

BSH-Det3D: Improving 3D Object Detection with BEV Shape Heatmap

You Shen¹, Yunzhou Zhang^{1*}, Yanmin Wu², Zhenyu Wang³, Linghao Yang¹,
Sonya Coleman⁴, Dermot Kerr⁴

Abstract—The progress of LiDAR-based 3D object detection has significantly enhanced developments in autonomous driving and robotics. However, due to the limitations of LiDAR sensors, object shapes suffer from deterioration in occluded and distant areas, which creates a fundamental challenge to 3D perception. Existing methods estimate specific 3D shapes and achieve remarkable performance. However, these methods rely on extensive computation and memory, causing imbalances between accuracy and real-time performance. To tackle this challenge, we propose a novel LiDAR-based 3D object detection model named BSH-Det3D, which applies an effective way to enhance spatial features by estimating complete shapes from a bird’s eye view (BEV). Specifically, we design the Pillar-based Shape Completion (PSC) module to predict the probability of occupancy whether a pillar contains object shapes. The PSC module generates a BEV shape heatmap for each scene. After integrating with heatmaps, BSH-Det3D can provide additional information in shape deterioration areas and generate high-quality 3D proposals. We also design an attention-based densification fusion module (ADF) to adaptively associate the sparse features with heatmaps and raw points. The ADF module integrates the advantages of points and shapes knowledge with negligible overheads. Extensive experiments on the KITTI benchmark achieve state-of-the-art (SOTA) performance in terms of accuracy and speed, demonstrating the efficiency and flexibility of BSH-Det3D. The source code is available on <https://github.com/mystorm16/BSH-Det3D>.

I. INTRODUCTION

Over the past decade, deep learning has made significant progress in 2D vision tasks such as detection [1]–[3], segmentation [4]–[6], and pose estimation [7]. While 2D images have valuable information, 3D point clouds can provide more geometric, shape and scale information [41], significantly improving scene perception capability. LiDAR sensors have been widely used to obtain 3D point clouds in autonomous driving, mobile robotics, and augmented reality/virtual reality thanks to high-precision measurements and robustness to illumination changes. Although LiDAR sensors have these advantages, achieving high-performance detection in point clouds is still challenging due to two inherent limitations:

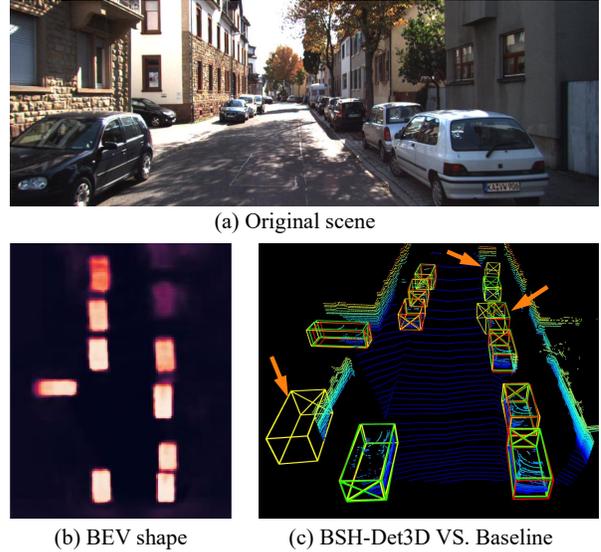


Fig. 1. Performance comparisons of our BSH-Det3D with the baseline [16] on the KITTI *val* set. (a) RGB image of the original scene. (b) Result of BEV shape heatmap. (c) Result of our BSH-Det3D and the baseline detector. The ground-truth boxes, predicted boxes of the baseline and BSH-Det3D are shown in red, yellow, and green. As indicated by the orange arrows, our method can fix box offset and removes false positives effectively.

- Laser beams return after hitting the first object, causing shapes behind the occluder to be missing.
- The faraway objects receive only a few points on their surfaces, so shapes at far-range areas will be sparse and incomplete.

Both limitations cause shape deterioration during detection. To address this issue, Xu *et al.* [9] complete entire shapes by manual ground truth. Subsequently, almost all objects are detected correctly (Average Precision > 99%), proving that complete shapes are essential for high-performance detectors. One straightforward way to alleviate the missing shape problem is learning object shapes from priors: SPG [10] and BtcDet [9] manually fill points into the labeled bounding boxes and try to recover the specific shape of objects. However, since they focus on entire 3D shapes, these methods are computationally expensive. Other methods convert shape information to features instead of specific shapes [13]–[15] to reduce computation. However, this is challenging due to shape feature extraction and fusion resulting lower accuracy. Thus, a new challenge arises: **How do detectors alleviate shape deterioration while remaining efficient and flexible?**

To tackle this challenge, we chose the BEV-based method for its speed and accuracy performance, *e.g.*, PointPillars [16]. These methods encode points into pseudo-images and

*The corresponding author of this paper.

¹You Shen, Yunzhou Zhang, Linghao Yang are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (Email: zhangyunzhou@mail.neu.edu.cn).

²Yanmin Wu is with School of Electronic and Computer Engineering, Peking University, Shenzhen, China.

³Zhenyu Wang is with Faculty of Robotics and Engineering of Northeastern University, Shenyang, China, and the Department of Electronic and Computer Engineering of Technical University in Munich, Germany.

⁴Sonya Coleman and Dermot Kerr are with School of Computing, Engineering and Intelligent Systems, Ulster University, N. Ireland, UK.

This work was supported by National Natural Science Foundation of China (No. 61973066), Major Science and Technology Projects of Liaoning Province (No. 2021JH1/10400049), Fundamental Research Funds for the Central Universities (N2004022).

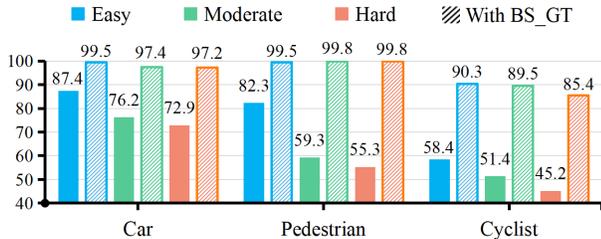


Fig. 2. This shows the performance of vanilla PointPillars [16] (solid bar) and PointPillars with ground truth BEV shape heatmap (striped bar) on the KITTI [12] val split. We show the result in three classes: Car, Pedestrian, and Cyclist; with three difficulty levels: Easy, Moderate, and Hard. After associating with the BEV shape, the performance significantly improves.

use well-established 2D convolutions to extract features. We conduct a pilot study to explore the possibility of detectors with BEV shapes. We design ground truth BEV shapes as heatmaps (see details in III-B). Then, we directly concatenate heatmaps with raw point features extracted by the baseline detector [16]. Both training and evaluation is based on the KITTI [12] dataset. We show the average precision (AP) of cars, pedestrians, and cyclists with three occlusion levels. As illustrated in Fig.2, the performance improves significantly with the fusion of BEV shape heatmaps, demonstrating the potential of utilizing BEV shapes to enhance detection.

In this paper, we present a novel approach to improving 3D object detectors with BEV Shape Heatmap (BSH-Det3D), which alleviates shape deterioration efficiently, as illustrated in Fig.1. BSH-Det3D proposes an effective method for improving detection quality by learning BEV shape knowledge. We design a pillar-based shape completion module (PSC) to obtain a BEV shape heatmap in each scene. PSC extracts multi-scale pillar features and estimates the occupancy probability of whether a pillar belongs to complete shapes. Furthermore, targeting the sparsity of both point cloud features and shape heatmaps, we propose an attention-based densification fusion module (ADF) to associate points and shapes. We conduct experiments on the KITTI dataset, and achieve SOTA performance in terms of accuracy and speed. It is worth noting that our method is general and can be used with various detectors. To validate the flexibility, we conduct experiments integrating our method into different mainstream 3D detection frameworks [16], [21] and design a two-stage module for box refinement.

The main contributions can be summarized as follows:

- We propose a novel 3D object detector that learns to associate object shape knowledge from a bird’s eye view, enhancing spatial features and providing implicit guidance for detection.
- We design a pillar-based shape completion module (PSC) to estimate a probability-based shape heatmap for each scene, alleviating shape deterioration efficiently.
- We design an attention-based densification fusion module (ADF) that adapts to the sparse features of shapes and point clouds with negligible overheads.
- Our proposed BSH-Det3D achieves SOTA performances on the KITTI benchmark in terms of accuracy and real-time performance. We also test our approach with different baseline detectors to verify its flexibility.

II. RELATED WORK

A. LiDAR-based 3D Object Detectors

According to the representation of point clouds, LiDAR-based 3D object detection can be divided into point-based and grid-based methods. Point-based methods inherit the success of feature extraction modules [22] and propose diverse architectures to detect objects from raw points directly. PointRCNN [20] segments point clouds by PointNet++ [23] and estimates proposals for each foreground point. STD [25] presents a point-based proposal generation paradigm with spherical anchors to reduce computation. 3DSSD [24] combines feature-based and point-based sampling to improve the classification. Grid-based 3D detectors first transfer raw points into discrete grid representations such as voxels and pillars; then, detectors use 2D or 3D convolutional neural networks to extract features from grids and detect objects from grid cells. VoxelNet [42] divides point cloud into 3D voxels, which are further processed by the voxel feature extractor and 3D CNN encoder network. SECOND [21] introduces sparse 3D convolutions for efficient 3D processing of voxels, significantly improving real-time performance. PointPillars [16] collapses raw points into vertical pillars, uses a per-pillar feature extractor by PointNet [22] to compress the height dimension, and then utilizes the pillar feature as a BEV pseudo-image for detection. Unlike existing BEV detectors which encode points into pseudo-image and directly estimate proposals, we propose a novel way of predicting shape heatmaps in BEV to enhance spatial features. Our BSH-Det3D is designed on grid-based methods and can be easily integrated into various detectors.

B. Shape Priors for 3D Object Detection

Many advanced 3D detectors focus on alleviating missing shape problems by using shape priors. One class of methods attempts to learn shape knowledge as features: Part-A² [13] applies a part-aware stage to obtain object part locations. SA-SSD [14] and Associate-3Ddet [15] aim to exploit structure modules to conserve shape features by auxiliary networks. However, due to difficulties in extracting and fusing shape features, these methods suffer from accuracy bottlenecks. Another class of methods recovers the specific shape of objects: SPG [10] locates foreground regions and generates semantic points for each foreground voxel. BtcDet [9] finds similar shapes as priors and generates new shape points. Due to the estimate of entire 3D shapes, these methods consume significant computation and memory, resulting in imbalances between accuracy and real-time performance. Given this, BSH-Det3D proposes a novel mechanism for utilizing shape priors as BEV heatmaps, which effectively reduces shape deterioration with a small amount of extra time, particularly for occluded objects or distant objects.

III. METHODOLOGY

A. Model Overview

As illustrated in Fig.3, the PSC module first pillarizes raw points and utilizes a pillar-wise occupancy network to

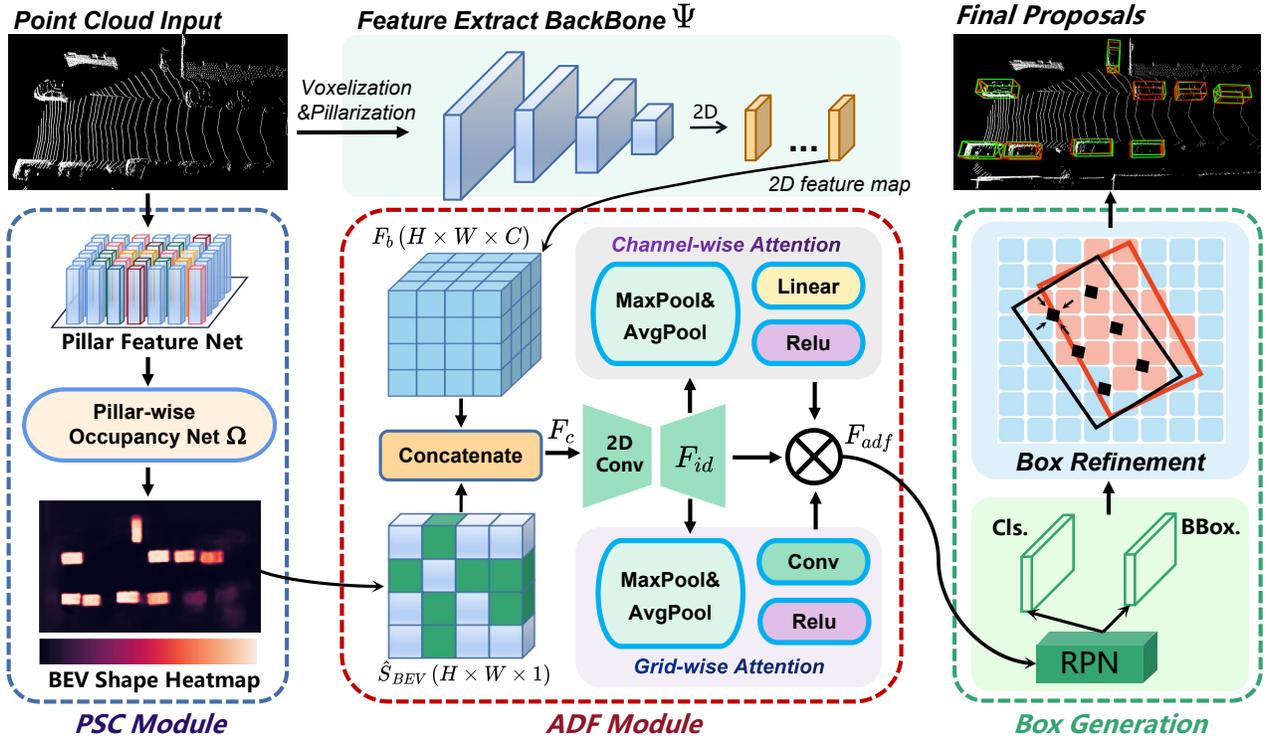


Fig. 3. **Detection pipeline:** The PSC module first splits points into pillars and estimates pillar-wise shape occupancy probability to generate BEV shape heatmap \hat{S}_{BEV} . The backbone network Ψ extracts feature from raw points, and \hat{S}_{BEV} concatenates with the output feature of Ψ . Then, the ADF module uses a hybrid-attention based strategy to fuse features, linking with an RPN network to generate 3D proposals. Furthermore, for each proposal, BSH-Det3D constructs local grids and pools the local features with \hat{S}_{BEV} to the nearby grids for further refinement (see black box and red box in Box Refinement).

estimate the BEV shape heatmap \hat{S}_{BEV} (III-B). Next, BSH-Det3D uses a backbone network Ψ [21] [16] to extract features F_b of point clouds. To associate the shape knowledge, the ADF module concatenates the \hat{S}_{BEV} to the output feature from Ψ and gets F_c ; we then use a hybrid-attention based strategy to get fusion features F_{adf} . Finally, F_{adf} is sent to a Region Proposal Network (RPN) generating 3D proposals. During box refinement, we construct local grids covering each proposal box, and aggregate the grid features with shape heatmaps to generate the final proposals.

B. Pillar-based Shape Completion Module

Generation of ground truth labels. The process of generating BEV shape labels is illustrated in Fig.4. First, we follow methods in [9] using a heuristic-based strategy to find the top three shapes which are most similar to current shapes S_d in the dataset. After assembling similar shapes, the approximate 3D shapes S_c can be obtained. Next, we compress the height dimension of S_c and get the corresponding 2D shapes S_{2d} . In S_{2d} , we set shape occupancy probability $P(O_s) = 1$ for pillars that contain shapes and $P(O_s) = 0$ for the others. When multiple objects are assembled as S_c , the point density varies, which causes voids in S_{2d} . To counteract this, we increase the positive supervision for the S_{2d} by span locations p_{xy} where $P(O_s) = 1$ with a Gaussian kernel:

$$Y_{xy} = \exp\left(-\frac{(x - p_x)^2 + (y - p_y)^2}{2\sigma_p^2}\right), \quad (1)$$

where σ_p is an object size-adaptive standard deviation depending on the size of the object. Finally, S_g is used as the ground truth label of the BEV shape heatmap.

Estimation of BEV shape heatmap. First, we adopt the strategy of [16] to extract pillar features: PSC module projects raw points on the X-Y plane via a tiny one-layer PointNet [22] to fetch pillar feature f_p . Then, f_p is processed by a pillar-wise shape occupancy network Ω (Fig.5). Ω adopts a top-down architecture that processes the f_p with stride $1\times$, $2\times$, and $4\times$ convolution blocks, each block linking to a transposed 2D convolution for upsampling and then concatenating the multi-scale features for the detection head. The detection head uses 3×3 convolutional layers separated by ReLU and BatchNorm. The last convolutional layer produces a K-channel heatmap \hat{Y} showing the shape occupancy probability, where the channel of \hat{Y} indicates object classes. To highlight the shapes, we use a sigmoid function with the threshold (≥ 0.5) to filter \hat{Y} . Finally, we get the estimated BEV shape heatmap \hat{S}_{BEV} .

Sigmoid cross-entropy Focal Loss [26] supervises the output of Ω , if $Y_{xy} = 1$:

$$L_{shape} = -\frac{1}{N} \sum_{xy} (1 - \hat{Y}_{xy})^\alpha \log(\hat{Y}_{xy}), \quad (2)$$

otherwise:

$$L_{shape} = -\frac{1}{N} \sum_{xy} (1 - Y_{xy})^\beta (\hat{Y}_{xy})^\alpha \log(1 - \hat{Y}_{xy}), \quad (3)$$

where $\alpha = 2$ and $\beta = 4$ are default hyper-parameters and N is the number of p where $P(O_s) = 1$. Since PSC module involves only fast 2D convolutions and MLPs, this guarantees detection efficiency.

C. Attention-based Densification Fusion Module

Some parts of object shapes significantly determine the performance and deserve more attention. Inspired by [27] we

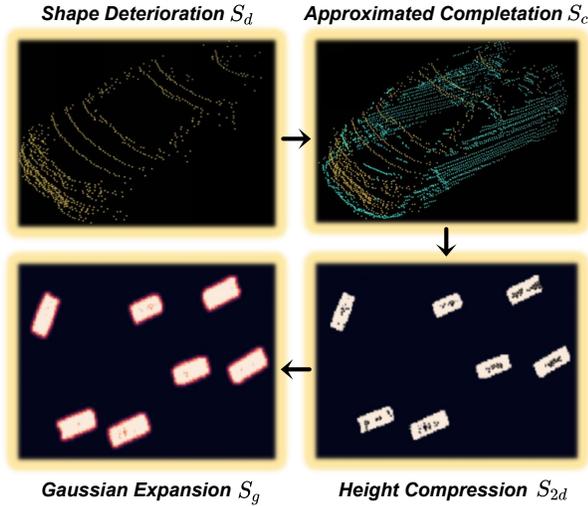


Fig. 4. The generation process of our BEV shape label. For each shape deteriorated object S_d , we obtain completed 3D shapes by following the method in [9]. Next, we compress S_c into 2D by height dimension, getting S_{2d} . After that, we apply rendered Gaussian kernels to counteract the noise introduced by the shape priors, getting high-quality BEV shape labels S_g .

exploit an effective hybrid-attention-based ADF module for adaptive feature refinement. The ADF module extracts shape knowledge from \hat{S}_{BEV} to enhance the point cloud features F_b from the backbone network. As shown in Fig.3, targeting the sparsity of both \hat{S}_{BEV} and F_b , the concatenation feature F_c first densifies by 2D convolutional layers making the initial dense fusion feature F_{id} available. The ADF module consists of channel-wise attention and grid-wise attention.

Channel-wise attention focuses on filtering significant semantics in F_{id} . We first aggregate spatial information in two descriptors by using average-pooling and max-pooling, then both descriptors are forwarded to a shared network producing a channel map. Afterwards, we merge the feature maps by summation, followed by a sigmoid function σ . In short, the computation of channel-wise attention map is:

$$M_c(F_{id}) = \sigma(MLP(F_{id}^{Avg}) + MLP(F_{id}^{Max})). \quad (4)$$

Grid-wise attention focuses on filtering significant positions in F_{id} . We apply average-pooling and max-pooling to compress the channel dimension of F_{id} , connected with a convolution layer to encode the part of shape requiring more attention. Finally, we generate a grid attention map by utilizing the inter-spatial relationship of F_{id} . The computation of grid-wise attention map is:

$$M_g(F_{id}) = \sigma(f^{7 \times 7}([F_{id}^{Avg}'; F_{id}^{Max}'])), \quad (5)$$

where σ denotes the sigmoid function and $f^{7 \times 7}$ represents a 7×7 convolutional layer.

These two attention modules focus on semantics and positions respectively. Using \otimes to denote element-wise multiplication, the overall attention process can be computed as:

$$F_{adf} = M_g(F_{id}) \otimes M_c(F_{id}) \otimes F_{id}. \quad (6)$$

D. One-Stage and Two-Stage BSH-Det3D

Choosing to focus on accuracy or speed, the backbone network adopts two ways to encode raw points. For speed,

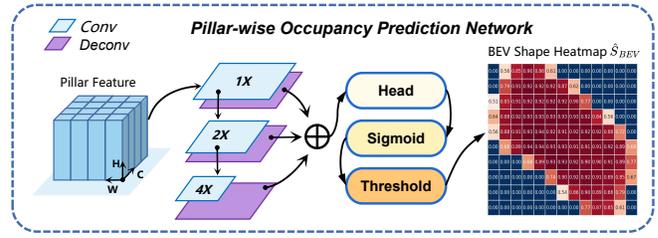


Fig. 5. The estimation of the BEV shape heatmap. We adopt a top-down architecture to extract multi-scale pillar features. The detection head predicts the shape occupancy probability as a heatmap. After sending it to the sigmoid function and threshold, we can get the final heatmap \hat{S}_{BEV} .

we encode points into pillar feature maps [16]. For accuracy, we encode points into voxels [42], and then the 3D backbone extracts features by sparse convolution and compresses along the height axis to generate 2D feature maps. After encoding, the 2D backbone uses a multi-scale feature fusion network to generate a 2D feature map.

One-Stage. The Region Proposal Network (RPN) takes the output features of the ADF module and predicts stage-one proposals with the anchor-based approaches [16], [21]. Two anchors of 0° , 90° are evaluated for each pixel of the BEV feature map. Each proposal contains eight parameters: center coordinates (x_p, y_p, z_p) , box size (l_p, w_p, h_p) , yaw rotation angle θ_p , and classification confidence c_p .

Two-Stage. Based on proposals and grid features learned from stage-one, the refinement module further exploits the BEV shape heatmaps, as shown in Fig.3. Each box needs to consider the nearby geometry structure to generate accurate final proposals. We design a RoI-grid pooling strategy inspired by [18]. For each proposal, we construct local grids of $6 \times 6 \times 6$ to capture contextual information among the neighboring voxel features. To further enhance the awareness of shape deterioration, we pool the BEV shape heatmap \hat{S}_{BEV} onto the nearby grids through bilinear-interpolation and aggregate them by fusing multiple levels of F_{adf} . After that, two branches of MLP are used to predict an IoU-related class confidence score and residuals between the 3D proposal and ground truth bounding box for two-stage results.

E. Loss Function

The proposed BSH-Det3D is trained in an end-to-end manner. Our overall loss includes L_{rpn} in stage-one, box refinement loss L_{rcnn} in stage-two, and BEV shape heatmap estimation loss L_{shape} in Eq.2 as:

$$L_{total} = \lambda L_{shape} + L_{rpn} + L_{pr}. \quad (7)$$

Following [16], [21], L_{rpn} is defined as the summation of classification loss and box regression loss as:

$$L_{rpn} = L_{cls} + \sum_r L_{smooth-L1}(\Delta r_1), \quad (8)$$

$$r_1 \in \{x, y, z, l, h, w, \theta\}$$

where r_1 is the parametric representation of the proposal, and the smooth-L1 loss is used to anchor box regression with the predicted residual and the regression target. We use focal loss [26] to calculate the anchor classification:

$$L_{cls} = \alpha (1 - p_t)^\gamma \log(p_t), \quad (9)$$

TABLE I
PERFORMANCE COMPARISON OF OBJECT DETECTION WITH SOTA LIDAR METHODS OF KITTI *val* SPLIT.

Stage	Method	Car 3D AP_{R40}			Car BEV AP_{R40}			Cyc. 3D AP_{R40}			Time (ms)
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
One-stage	PointPillars [16]	87.75	78.39	75.18	92.40	87.79	86.39	81.57	62.94	58.98	24
	BSH-Det3D(Pillars)	89.08	81.66	79.01	92.80	89.15	88.46	86.48	69.22	63.59	32
	Improvement	+1.33	+3.27	+3.83	+0.40	+1.36	+2.07	+4.91	+6.28	+4.61	-8
	SECOND [21]	90.97	79.94	77.09	95.61	89.54	86.96	78.50	56.74	52.83	50
	BSH-Det3D(Voxels)	91.07	82.53	79.54	93.04	89.28	88.43	85.32	66.23	64.92	43
	Improvement	+0.10	+2.59	+2.45	-2.57	-0.26	+3.27	+6.82	+9.49	+12.09	+7
Two-stage	PV-RCNN [19]	92.10	84.36	82.48	93.02	90.33	88.53	88.88	71.95	66.78	80
	Voxel R-CNN [18]	92.38	85.29	82.86	95.52	91.25	88.99	-	-	-	40
	BSH-Det3D(Refinement)	92.88	85.93	83.49	93.97	91.94	89.60	89.64	72.33	69.05	48
	Improvement	+0.50	+0.64	+0.63	-1.55	+0.69	+0.61	+0.76	+0.38	+2.27	-8

where p_t is the class probability of an anchor and we use the default hyper-parameters $\alpha = 0.25$ and $\gamma = 2$.

The proposal refinement loss L_{rcnn} includes the IoU-guided confidence prediction loss [19] and box refinement loss as:

$$L_{rcnn} = L_{iou} + \sum_{r_2} L_{smooth-L1}(\Delta r_2), \quad (10)$$

where Δr_2 is the residual between the predicted box and proposal target which are encoded similarly to Δr_1 .

IV. EXPERIMENTS

A. Dataset and Evaluation

We train and evaluate our proposed BSH-Det3D on the widely acknowledged KITTI dataset [12], which offers 7,481 samples for training and 7,518 samples for testing. Referring to previous work such as [19], [21], we split training examples into the train set (3,712 samples) and the *val* set (3,769 samples). The samples are classified as three difficulty levels: easy, moderate, and hard, the official KITTI leaderboard is ranked on the moderate levels. We adopt AP with a 3D overlap threshold of 0.7 as the evaluation metric of the Car class and 0.5 for Cyclist.

B. Implementation Details

Pillarization&Voxelization. Before sending to networks, the raw points are first encoded into pillars or voxels. For voxelization, we clip the range of point clouds into $[0, 70.4]m$ for the X-axis, $[-40, 40]m$ for the Y-axis, and $[-3, 1]m$ for the Z-axis. The input voxel size is set as $(0.05m, 0.05m, 0.1m)$. For pillarization, we define the detection range as $[0, 69.12]m$ for the X-axis, $[-39.68, 39.68]m$ for the Y-axis, and $[-3, 1]m$ for the Z-axis. We set the pillar size to $(0.16m, 0.16m, 4m)$.

Training. The BSH-Det3D is end-to-end optimized by the ADAM optimizer [28] from scratch. The parameter λ in Eq.7 is set to 6.0 empirically. We train our models with a batch size of 8 on a GTX 3090 GPU for 80 epochs. The learning rate is initialized as 0.01 and updated by the cosine annealing strategy. We randomly sample 128 proposals for training, and 50% of them are positive samples that have $\text{IoU} > 0.55$ with the corresponding ground truth boxes.

Inference. During the inference stage, non-maximum suppression (NMS) is conducted with a threshold > 0.7 to filter

the redundant proposals. We choose the top 100 proposals for refinement. After refinement, NMS is applied with IoU threshold 0.01 to remove redundant box predictions.

Data augmentation. First, we randomly sample objects from the training data and inject them into the training samples as [21]. Next, we randomly flip scenes along X-axis with 50% probability. Then, we rotate each scene around Z-axis with a random angle sampled from $[-45^\circ, 45^\circ]$. Finally, we uniformly sample a scaling factor from the range of $[0.95, 1.05]$ and use it to scale the point cloud.

C. Evaluation Results

We evaluate BSH-Det3D for 3D detection and BEV detection benchmark on KITTI *val* split and test split. Corresponding to the KITTI [12] protocol, we calculate the AP results under 40 recall thresholds (R40). The evaluation results show three major advantages of our method.

High performance. We compare our two-stage detector BSH-Det3D(Refinement) with the front runners on the KITTI leaderboard by submitting our results to the online test server, as illustrated in Table II. Our BSH-Det3D can effectively improve detection performance and achieves balance between accuracy and efficiency. By taking full advantage of BEV shape knowledge, BSH-Det3D(Refinement) achieves 81.91% average precision on the moderate level of class Car with 48ms. Our method outperforms many multi-modality fusion-based methods, including UberATG-MMF [33], 3D-CVF [34], and CLOCs PVCas [35], by a large margin (1.24% to 4.48% of moderate AP and 0.21% to 7.14% of hard AP). Compared with the LiDAR-based methods, we also outperform the recent SOTA detectors, *e.g.*, PV-RCNN [19], Voxel R-CNN [18], and CIA-SSD [38] by 0.29% to 1.63% of moderate AP and 0.30% to 4.49% of hard AP. The KITTI test set results demonstrate that our proposed BSH-Det3D achieves the SOTA performance on 3D object detection and keeps the high efficiency to address shape deterioration.

Flexibility. As summarized in Table I, to demonstrate that BSH-Det3D can generalize across models, we build BSH-Det3D using two mainstream detectors [16], [21] implemented in OpenPCDet [40]. The box refinement module is designed based on the SECOND detector [21]. We evaluate BSH-Det3D on Car and Cyclist categories and use the AP_{R40} for PV-RCNN [19] and Voxel R-CNN [18] from their

TABLE II
PERFORMANCE COMPARISON OF 3D AND BEV DETECTION OF CAR CLASS ON KITTI TEST SPLIT.

Modality	Method	Stage	3D Detection AP_{R40}			BEV Detection AP_{R40}			Time (ms)
			Easy	Mod.	Hard	Easy	Mod.	Hard	
RGB+LiDAR	MV3D [29]	Two	74.97	63.63	54.00	86.62	78.93	69.80	360
	AVOD [31]	Two	83.07	71.76	65.73	89.75	84.95	78.32	100
	ContFuse [32]	One	83.68	68.78	61.67	94.07	85.35	75.88	60
	UberATG-MMF [33]	Two	88.40	77.43	70.22	93.67	88.21	81.99	80
	3D-CVF [34]	Two	89.20	80.05	73.11	93.52	89.56	82.45	75
	CLOCs PVCas [35]	Two	88.94	80.67	77.15	93.05	89.80	86.57	100
LiDAR only	SECOND [21]	One	83.34	72.55	65.82	89.39	83.77	78.59	50
	PointPillars [16]	One	82.58	74.34	68.99	90.07	86.56	82.81	24
	PointRCNN [20]	Two	86.96	75.64	70.70	92.13	87.39	82.72	100
	STD [25]	Two	87.95	79.71	75.09	94.74	89.19	86.42	80
	Part-A2 [13]	Two	87.81	78.49	73.51	91.70	87.79	84.61	80
	Associate-3Det [15]	One	85.99	77.40	70.53	91.40	88.09	82.96	60
	3DSSD [24]	One	88.36	79.57	74.55	92.66	89.02	85.86	38
	PV-RCNN [19]	Two	90.25	81.43	76.82	94.98	90.65	86.14	80
	Point-GNN [37]	One	88.33	79.47	72.29	93.11	89.17	83.90	643
	TANet [39]	One	84.39	75.94	68.82	91.58	86.54	81.19	35
	Voxel R-CNN [18]	Two	90.90	81.62	77.06	94.85	88.83	86.13	40
	CIA-SSD [38]	One	89.59	80.28	72.87	93.74	89.84	82.39	31
	BSH-Det3D(ours)	Two	88.75	81.91	77.36	92.90	90.99	86.43	48

papers and the AP_{R40} for PointPillars [16] and SECOND [21] are generated from the officially released code. Compared to baseline methods, the BSH-Det3D detectors increase the performance in 3D and BEV object detection by a large margin. Results demonstrate the effectiveness and flexibility of our method.

Efficiency. Notably, owing to the fast 2D CNN architecture in PSC module and the effective fusion strategy in ADF module, our method is quite efficient (Table I). We measure the runtime of BSH-Det3D on an Intel i7-12700F and a single 3060Ti GPU. The moderate AP and running speed comparisons in the KITTI server are shown in Fig.6.

Comparing to other detectors focus on shape missing, our one-stage detector BSH-Det3D(Pillars) is the only detector that works in real-time (32ms). It also achieves 79.10% AP, significantly improving the baseline PointPillars [16] by 4.76% of moderate AP. Moreover, BSH-Det3D(Refinement) achieves comparable accuracy with the strong competitors, *i.e.*, BtcDet [9] and SPG [10], with only about 50% of the running time. This verifies that the BEV shapes are almost sufficient for shape deterioration in 3D object detection, and the BEV shape representation is more efficient than specific 3D shapes.

Qualitative results. Some visualization results for BSH-Det3D(Voxels) and the corresponding BEV shape heatmaps are illustrated in Fig.7. We show detection results on three typical KITTI scenes: City, Rural, and Highway. In comparison to the baseline [21] method, we mark the advantage of BSH-Det3D with orange arrows. In the city, objects shapes are obscured at corner intersections, which causes missed detection in the baseline. However, due to the guidance of the BEV shape, the missing shapes are completed with shape heatmaps, and all objects can get effective detection in BSH-Det3D. In rural areas, the vehicles on the left are parked

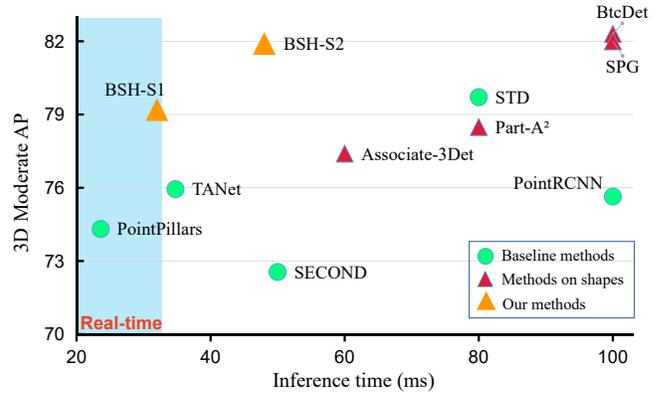


Fig. 6. The 3D detection performance and speed for our BSH-Det3D on the KITTI test set compared with SOTA detectors, especially those focusing on shapes missing. BSH-S1 and BSH-S2 show the result of BSH-Det3D(Pillars) and BSH-Det3D(Refinement), respectively.

compactly, and the baseline [21] suffers detection drift by limitation of points information. In comparison, by taking pillar-wise shape estimation, shapes can be separated under compact parking areas, and the drift box can be corrected. Regarding highways, the vehicles driving side by side can cause multiple occlusions, resulting in the baseline method facing critical false detection, especially in far-range areas. By associating shape heatmaps, our detector can provide more descriptions of objects, significantly alleviating false detections. In conclusion, BSH-Det3D has benefits in various scenes, effectively suppressing missing and false detections.

D. Ablation Studies

Effect of Components. Table III details how each proposed module influences the accuracy and efficiency of our BSH-Det3D. The results are evaluated with AP_{R40} of moderate level for the car class. *Method(a)* is the one-stage baseline that performs detection on BEV features which runs at 25.5ms. *Method(b)* extends (a) with a pillar-based

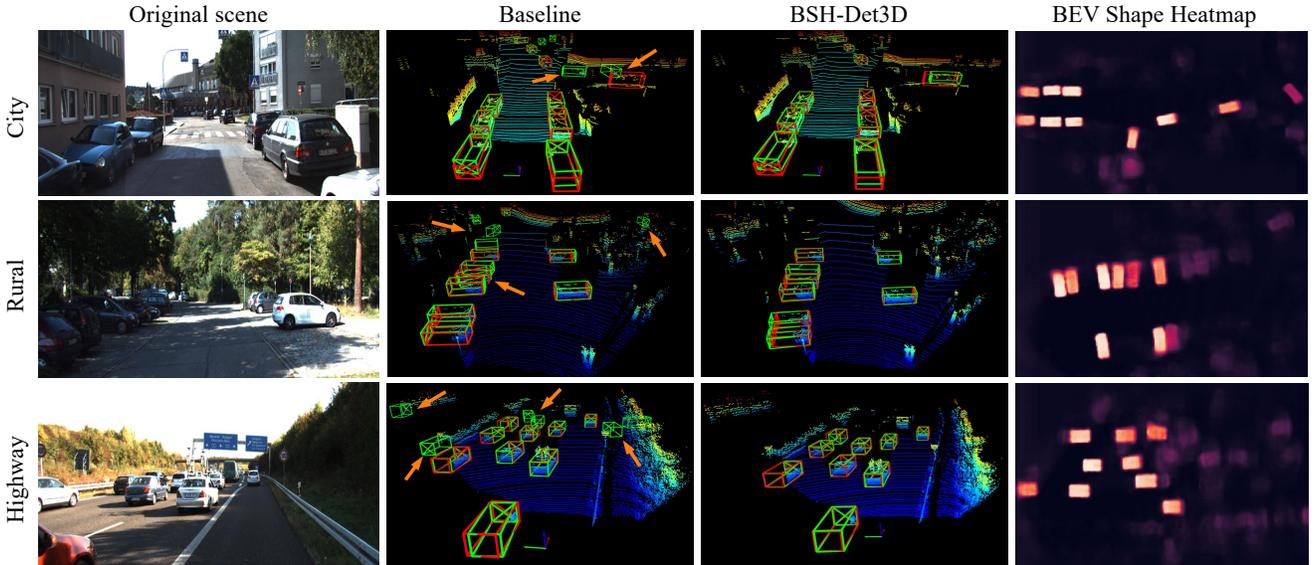


Fig. 7. Qualitative results of the KITTI *val* split. We select three typical scenes in KITTI: City, Rural, and Highway. We show the BSH-Det3D(Voxels) results with corresponding BEV shape heatmaps for each scene. The advantages compared with the baseline [21] are marked with the orange arrows. The boxes of prediction and ground truth are rendered in green and red. Results show that our method positively reduces false detections and corrects offset.

TABLE III

PERFORMANCE WITH DIFFERENT CONFIGURATIONS ON KITTI *val* SET.

Methods	PSC	GAU	ADF	REF	3D AP_{R40}	Time (ms)
(a)					79.94	25.5
(b)	✓				81.73	33.2
(c)	✓		✓		82.84	35.6
(d)	✓	✓	✓		83.17	35.6
(e)	✓	✓	✓	✓	85.93	48.2

shape completion module, which directly concatenates with raw points feature and fusion by 2D convolution. The PSC module leads to a boost of 1.79% moderate AP, which verifies that PSC can strengthen the robustness of spatial features. We apply pillarization and 2D CNN leading to a decrease of 33.2ms. *Method(c)* replaces feature fusion with our ADF module, which makes 1.11% AP improvement and only an extra 2.4ms thanks to a simple yet effective Hybrid-Attention feature fusion strategy. *Method(d)* uses our rendered Gaussian kernels (GAU) to counteract the noise of shape priors during training, boosting 0.33% AP without extra cost. *Method(e)* is the proposed BSH-Det3D(Refinement) by extending the one-stage detector with a box refinement module combined with BEV shape knowledge. BSH-Det3D(Refinement) achieves SOTA accuracy for 3D object detection and maintains efficiency.

Effect on different recall thresholds. We designed an experiment based on four intervals to further analyze how BSH-Det3D enhances the detectors. Since the recall positions of R40 are recorded according to the sorted predicted confidence, objects in $[R1, R30]$ generally have low detection difficulty and retain almost complete object shapes. In comparison, the objects of $[R31, R40]$ usually suffer from shape deterioration in occluded and distant areas. Thus, our experiment evenly slices the 40 recall points into four pieces, counts the number of true positives (TP) and false positives (FP), then calculates the TP's ratios (TP/(TP+FP))

TABLE IV

DETECTION OF DIFFERENT INTERVALS ON KITTI *val* SET.

Recall Interval	Method	Car 3D Detection		Ration (%)
		TP	FP	
R1-R10	PointPillars [16]	1772	23	98.72
	BSH-Det3D		17	99.04 (0.32↑)
R11-R20	PointPillars [16]	3740	157	95.97
	BSH-Det3D		124	96.80 (0.83↑)
R21-R30	PointPillars [16]	5709	767	88.16
	BSH-Det3D		600	90.49 (2.33↑)
R31-R40	PointPillars [16]	6496	3567	64.55
	BSH-Det3D		1836	77.96 (13.41↑)

in each interval. Our experiment is conducted on BSH-Det3D(Voxels) of the KITTI *val* set, moderate level on Car category. As demonstrated in Table IV, for $[R1, R30]$, the TP ratio of our methods has a slight improvement over the baseline. However, for $[R31, R40]$, the ratio of FP is greatly reduced in BSH-Det3D, which plays a significant role in detection performance. It can be inferred that with the associated BEV shape heatmap, BSH-Det3D can provide additional descriptions of objects, which is especially useful in correcting low-confidence boxes that suffer from shape deterioration.

V. CONCLUSION

In this paper, we demonstrate that shape deterioration is a fundamental challenge in 3D object detection. To tackle this, we present the novel BSH-Det3D exploiting the potential of BEV shapes to improve detection. Specifically, we design a efficient PSC module that learns to enhance spatial features by producing the complete BEV shapes in each scene. Additionally, we introduce a hybrid-attention based ADF module for adaptive feature refinement between shapes

and raw points with negligible overhead. It should also be noted that the BEV shape heatmaps of our approach can be easily integrated into many existing detectors in 3D point clouds. Experimental results on the KITTI benchmark dataset have validated the efficiency and flexibility of our BSH-Det3D. In future work, we plan to integrate the BEV shape completion module with RGB images to further improve the performance.

REFERENCES

- [1] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [6] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.
- [7] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, “Cascaded pyramid network for multi-person pose estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7103–7112.
- [8] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5693–5703.
- [9] Q. Xu, Y. Zhong, and U. Neumann, “Behind the curtain: Learning occluded shapes for 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2893–2901.
- [10] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov, “Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 446–15 456.
- [11] Z. Li, Y. Yao, Z. Quan, W. Yang, and J. Xie, “Sienet: Spatial information enhancement network for 3d object detection from point cloud,” *arXiv preprint arXiv:2103.15396*, 2021.
- [12] 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.
- [13] S. Shi, Z. Wang, J. Shi, X. Wang, and H. 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*, vol. 43, no. 8, pp. 2647–2664, 2020.
- [14] 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.
- [15] L. Du, X. Ye, X. Tan, J. Feng, Z. Xu, E. Ding, and S. 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*, 2020, pp. 13 329–13 338.
- [16] 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.
- [17] T. Yin, X. Zhou, and P. Krahenbuhl, “Center-based 3d object detection and tracking,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.
- [18] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, “Voxel r-cnn: Towards high performance voxel-based 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1201–1209.
- [19] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, “Pv-rnn: 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.
- [20] S. Shi, X. Wang, and H. Li, “Pointcnn: 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.
- [21] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, no. 10, p. 3337, 2018.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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 *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer, 2020, pp. 720–736.
- [35] S. Pang, D. Morris, and H. Radha, “Clocs: Camera-lidar object candidates fusion for 3d object detection,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 386–10 393.
- [36] Y. Chen, S. Liu, X. Shen, and J. Jia, “Fast point r-cnn,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9775–9784.
- [37] 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*, 2020, pp. 1711–1719.
- [38] 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.

- [39] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust 3d object detection from point clouds with triple attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 677–11 684.
- [40] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [41] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.
- [42] 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.