.9	**54.2**	91.6	**63.6**

along (X, Y, Z) axes in the camera coordinate, respectively. The orientation of θ is set to the range of $[-\pi, \pi]$. Similar to PointRCNN [33], we subsample 16,384 points from each 3D point cloud scene as the inputs of the point stream. Then four set abstraction layers with multi-scale grouping are used to subsample the aforementioned input points into groups with the sizes of 4096, 1024, 256, 64 respectively, and four feature propagation layers are employed to obtain the per-point feature vectors. The image stream takes camera images of the size of 1280 × 384 as input.

Training details. ImLiDAR is trained by SGD with an initial learning rate being 0.002, the momentum being 0.9, and the weight decay being 0.001 respectively. We train the model for around 50 epochs on an Nvidia GeForce RTX 3090 GPU with a batch size of 2 in an end-to-end manner. The balancing weight λ in the loss function is set to 1.

A. Comparison with the State-of-the-Arts

We evaluate our ImLiDAR with state-of-the-art 3D object detection methods on the KITTI test set (see Table I) and the SUN-RGBD test set (see Table II). As shown in Table I, the point cloud-based methods outperform most of the cross-sensor methods, indicating that fusing the representations of camera images and LiDAR point clouds remains a challenging task. While ImLiDAR achieves remarkable results over the state-of-the-art methods on all three categories. For the car category, we improve the baseline PointRCNN [33] by 6.20% on the mAP metric, and ImLiDAR outperforms all point cloud-based methods on the mAP metric. Further, ImLiDAR surpasses all the cross-sensor methods by a large margin.

Fig. 5. Qualitative results on the val set of the KITTI dataset. We show our detected results of four different scenes in (a)–(d), in which car, pedestrian, and cyclist are shown in green, blue, and yellow, respectively. Note that all ground truth bounding boxes are shown in red.

For example, ImLiDAR outperforms EPNet [12] by 1.17%, 3.95%, and 3.08% on the easy, moderate, and hard metrics respectively, which demonstrates the superiority of the CDMP module. It is noteworthy that ImLiDAR also ranks the first on the pedestrian and cyclist categories, although most of existing approaches do not provide evaluations on these two categories. The small or partial instances in these categories require more context information with large receptive fields, which can be fully gained by the CDMP module. We also provide qualitative detection results on the KITTI validation dataset in Fig. 5, from which we can see that ImLiDAR can detect more hard examples, even occluded and distant instances in crowded scenes.

To verify the effectiveness of all methods in the indoor scenes, we compare ImLiDAR with its competitors on the SUN-RGBD dataset in Table II. It is noteworthy that our ImLiDAR still outperforms its competitors. Especially, PointFusion [14] and F-PointNet [10] generate 2D bounding boxes from camera images using 2D detectors and output the 3D boxes in a cascading manner. While ImLiDAR does not add explicit supervision information (e.g., annotations of 2D detection boxes), and outperforms them by 19.5% and 9.6% mAP, respectively. EPNet [12] is a two-branch detector, which directly fuses point clouds and camera images, and the following work [18] designs a multi-branch fusion manner. Our ImLiDAR

outperforms EPNet [12] and MBDF-Net [18] by 3.8% and 4.1% in terms of 3D mAP. Such a large improvement verifies the superiority of our CDMP module over the other fusion schemes.

B. Ablation Study

We conduct extensive experiments on the KITTI validation dataset to evaluate the effectiveness of our CDMP module and the set-based detector.

Effectiveness of the CDMP module. We conduct some ablation experiments on the CDMP module. For fair comparisons, all the models adopt the same NMS procedure for filtering out low-quality proposals. Table III shows the results of different fusion modules. It is found that: (1) Simple concatenation (SC) and addition (AD) yield the decrease of 3D mAP 0.58% and 3.26% over the baseline, which indicates that such simple fusion manners cannot obtain more accurate 3D detection results than only using LiDAR data, and even worse. (2) The combination of LI and LI*, along with the combination of CDMP and CDMP*, performs better than single-scale fusion modules, which verifies the effectiveness of cross-sensor fusion in multiple scales and stages. (3) The combination of CDMP and CDMP* modules yields the most significant improvement of 5.81% in terms of 3D mAP, demonstrating that our CDMP modules actually provide a

TABLE III
ANALYSIS OF DIFFERENT FUSION MANNERS ON THE KITTI VAL SET (CAR). NOTE THAT SC AND AD REPRESENT THE SIMPLE CONCATENATION AND ADDITION OF POINT FEATURES AND POINT-WISE IMAGE FEATURES, RESPECTIVELY. CDMP AND CDMP* REPRESENT THE CDMP (1×1) MODULES IN THE SET ABSTRACTION LAYERS AND THE CDMP (1×4) MODULE IN THE LAST FEATURE PROPAGATION LAYER. LI AND LI* DENOTE LI-FUSION MODULES [12] IN SIMILAR ARCHITECTURES. IT SHOULD BE NOTED THAT NO IMAGE STREAM IS EMPLOYED FOR THE BASELINE (THE FIRST ROW), AND ALL MODELS ADOPT THE NMS PROCEDURE TO KEEP MORE ACCURATE BOUNDING BOXES.

Fusion						3D Detection				
SC	AD	LI	LI*	CDMP	CDMP*	Easy	Moderate	Hard	3D mAP	Gain
✗	✗	✗	✗	✗	✗	86.24	77.36	75.88	79.82	-
✓	✗	✗	✗	✗	✗	85.68	76.78	75.26	79.24	↓ 0.58
✗	✓	✗	✗	✗	✗	84.10	73.59	71.97	76.55	↓ 3.26
✗	✗	✓	✗	✗	✗	87.17	78.31	76.10	80.52	↑ 0.70
✗	✗	✗	✓	✗	✗	86.45	77.94	76.39	80.26	↑ 0.44
✗	✗	✓	✓	✗	✗	89.26	78.88	76.82	81.65	↑ 1.83
✗	✗	✗	✗	✓	✗	90.42	81.84	79.38	83.88	↑ 4.06
✗	✗	✗	✗	✗	✓	90.32	80.93	78.72	83.32	↑ 3.50
✗	✗	✗	✗	✓	✓	91.89	83.38	81.62	85.63	↑ 5.81

(a) 2D Image (b) LI-Fusion (c) CMMP

Fig. 6. Visualization of the semantic image feature. Foreground objects are highlighted with yellow rectangle boxes. The red rectangle box marks the interfering image features.

more effective way to fuse the multi-scale image and point features, thus leading to more quality 3D object detection results than the LI-Fusion modules [12].

Visualization of semantic image features. How to effectively deliver useful semantic information to enrich the point features is critical for cross-sensor 3D detection. We compare the image features learned from the LI-Fusion module [12] and our CDMP module as shown in Fig. 6. Although the LI-Fusion modules attempt to suppress the bad information by assigning coarse learnable weight matrices, they achieve a limited effect and still retain much interfering image information. Besides, the limited receptive fields will cause the loss of important semantic information. Comparatively, our CDMP modules can effectively gather long-range key context information and suppress harmful semantic information, thus leading to more accurate detection results.

Effectiveness of the set-based detector. In Table IV, the set-based detector is evaluated with CE and IoU loss functions. The balancing weights in IoU and CE loss functions are also set to 5. In the car category, the set-based detector yields a significant improvement of 1.50% mAP over the baseline, which indicates the superiority of our set-based detector in improving the 3D detection performance. Besides, we also

TABLE IV
ANALYSIS OF THE SET-BASED DETECTOR ON THE KITTI VAL SET.

Model				3D Detection (Car)				
IoU	CE	Set-based	NMS	Easy	Moderate	Hard	3D mAP	Gain
✗	✗	✗	✓	91.89	83.38	81.62	85.63	-
✓	✗	✗	✓	91.38	83.47	81.89	85.58	↓ 0.05
✓	✗	✗	✗	87.57	78.59	76.33	80.83	↓ 4.80
✗	✓	✗	✓	92.21	83.23	81.79	85.74	↑ 0.11
✗	✗	✗	✓	88.12	79.70	78.01	81.94	↓ 3.68
✗	✗	✓	✗	92.61	85.52	83.25	87.13	↑ 1.50
✗	✗	✓	✓	92.66	85.51	83.24	87.14	↑ 1.51
PointRCNN [33]				89.19	78.85	77.91	81.98	-
PointRCNN + Set-based				91.09	80.31	78.67	83.35	↑ 1.37
EPNet [12]				92.17	82.68	80.10	84.98	-
EPNet + Set-based				92.49	84.06	81.19	85.91	↑ 0.93

Fig. 7. Illustration of the ratio of keeping positive boxes with different classification confidence thresholds.

(a) EPNet (b) EPNet+Set-based

Fig. 8. Visualization results by EPNet [12] and the combination of EPNet [12] and our set-based detector. It is noteworthy that our set-based detector can filter out false positives, avoid missing distant objects, and even improve the predicted results.

TABLE V
ANALYSIS OF THE SET-BASED DETECTOR ON THE SUN-RGBD TEST SET.

Model	chair	desk	table
Baseline	73.3	27.1	50.0
IoU	74.8	28.2	51.4
CE	75.7	28.5	51.8
Set-based	79.4	35.4	54.2

compare the set-based detector, CE loss function, and IoU loss function with and without NMS. Without NMS, the set-based detector drops in performance by only 0.01% while CE and IoU drop by 3.79% and 4.75% respectively. It shows that our set-based detector still works well even without NMS. Moreover, we combine PointRCNN [33] and EPNet [12] with the set-based detector. It indicates that the set-based detector is beneficial for generating more high-quality proposals even without the NMS post-processing. As shown in Fig. 8, our set-based detector can filter out false positives, avoid missing distant objects, and even improve the predicted results.

Following the protocol of EPNet [12], we adopt the ratio of $\mathcal{R}$ to figure out how the consistency between these two confidences is improved, which is formulated as:

$$\mathcal{R} = \frac{\mathcal{N}(b|b \in \mathcal{B} \, and \, c_b > v)}{\mathcal{N}(\mathcal{B})} \quad (13)$$

where $\mathcal{B}$ denotes the set of positive candidate boxes, which are filtered by a predefined IoU threshold τ. And following [12], we set τ to 0.7, c_b represents the classification confidence of the positive candidate box b, and v is another threshold to filter positive candidate boxes with smaller classification confidence. $\mathcal{N}(.)$ calculates the number of boxes. In all different settings of classification confidence threshold v, all models generate 64 boxes without the NMS procedure employed. These boxes are used to get the positive candidate boxes by calculating the overlaps with the ground truth boxes. As shown in Fig. 7, the model with the set-based detector demonstrates better consistency than that trained with IoU loss and CE loss functions. Further, we evaluate different models on the SUN-RGBD test set. Especially, we select three categories "chair", "desk" and "table" in crowded scenes. As shown in Table V, the proposed set-based detector still outperforms other models in these categories. Such large gains demonstrate that our set-based detector performs better post-processing, keeping more highly-overlapped true positives, especially in the crowded scene.

C. Limitations

Fig. 9 shows the inference time among several compared methods. Our two-stage framework takes around 70ms to process a single point cloud sample from the KITTI dataset on an RTX 3090 GPU. Our model needs more time than one-stage frameworks, but it surpasses all one-stage detectors by a great margin. Meanwhile, our model is significantly faster than all two-stage detectors. This is because our model bypasses the consuming calculation of the NMS post-processing procedure. Besides, our model still obtains much higher mAP compared with two-stage detectors.

VI. Conclusion

In this paper, we present ImLiDAR, a novel 3D object detection paradigm, which progressively fuses camera images and LiDAR point clouds in multiple scales for quality 3D object detection. Two core designs exist in ImLiDAR. First, we present a cross-sensor dynamic message propagation module to combine the best of image and point features. Second, we

Fig. 9. Comparisons on inference time (ms) on the KITTI dataset. Our method reaches top performance among both one-stage and two-stage detectors.

design a set-based detector to select high-quality bounding boxes with both high classification and localization. It can be easily implemented in any detection network. Moreover, ImLiDAR does not require additional image annotations, the complex BEV data, and the commonly used NMS post-processing step. Extensive experiments on KITTI and SUN-RGBD datasets verify the superiority of ImLiDAR.

References

[1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2147–2156.

[2] L. Wang, L. Du, X. Ye, Y. Fu, G. Guo, X. Xue, J. Feng, and L. Zhang, "Depth-conditioned dynamic message propagation for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 454–463.

[3] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2040–2049.

[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," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2017.

[5] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7644–7652.

[6] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," *Advances in neural information processing systems*, vol. 28, 2015.

[7] 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 *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.

[8] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[9] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[10] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[11] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353.

[12] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 35–52.

[13] X. Zhao, Z. Liu, R. Hu, and K. Huang, "3d object detection using scale invariant and feature reweighting networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9267–9274.

[14] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 244–253.

[15] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 641–656.

[16] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

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

[18] X. Tan, X. Chen, G. Zhang, J. Ding, and X. Lan, "Mbdf-net: Multi-branch deep fusion network for 3d object detection," in *Proceedings of the 1st International Workshop on Multimedia Computing for Urban Data*, 2021, pp. 9–17.

[19] K. Huang and Q. Hao, "Joint multi-object detection and tracking with camera-lidar fusion for autonomous driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6983–6989.

[20] A. Piergiovanni, V. Casser, M. S. Ryoo, and A. Angelova, "4d-net for learned multi-modal alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 435–15 445.

[21] Z. Liu, B. Li, X. Chen, X. Wang, X. Bai *et al.*, "Epnet++: Cascade bi-directional fusion for multi-modal 3d object detection," *arXiv preprint arXiv:2112.11088*, 2021.

[22] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 720–736.

[23] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 867–11 876.

[24] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1057–1066.

[25] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8555–8564.

[26] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[27] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "Gs3d: An efficient 3d object detection framework for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1019–1028.

[28] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition workshops*, 2020, pp. 1000–1001.

[30] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 536–12 545.

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

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[32] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," *arXiv preprint arXiv:2012.15712*, vol. 1, no. 2, p. 4, 2020.

[33] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[34] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[35] Y. Zhang, D. Huang, and Y. Wang, "Pc-rgnn: Point cloud completion and graph neural network for 3d object detection," *arXiv preprint arXiv:2012.10412*, 2020.

[36] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7463–7472.

[37] H. Wu, J. Deng, C. Wen, X. Li, C. Wang, and J. Li, "Casa: A cascade attention network for 3-d object detection from lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[38] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 808–816.

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

[40] C. Yu, J. Lei, B. Peng, H. Shen, and Q. Huang, "Siev-net: A structure-information enhanced voxel network for 3d object detection from lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[41] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[42] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[43] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[44] Z. Li, F. Wang, and N. Wang, "Lidar r-cnn: An efficient and universal 3d object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7546–7555.

[45] L. Xie, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai, and X. He, "Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 460–12 467.

[46] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3194–3200.

[47] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[48] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4558–4567.

[49] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5199–5208.

[50] F. Ma, F. Zhang, Q. Yin, D. Xiang, and Y. Zhou, "Fast sar image segmentation with deep task-specific superpixel sampling and soft graph convolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[51] J. Zarzar, S. Giancola, and B. Ghanem, "Pointrgcn: Graph convolution networks for 3d vehicles detection refinement," *arXiv preprint arXiv:1911.12236*, 2019.

[52] W. Shi and R. 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 (CVPR)*, June 2020.

[53] S. Tian, L. Kang, X. Xing, J. Tian, C. Fan, and Y. Zhang, "A relation-augmented embedded graph attention network for remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.

[54] L. Zhang, D. Xu, A. Arnab, and P. H. Torr, "Dynamic graph message passing networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3726–3735.

[55] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[56] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[57] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

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

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

[60] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 040–11 048.

[61] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 873–11 882.

[62] J. Li, H. Dai, L. Shao, and Y. Ding, "Anchor-free 3d single stage detector with mask-guided attention for point cloud," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 553–562.

[63] J. Noh, S. Lee, and B. 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*, 2021, pp. 14 605–14 614.

[64] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "Cia-ssd: Confident iou-aware single-stage object detector from point cloud," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 4, 2021, pp. 3555–3562.

[65] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, "Improving 3d object detection with channel-wise transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2743–2752.

[66] C. Chen, Z. Chen, J. Zhang, and D. Tao, "Sasa: Semantics-augmented set abstraction for point-based 3d object detection," in *AAAI Conference on Artificial Intelligence*, vol. 1, 2022.

[67] Q. He, Z. Wang, H. Zeng, Y. Zeng, S. Liu, and B. Zeng, "Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds," *arXiv preprint arXiv:2006.04043*, 2020.

[68] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[69] S. Pang, D. Morris, and H. Radha, "Fast-clocs: Fast camera-lidar object candidates fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 187–196.

[70] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5428–5437.

[71] Y. Zhang, J. Chen, and D. Huang, "Cat-det: Contrastively augmented transformer for multi-modal 3d object detection," 2022.

[72] Q. Xie, Y.-K. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, and J. Wang, "Mlcvnet: Multi-level context votenet for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 447–10 456.

[73] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3dnet: 3d object detection using hybrid geometric primitives," in *European Conference on Computer Vision*. Springer, 2020, pp. 311–329.

[74] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A hierarchical graph network for 3d object detection on point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 392–401.

[75] Q. Xie, Y.-K. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, and J. Wang, "Vote-based 3d object detection with context modeling and sob-3dnms," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1857–1874, 2021.

[76] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2949–2958.

[77] J. Lahoud and B. Ghanem, "2d-driven 3d object detection in rgb-d images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4622–4630.

[78] Z. Ren and E. B. Sudderth, "Three-dimensional object detection and layout prediction using clouds of oriented gradients," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1525–1533.

[79] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[80] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.