Sparse Fuse Dense: Towards High Quality 3D Detection with Depth Completion

Xiaopei Wu[1,2], Liang Peng[1,2], Honghui Yang[1,2], Liang Xie[1], Chenxi Huang[1],
Chengqi Deng[1], Haifeng Liu[1], Deng Cai[1,2*]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Fabu Inc., Hangzhou, China
{wuxiaopei, pengliang, yanghonghui}@zju.edu.cn

Abstract

Current LiDAR-only 3D detection methods inevitably suffer from the sparsity of point clouds. Many multi-modal methods are proposed to alleviate this issue, while different representations of images and point clouds make it difficult to fuse them, resulting in suboptimal performance. In this paper, we present a novel multi-modal framework ***SFD*** *(Sparse Fuse Dense), which utilizes pseudo point clouds generated from depth completion to tackle the issues mentioned above. Different from prior works, we propose a new RoI fusion strategy* ***3D-GAF*** *(3D Grid-wise Attentive Fusion) to make fuller use of information from different types of point clouds. Specifically, 3D-GAF fuses 3D RoI features from the pair of point clouds in a grid-wise attentive way, which is more fine-grained and more precise. In addition, we propose a* ***SynAugment*** *(Synchronized Augmentation) to enable our multi-modal framework to utilize all data augmentation approaches tailored to LiDAR-only methods. Lastly, we customize an effective and efficient feature extractor* ***CPConv*** *(Color Point Convolution) for pseudo point clouds. It can explore 2D image features and 3D geometric features of pseudo point clouds simultaneously. Our method holds the highest entry on the KITTI car 3D object detection leaderboard[†], demonstrating the effectiveness of our SFD. Code will be made publicly available.*

1. Introduction

In recent years, the rise of deep learning and autonomous driving has led to a rapid development of 3D detection. Current 3D detection methods are mainly based on LiDAR point clouds [1, 3, 6, 22, 23, 29, 30, 42, 43, 50], while the sparsity of point clouds considerably limits their performances. The sparse LiDAR point clouds provide poor information in distant and occluded regions, making it difficult to generate precise 3D bounding boxes. Many multi-modal methods are proposed to address this problem. MV3D [2] introduces an RoI fusion strategy to fuse features of images and point clouds on the second stage. AVOD [15] proposes to fuse full resolution feature crops from the image feature maps and BEV feature maps for a high recall. MMF [20] leverages 2D detection, ground estimation and depth completion to assist 3D detection. In MMF, pseudo point clouds are used for backbone feature fusion, and depth completion feature maps are used for RoI feature fusion. Despite their great success, they have two shortcomings.

Coarse RoI Fusion Strategy When fusing RoI features, as shown in Figure 2(a), previous RoI fusion methods concatenate *2D LiDAR RoI features* cropped from BEV LiDAR feature maps and *2D image RoI features* cropped from FOV image feature maps. We note that this RoI fusion strategy is coarse. Firstly, 2D image RoI features are usually mixed with features from other objects or backgrounds, which will confuse the model. Secondly, the RoI fusion strategy ignores object part correspondences in 2D images and 3D point clouds. In this paper, we propose a more fine-grained RoI fusion strategy **3D-GAF** (3D Grid-wise Attentive Fusion), which fuse *3D RoI features* instead of *2D RoI features* as shown in Figure 2(b). We elaborate on three advantages of 3D-GAF over previous RoI fusion methods in Section 3.3.

Insufficient Data Augmentation This shortcoming exists in most multi-modal methods. Because 2D image data cannot be operated like 3D LiDAR data, many data augmentation approaches are difficult to deploy in multi-modal methods. It is a crucial reason why multi-modal methods are usually inferior to single-modal methods [47]. To this end, we introduce our **SynAugment** (Synchronized Augmentation). We observe that after converting 2D images to 3D pseudo point clouds, the representations of images and raw point clouds are unified, suggesting that we can operate images just like raw point clouds. However, it is not enough. Some complicated data augmentation approaches such as gt-sampling [41] and local rotation [49] may cause occlusions on the FOV (field of view). It is a non-negligible issue because image features need to be extracted on the FOV. Now, it is time to jump out of the mindset. With 2D images converted to 3D pseudo point clouds, why don't we

*Corresponding author
[†]On the date of CVPR deadline, *i.e.*, Nov.16, 2021

directly extract image features in 3D space? In this way, we no longer need to consider the FOV occlusion issue.

Nevertheless, it is non-trivial to extract features of pseudo point clouds in 3D space. Thus, we present a **CPConv** (Color Point Convolution), which searches neighbors of pseudo points on the image domain. It enables us to extract both image features and geometric features of pseudo point clouds efficiently. Considering the FOV occlusion issue, we cannot project all pseudo points to the image space of current frame for neighbor search. Here we propose an *RoI-aware Neighbor Search*, which projects pseudo points in each 3D RoI to their original image space, as illustrated in Figure 3. Hence, pseudo points that occlude each other on the FOV will not become neighbors when performing neighbor search. In other words, their features will not interfere with each other.

To summarize, our contributions are listed as follows:

- We propose a new RoI feature fusion strategy **3D-GAF** to fuse RoI features from raw point clouds and pseudo point clouds in a more fine-grained manner.
- We present a data augmentation method **SynAugment** to solve the insufficient data augmentation issue that multi-modal methods suffer from.
- We customize an effective and efficient feature extractor **CPConv** for pseudo point clouds. It can extract both 2D image features and 3D geometric features.
- We demonstrate the effectiveness of our method with extensive experiments. Specially, we rank 1^{st} on the KITTI car 3D object detection leaderboard.

2. Related Work

3D Detection Using Single-modal Data. Current 3d detection methods are mainly based on LiDAR data. SECOND [41] proposes a sparse convolution operation to speed up 3D convolution. SA-SSD [10] exploits an auxiliary network to guide the features. PV-RCNN [29] leverages the advantages of voxel-based methods and point-based methods to get more discriminative features. Voxel-RCNN [4] points out that precise positioning of raw points is unnecessary. SE-SSD [50] attains an excellent performance with self-ensembling. CenterPoint [44] provides a simple but effective anchor-free framework for 3D detection. LiDAR R-CNN [19] gives an effective solution to remedy the scale ambiguity problem issue. SPG [40] generates semantic points to recover missing parts of the foreground objects. VoTr [24] presents a transformer-based architecture to capture large context information efficiently. Pyramid R-CNN [23] designs a pyramid RoI head to adaptively learn the features from the sparse points of interest. CT3D [28] devises a channel-wise transformer to capture rich contextual dependencies among points. However, LiDAR data is usually sparse, posing a challenge for these methods.

3D Detection Using Multi-modal Data. Due to the sparsity of point clouds, researchers seek help from multi-modal methods that utilize both images and point clouds. Some methods [26, 37, 39, 48] use a cascading approach to exploit multi-modal data. However, their performances are bounded by the 2D detector. MV3D [2] realizes a two-stage multi-modal framework with an RoI feature fusion strategy that uses images for RoI refinement. ContFuse [21] proposes a continuous fusion layer to fuse BEV feature maps and image feature maps. MMF [20] benefits from multi-task learning and multi-sensor fusion. VMVS [16] generates a set of virtual views for each detected pedestrian in pseudo point clouds. Then the different views are used to produce an accurate orientation estimation. 3D-CVF [45] fuses features from multi-view images. CLOCs PVCas [25] refines confidences of 3D candidates with 2D candidates in a learnable manner. Some works [13, 32, 34, 38] realize a fine-grained fusion by establishing correspondence between images and point clouds, and then indexing image features by point clouds. However, the image information they index is limited due to the sparse correspondence between images and point clouds. It is noteworthy that although MMF [20] also employs the depth completion, it does not solve the two issues mentioned in Section 1. In this paper, we make full use of pseudo point clouds and give an effective solution.

Depth Completion. Depth completion aims to predict a dense depth map from a sparse one with the guidance of a color image. Recently, many efficient depth completion methods are proposed [8, 9, 12, 14]. [12] utilizes a two-branch backbone to realize a precise and efficient depth completion network. [14] proposes a multi-hypothesis depth representation that can sharp depth boundary between foreground and background. Although the primary purpose of the depth completion task is to serve downstream tasks, there are few methods using depth completion in 3D detection. In the image-based 3D object detection, there are some works [36, 46] that use depth estimation to generate pseudo point clouds. However, their performances are greatly limited due to the lack of accurate or sufficient raw LiDAR point clouds.

3. Sparse Fuse Dense

3.1. Preliminaries

For simplicity, we name the raw LiDAR point clouds generated by LiDAR and the pseudo point clouds generated from depth completion as *raw clouds* and *pseudo clouds*, respectively. Given a frame of raw clouds $\mathcal{R}$, we can convert it into a sparse depth map $\mathcal{S}$ with a known projection $T_{\text{LiDAR}\rightarrow\text{image}}$. Let $\mathcal{I}$ denote the image that corresponds to $\mathcal{R}$. Feeding $\mathcal{I}$ and $\mathcal{S}$ to a depth completion network, we can get a dense depth map $\mathcal{D}$. With a known projection $T_{\text{image}\rightarrow\text{LiDAR}}$, we can get a frame of pseudo clouds $\mathcal{P}$.

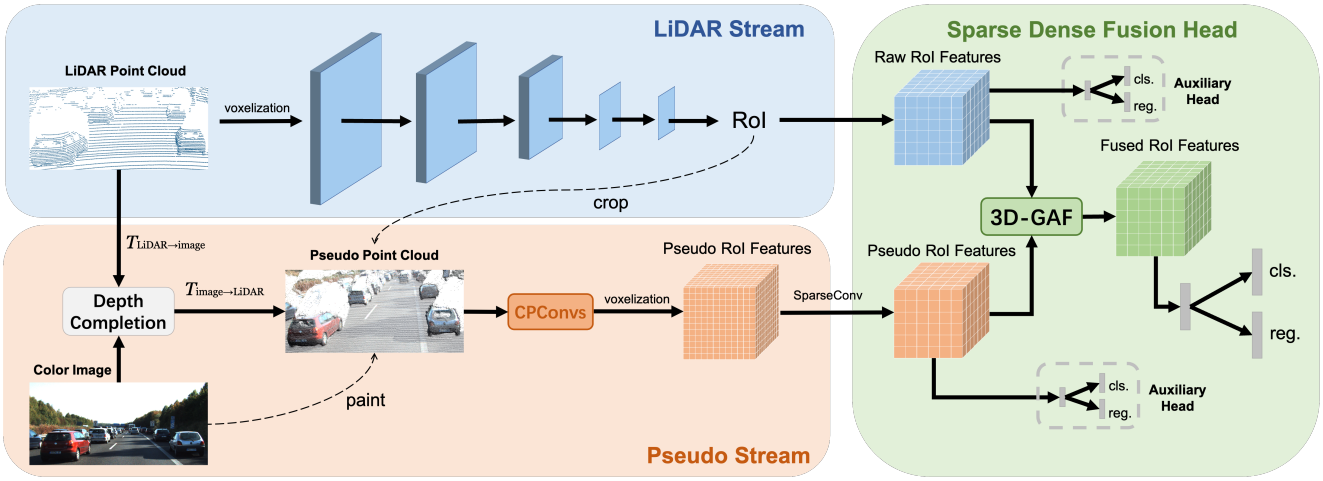

Figure 1. SFD consists of three parts: *LiDAR Stream*, *Pseudo Stream* and *Sparse Dense Fusion Head*. (1) *LiDAR Stream* only uses raw clouds to predict 3D RoIs. Then RoIs are used to crop raw clouds and pseudo clouds. (2) *Pseudo Stream* uses raw clouds and images to generate pseudo clouds. Painting pseudo clouds with RGB, we get colorful pseudo clouds. Then several CPConvs (see Section 3.5) are performed to extract rich information of pseudo clouds in RoIs. At the end of Pseudo Stream, pseudo clouds in RoIs are voxelized, and 3D sparse convolutions are applied. (3) In *Sparse Dense Fusion Head*, RoI features from raw clouds and pseudo clouds are fused by 3D-GAF (see Section 3.3), then the fused RoI features are used to predict class confidences and bounding boxes. In addition, two auxiliary heads are employed to regularize our network. They can be detached at inference time.

3.2. Overview of Methods

We show our framework in Figure 1, including: (1) a *LiDAR Stream* using only raw clouds and serving as an RPN to produce 3D RoIs; (2) a *Pseudo Stream* that extracts point features with proposed CPConv, and extracts voxel features with sparse convolutions; (3) a *Sparse Dense Fusion Head* that fuses 3D RoI features from raw clouds and pseudo clouds in a grid-wise attentive manner, and produces final predictions. We detail our method in the following sections.

3.3. 3D Grid-wise Attentive Fusion

Due to the dimensional gap between images and point clouds, previous works [2, 15, 20] directly concatenate *2D LiDAR RoI features* cropped from BEV LiDAR feature maps and *2D image RoI features* cropped from FOV image feature maps, which is a coarse RoI fusion strategy. In our method, with 2D images converted to 3D pseudo clouds, we can fuse the RoI features from images and point clouds in a more fine-grained manner, as shown in Figure 2. Our 3D-GAF consists of 3D Fusion, Grid-wise Fusion and Attentive Fusion.

(1) ***3D Fusion***. We use a 3D RoI to crop 3D raw clouds and 3D pseudo clouds, which only includes LiDAR features and image features in the 3D RoI, as shown in Figure 2(b). Previous methods use 2D RoI to crop image features, which will involve features from other objects or backgrounds. It causes a lot of interference, especially for occluded objects, as shown in Figure 2(a). (2) ***Grid-wise Fusion***. In previous RoI fusion methods, there are no correspondences between image RoI grids and LiDAR RoI grids, so they directly concatenate image RoI features and LiDAR RoI features. In

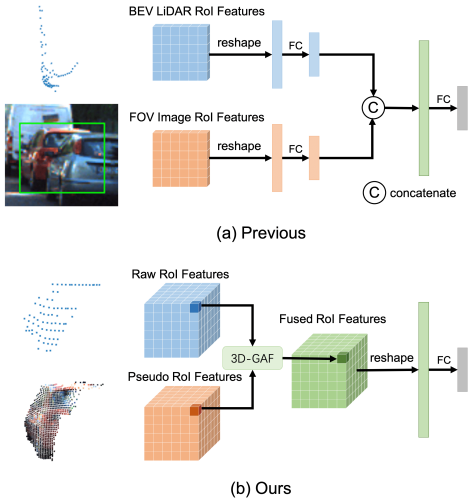

Figure 2. Comparison between previous methods and 3D-GAF.

our methods, thanks to the same representation of raw RoI features and pseudo RoI features, we can fuse each pair of grid features separately. It enables us to accurately enhance each part of an object with the corresponding pseudo grid features. (3) ***Attentive Fusion***. Aiming to fuse each pair of grid features from raw RoI and pseudo RoI adaptively, we utilize a simple attention module motivated by [11, 13, 18]. Generally, we predict a pair of weights for each pair of grids and weight the pair of grid features with the weights to get the fused grid features. To validate the effectiveness of 3D Fusion, Grid-wise Fusion and Attentive Fusion, we provide ablation studies in Section 4.

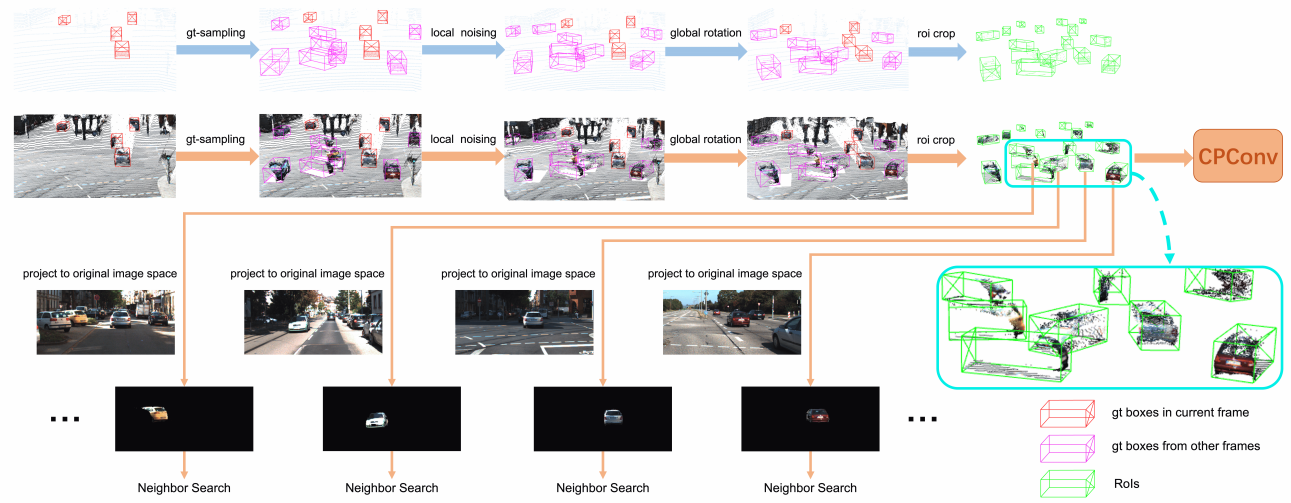

Figure 3. Illustration of SynAugment and RoI-aware Neighbor Search in CPConv. We show original gt boxes, sampled gt boxes and RoIs in red, purple and green, respectively. For the convenience of visualization, we only show 3 data augmentation approaches, and we remove some redundant and low score RoIs.

Here we provide a detailed description of our 3D-GAF. Let $\mathbf{b}$ denote a single 3D RoI. We denote $F^{\text{raw}} \in \mathbb{R}^{n \times C}$ and $F^{\text{pse}} \in \mathbb{R}^{n \times C}$ as the raw cloud RoI feature and pseudo cloud RoI feature in $\mathbf{b}$, respectively. Here n ($6 \times 6 \times 6$ by default, following our baseline Voxel-RCNN [4]) is the total number of grids in a 3D RoI, and C is the grid feature channel. The i^{th} RoI grid feature of F^{raw} and F^{pse} are denoted as F_i^{raw} and F_i^{pse}, respectively. Given a pair of RoI grid features $(F_i^{\text{raw}}, F_i^{\text{pse}})$, we concatenate the F_i^{raw} and F_i^{pse}. Then the result is fed to a fully connected layer and a sigmoid layer, producing a pair of weights $(w_i^{\text{raw}}, w_i^{\text{pse}})$ for the pair of grid features, where w_i^{raw} and w_i^{pse} are all scalars. Finally, we weight $(F_i^{\text{raw}}, F_i^{\text{pse}})$ with $(w_i^{\text{raw}}, w_i^{\text{pse}})$ to get the fused grid feature F_i. Formally, F_i is attained as follow:

$$(w_i^{\text{raw}}, w_i^{\text{pse}}) = \sigma(\text{MLP}(\text{CONCAT}(F_i^{\text{raw}}, F_i^{\text{pse}}))) \quad (1)$$

$$F_i = \text{MLP}(\text{CONCAT}(w_i^{\text{raw}} F_i^{\text{raw}}, w_i^{\text{pse}} F_i^{\text{pse}})) \quad (2)$$

In practice, all pairs of RoI grid features in a batch can be processed in parallel, so our 3D-GAF is efficient.

3.4. Synchronized Augmentation

Due to the different representations of images and point clouds, it is difficult for multi-modal methods to utilize many data augmentation approaches, such as gt-sampling [41] and local noising [49]. Insufficient data augmentation greatly limits the performance of many multi-modal methods. Therefore, we present a multi-modal data augmentation method SynAugment to enable our SFD to use all data augmentation approaches tailored to LiDAR-only methods. Concretely, SynAugment consists of two-folds: *manipulate images like point clouds* and *extract image features in 3D Space*.

Manipulate Images like Point Clouds The greatest challenge of multi-modal data augmentation is how to manipulate images like point clouds. Depth completion gives the answer. With depth completion, 2D images can be converted into 3D pseudo clouds. Painting pseudo clouds with RGB, the pseudo clouds carry all information of images. Then we only need to perform data augmentation on pseudo clouds as same as raw clouds, as shown at the top of Figure 3.

Extract Image Features in 3D Space Manipulating images like point clouds is not enough for multi-modal data augmentation. Currently, most multi-modal methods need to extract image features on the FOV images. Nevertheless, that will restrict the model from using data augmentation methods (such as gt-sampling and local rotation) that may cause the FOV occlusion issue. To address this problem, we propose to extract image features in 3D space with 2D images converted to 3D pseudo clouds. In this way, it is unnecessary to consider the occlusion issue because we no longer extract image features on the FOV images. To extract features in 3D space, we can use 3D sparse convolutions. However, there is a more effective method (see Section 3.5).

It is noteworthy that [35, 47] can realize multi-modal gt-sampling by performing additional occlusion detection on images, while they are not suitable for more complicated data augmentation, which cannot be simply solved by occlusion detection, such as local noising [49] and SA-DA [50]. Some works [34, 38] that project image segmentation scores to raw clouds can also use multi-modal data augmentation, but the image information carried by raw clouds is sparse due to the sparse correspondence between images and point clouds. In our method, the image information of each gt sampler is dense because we can crop complete image information of samplers in pseudo clouds.

3.5. Color Point Convolution

Definition For a frame of pseudo clouds $\mathcal{P}$, we concatenate the RGB (r, g, b) and coordinate (u, v) of each pixel in the image to its corresponding pseudo point. Therefore, the i^{th} pseudo point p_i can be represented as $(x_i, y_i, z_i, r_i, g_i, b_i, u_i, v_i)$.

A naive approach to extract features of pseudo clouds is directly voxelizing the pseudo clouds and performing 3D sparse convolutions, while it actually does not fully explore the rich semantic and structural information in pseudo clouds. PointNet++ [27] is a good example for extracting features of points, but it is not suitable for pseudo clouds. *Firstly*, the ball query operation in PointNet++ will bring massive calculations due to the vast amounts of pseudo points. *Secondly*, PointNet++ cannot extract 2D features because the ball query operation does not take 2D neighborhood relationships into account. In light of this, we need a feature extractor that can efficiently extract both 2D semantic features and 3D structural features.

RoI-aware Neighbor Search on the Image Domain Based on the above analysis, we propose a CPConv (Color Point Convolution), which searches neighbors on the *image domain*, as inspired by the voxel query [4] and grid search [5]. In this way, we can overcome the shortcomings of PointNet++. *Firstly*, a pseudo point can search its neighbors in constant time, making it much faster than the ball query. *Secondly*, neighborhood relationships on the image domain make it possible to extract 2D semantic features.

Unfortunately, we cannot project all pseudo points to current frame image space for neighbor search, because with gt-sampling, pseudo points coming from other frames may cause FOV occlusions. To this end, we propose an *RoI-aware Neighbor Search*. Concretely, we project pseudo points in each 3D RoI to their original image space separately according to the (u, v) attribute carried on pseudo points, as shown at the bottom of Figure 3. In this way, pseudo points occluded by each other will not become neighbors, so their features will not interfere with each other even if there are heavy occlusions between them on the FOV.

Pseudo Point Features For the i^{th} pseudo point p_i, we denote the feature of p_i as $f_i = (x_i, y_i, z_i, r_i, g_i, b_i)$, which consists of 3D geometric features (x_i, y_i, z_i) and 2D semantic features (r_i, g_i, b_i). As motivated by [4], we apply a fully connected layer on pseudo point features before performing the neighbor search to reduce the complexity. After the fully connected layer, the feature channel is raised to C_3, as shown in Figure 4.

Position Residuals We utilize 3D and 2D position residuals from p_i to its neighbors to make p_i's features aware of local relationships in 3D and 2D space, which is particularly important for extracting both 3D structural features and 2D semantic features of p_i. For p_i's k^{th} neighbor ${p_i}^k$, the position residual between p_i and ${p_i}^k$ is represented as ${h_i}^k = (x_i - {x_i}^k, y_i - {y_i}^k, z_i - {z_i}^k, u_i - {u_i}^k, v_i - {v_i}^k, ||p_i - {p_i}^k||)$, where $||p_i - {p_i}^k|| = \sqrt{(x_i - {x_i}^k)^2 + (y_i - {y_i}^k)^2 + (z_i - {z_i}^k)^2}$.

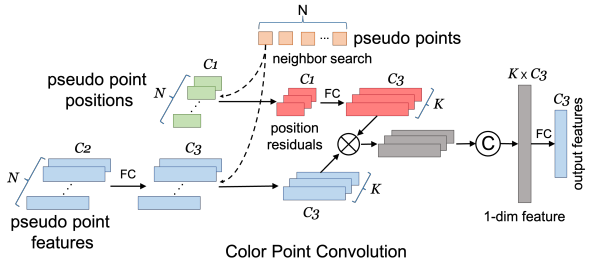

Figure 4. Illustration of CPConv.

Feature Aggregation For K neighbors of p_i, we gather their positions and calculate position residuals. Then we apply a fully connected layer on position residuals, raising their channels to C_3 to align with pseudo point features. Given a set of neighbor features $F_i = \{f_i^k \in \mathbb{R}^{C_3}, k \in 1, \cdots, K\}$ and a set of neighbor position residuals $H_i = \{h_i^k \in \mathbb{R}^{C_3}, k \in 1, \cdots, K\}$, we weight each ${f_i}^k$ with corresponding ${h_i}^k$. The weighted neighbor features are concatenated [5] instead of max-pooled [4] for maximum information fidelity. Finally, a fully connected layer is applied to map aggregated feature channel back to C_3.

Multi-Level Feature Fusion We stack three CPConvs to extract deeper features of pseudo clouds. Considering that high-level features provide a larger receptive field and richer semantic information, while low-level features can supply finer structure information, we concatenate the output of each CPConv to get a more comprehensive and discriminative representation for pseudo clouds.

3.6. Loss Function

We follow the RPN loss and RoI head loss of Voxel-RCNN [4], which are denoted as $\mathcal{L}_{\text{rpn}}$ and $\mathcal{L}_{\text{roi}}$, respectively. To prevent gradients from being dominated by a particular *Stream*, we add auxiliary RoI head loss on both *LiDAR Stream* and *Pseudo Stream*, which are denoted as $\mathcal{L}_{\text{aux}_1}$ and $\mathcal{L}_{\text{aux}_2}$, respectively. $\mathcal{L}_{\text{aux}_1}$ and $\mathcal{L}_{\text{aux}_2}$ are consistent with $\mathcal{L}_{\text{roi}}$, including class confidence loss and regression loss. The depth completion network loss $\mathcal{L}_{\text{depth}}$ follows the definition of [12]. Then the total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{roi}} + \lambda_1 \mathcal{L}_{\text{aux}_1} + \lambda_2 \mathcal{L}_{\text{aux}_2} + \beta \mathcal{L}_{\text{depth}} \quad (3)$$

where λ_1, λ_2 and β are the weight of $\mathcal{L}_{\text{aux}_1}$, $\mathcal{L}_{\text{aux}_2}$ and $\mathcal{L}_{\text{depth}}$ ($\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\beta = 1$ by default). More details about the methods that we propose in this paper are provided in the supplementary material.

4. Experiments

4.1. Dataset and Evaluation Metrics

We evaluate our method on the KITTI 3D and BEV object detection benchmark [7]. The KITTI dataset consists of

7481 training samples and 7518 testing samples in the object detection task. The training data are divided into a *train* set with 3712 samples and a *val* set with 3769 samples. For experimental studies, we use the *train* set and *val* set for training and evaluating, respectively. The results on the *val* set and *test* set are evaluated with the average precision calculated by 40 recall positions. We also provide results on *val* set with AP calculated by 11 recall positions for a fair comparison with previous works. For the reason that Waymo and NuScenes datasets have not yet generated depth labels for the depth completion task, we do not conduct experiments on these two datasets.

4.2. Implementation Details

The *LiDAR Stream* of SFD is based on Voxel-RCNN [4]. For the depth completion, we use [12]. SFD can also achieve comparable results with [14] as our depth completion network. We follow the data augmentation approaches mentioned in [4] (gt-sampling, global rotation, global flipping and global scaling) and [49] (local noising and training with similar class). Although our SFD can be trained end-to-end without the depth completion network pre-trained, we observe that initialization is essential for the performance of 3D detection. Thus, we pre-train the depth completion network on the KITTI dataset and fix the parameters of the depth completion network when training our SFD.

4.3. Comparison with State-of-the-Arts

We compare our SFD with state-of-the-art methods on the KITTI $test$ set by submitting our results to KITTI online test server. As shown in Table 1, our method achieves remarkable results. We surpass all state-of-the-art multi-modal methods by a large margin. For LiDAR-only methods, we improve our baseline Voxel-RCNN by 3.14% AP on the moderate metric and outperform published best method SE-SSD [50] by 2.22% and 1.07% AP on the moderate and mAP metric, respectively. As of Nov.16, 2021, our method ranks 1^{st} on the highly competitive KITTI car 3D detection benchmark. Besides, we provide a comparison on the KITTI val set, as seen in Table 2. In BEV detection, SFD is still in the leading position, as shown in Table 3. We improve Voxel-RCNN by 3.02% AP on the moderate metric and achieve comparable results with the state-of-the-art method SE-SSD.

4.4. Ablation Study

Here we provide extensive experiments to analyze the effectiveness of our method. In Table 4, experiment (a) is our baseline modified on Voxel-RCNN [4]. It only uses raw clouds as input. Experiments (b) and (c) are all equipped with our multi-modal data augmentation method SynAugment for a fair comparison with experiment (a), which is equipped with single-modal data augmentation.

Method	Modality	3D			
		mAP	Easy	Mod.	Hard
SECOND [41]	LiDAR	73.90	83.34	72.55	65.82
PointPillars [17]	LiDAR	75.29	82.58	74.31	68.99
Part-A^2 [31]	LiDAR	79.94	87.81	78.49	73.51
SA-SSD [10]	LiDAR	80.90	88.75	79.79	74.16
PV-RCNN [29]	LiDAR	82.83	90.25	81.43	76.82
Voxel-RCNN [4]	LiDAR	83.19	90.90	81.62	77.06
CT3D [28]	LiDAR	82.25	87.83	81.77	77.16
Pyramid R-CNN [23]	LiDAR	82.65	88.39	82.08	77.49
VoTr-TSD [24]	LiDAR	83.71	89.90	82.09	**79.14**
SPG [40]	LiDAR	83.84	90.50	82.13	78.90
SE-SSD [50]	LiDAR	83.73	91.49	82.54	77.15
MV3D [2]	LiDAR+RGB	64.20	74.97	63.63	54.00
ContFuse [21]	LiDAR+RGB	71.38	83.68	68.78	61.67
F-PointNet [26]	LiDAR+RGB	70.86	82.19	69.79	60.59
AVOD [15]	LiDAR+RGB	73.52	83.07	71.76	65.73
PI-RCNN [38]	LiDAR+RGB	76.41	84.37	74.82	70.03
UberATG-MMF [20]	LiDAR+RGB	78.68	88.40	77.43	70.22
EPNet [20]	LiDAR+RGB	81.23	89.81	79.28	74.59
3D-CVF [45]	LiDAR+RGB	80.79	89.20	80.05	73.11
CLOCs PVCas [25]	LiDAR+RGB	82.25	88.94	80.67	77.15
SFD (ours)	LiDAR+RGB	**84.80**	**91.73**	**84.76**	77.92

Table 1. Comparison with state-of-the-art methods on the KITTI *test* set for car 3D detection, with average precisions of 40 sampling recall points evaluated on the KITTI server.

Method	$3D_{R11}$			$3D_{R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
PV-RCNN [29]	89.35	83.69	78.70	92.57	84.83	82.69
Pyramid-PV [23]	89.37	84.38	78.84	-	-	-
Voxel-RCNN [4]	89.41	84.52	78.93	92.38	85.29	82.86
SE-SSD [50]	-	85.71	-	93.19	86.12	83.31
UberATG-MMF [20]	88.40	77.43	70.22	-	-	-
3D-CVF [45]	-	-	-	89.67	79.88	78.47
EPNet [13]	-	-	-	92.28	82.59	80.14
CLOCs PVCas [25]	-	-	-	92.78	85.94	83.25
SFD (ours)	**89.74**	**87.12**	**85.20**	**95.47**	**88.56**	**85.74**

Table 2. Comparison with state-of-the-art methods on the KITTI *val* set for car 3D detection. The results are evaluated with the average precision calculated by 11 and 40 recall positions.

Method	Modality	BEV			
		mAP	Easy	Mod.	Hard
Voxel-RCNN [4]	LiDAR	89.94	94.85	88.83	86.13
SA-SSD [10]	LiDAR	90.67	95.03	91.03	85.96
SE-SSD [50]	LiDAR	91.41	**95.68**	91.84	86.72
EPNet [20]	LiDAR+RGB	88.79	94.22	88.47	83.69
3D-CVF [45]	LiDAR+RGB	88.51	93.52	89.56	82.45
CLOCs PVCas [25]	LiDAR+RGB	89.81	93.05	89.80	86.57
SFD (ours)	LiDAR+RGB	**91.44**	95.64	**91.85**	**86.83**

Table 3. Comparison with state-of-the-art methods on the KITTI *test* set for car BEV detection, with average precisions of 40 sampling recall points evaluated on the KITTI server.

Effect of 3D-GAF Experiment (b) in Table 4 exploit 3D-GAF to fuse RoI features, making 0.61%, 1.10% and 2.32% AP improvement on easy, moderate and hard levels, respectively. To extract pseudo RoI features, we simply voxelize pseudo clouds and perform 3D sparse convolutions.

Effect of CPConv Experiment (c) in Table 4 uses CPConv to extract richer features of pseudo clouds based on experiment (b), yielding a moderate AP of 88.56% with 1.99% AP improvement, manifesting the effectiveness of CPConv.

Experiment	3D-GAF	CPConv	AP_{3D}		
			Easy	Mod.	Hard
(a)			92.88	85.47	82.98
(b)	√		93.49	86.57	85.30
(c)	√	√	**95.47**	**88.56**	**85.74**

Table 4. Effects of different components on the KITTI *val* set. The results are evaluated with the AP calculated by 40 recall positions for car class. "3D-GAF" and "CPConv" stand for 3D Grid-wise Attentive Fusion and Color Point Convolution, respectively.

Effect of SynAugment Our SynAugment enables our multi-modal framework to utilize the data augmentation approaches tailored only for LiDAR-only methods such as gt-sampling, local nosing and global scaling. We take off these data augmentation approaches from experiments (a) and (b) in Table 4, resulting in experiments (a) and (b) in Table 5. As shown in Table 5, without multi-modal data augmentation, the performance of our method drops drastically, which proves the importance of sufficient data augmentation for multi-modal methods.

Experiment	Data Augmentation	AP_{3D}		
		Easy	Mod.	Hard
(a)	Yes	**92.88**	**85.47**	**82.98**
	No	88.55	78.49	74.42
(b)	Yes	**93.49**	**86.57**	**85.30**
	No	90.88	80.31	77.87

Table 5. Ablation study on SynAugment. The results are evaluated with the AP calculated by 40 recall positions for car class.

Ablation Study on 3D Grid-wise Attentive Fusion We conduct an experiment to verify the effectiveness of each part of 3D-GAF, as shown in Table 6. Experiment (a) directly concatenates raw RoI features and pseudo RoI features cropped by 2D RoIs, which we call *2D RoI-wise Concat Fusion*. Experiment (b) concatenates raw RoI features and pseudo RoI features cropped by 3D RoIs, which we call *3D RoI-wise Concat Fusion*. Experiment (c) fuses a pair of RoI features in a grid-wise manner based on experiment (b), which we call *3D Grid-wise Concat Fusion*. Experiment (d) extends (c) with Attentive Fusion, which is our *3D Grid-wise Attentive Fusion*. Results show that each part of 3D-GAF can improve our SFD. Moreover, we find that the contribution of Grid-wise Fusion and Attentive Fusion mainly lie on the moderate level and easy level, respectively.

Experiment	3D	Grid-wise	Attentive	AP_{3D}		
				Easy	Moderate	Hard
(a)				93.08	85.27	82.79
(b)	√			94.83	87.77	85.27
(c)	√	√		94.84	88.23	85.57
(d)	√	√	√	**95.47**	**88.56**	**85.74**

Table 6. Ablation study on 3D-GAF. "3D": 3D Fusion. "Grid-wise": Grid-wise Fusion. "Attentive": Attentive Fusion. The results are calculated by 40 recall positions for car class.

Cooperating with Different Detectors To validate the universality of our method, we equip different LiDAR-only detectors with our SFD framework. In our experiments, we use the PointRCNN [30], Part-A^2 [31] and SECOND [41] implemented by OpenPCDet [33]. Table 7 suggests that our method can improve different detectors significantly. For the one-stage detector SECOND, we use the same architecture as *Pseudo Stream* (CPConvs with sparse convolutions) to extract features of raw clouds in 3D RoIs. The raw clouds are also painted with RGB to be consistent with pseudo clouds.

Method	with SFD	AP_{3D}		
		Easy	Mod.	Hard
PointRCNN	No	91.40	82.33	80.09
	Yes	**94.50**	**85.72**	**83.29**
	Improvement	*+3.10*	*+3.39*	*+3.20*
Part-A^2	No	91.87	82.74	80.42
	Yes	**93.17**	**85.91**	**83.56**
	Improvement	*+1.30*	*+3.17*	*+3.14*
SECOND	No	90.31	81.76	78.88
	Yes	**94.75**	**87.20**	**85.07**
	Improvement	*+4.44*	*+5.44*	*+6.19*

Table 7. Cooperating with different detectors. The average precisions are calculated by 40 recall positions.

Conditional Analysis To figure out in what cases our method improves the baseline most, we evaluate our SFD on different distances and different occlusion degrees. As shown in Table 8, distant and heavily occluded objects are improved most, which verifies our hypothesis that pseudo point clouds are helpful for objects with sparse raw points.

with SFD	Distance			Occlusion		
	0-20m	20-40m	40m-Inf	0	1	2
No	94.42	77.05	15.03	62.49	76.79	57.46
Yes	**95.28**	**79.34**	**21.91**	**63.46**	**80.03**	**62.68**
Improvement	*+0.86*	*+2.29*	*+6.88*	*+0.97*	*+3.24*	*+5.22*

Table 8. Performance on different distances and different occlusion degrees. The results are evaluated with 3D AP calculated by 40 recall positions for car class on the moderate level.

Inference Speed We test the inference speed of our SFD on an NVIDIA RTX 2080 Ti GPU. With the depth completion network, the speed of SFD is 10.2 FPS. Because SFD is a multi-modal detector, it is inevitably slower than some single-modal methods. However, in multi-modal methods, SFD is actually not slow, as shown in Table 9.

SFD	PointPainting [34]	F-PointNet [26]	EPNet [13]	3D-CVF [45]
10.2 FPS	2.5 FPS	5.9 FPS	10 FPS	16.7 FPS

Table 9. Inference speed of different multi-modal methods.

Training with Three Classes To further validate the effectiveness of our SFD, we train a single model for car, pedestrian and cyclist detection. As seen in Figure 10, SFD can consistently improve Voxel-RCNN.

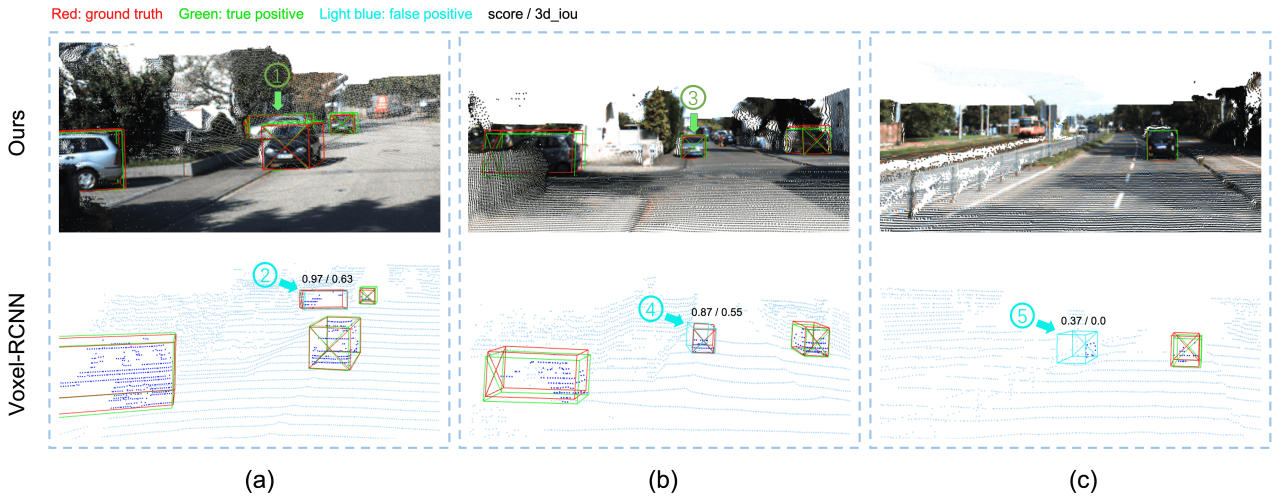

Figure 5. Comparison between our SFD and Voxel-RCNN. For the visualization of SFD and Voxel-RCNN, we use pseudo clouds and raw clouds, respectively. We show ground-truth boxes, true positives and false positives in red, green and light blue, respectively. Green arrows represent that our predictions are more accurate, and light blue arrows represent false positives of Voxel-RCNN.

Class	with SFD	AP_{3D}			AP_{BEV}		
		Easy	Mod.	Hard	Easy	Mod.	Hard
Car	No	89.39	83.83	78.73	90.26	88.35	87.81
	Yes	**95.52**	**88.27**	**85.57**	**96.24**	**92.09**	**91.32**
	Improvement	*+6.13*	*+4.44*	*+6.84*	*+5.98*	*+3.74*	*+3.51*
Pedestrian	No	70.55	62.92	57.35	71.62	64.95	61.11
	Yes	**72.94**	**66.69**	**61.59**	**75.64**	**69.71**	**64.75**
	Improvement	*+2.39*	*+3.77*	*+4.24*	*+4.02*	*+4.76*	*+3.64*
Cyclist	No	90.04	71.13	66.67	91.71	74.67	70.02
	Yes	**93.39**	**72.95**	**67.26**	**93.37**	**75.31**	**70.80**
	Improvement	*+3.35*	*+1.82*	*+0.59*	*+1.66*	*+0.64*	*+0.78*

Table 10. Performance of SFD on the KITTI *val* set for car, pedestrian and cyclist with AP calculated by 40 recall positions.

4.5. Qualitative Results and Analysis

Figure 5 shows the visualization of predictions by our SFD and Voxel-RCNN [4]. It provides 3 cases corresponding to 3 situations where SFD improves Voxel-RCNN.

Occlusion Occlusion is a challenging problem in the scenario of autonomous driving, as shown in Figure 5(a). Object ① is heavily occluded by the black car in front, making raw clouds on it insufficient (see ②). Fortunately, pseudo clouds can alleviate this by providing sufficient 3D geometric information and additional 2D semantic information.

Long Distance Figure 5(b) shows another common scene. Due to the limited resolution of LiDAR, faraway objects are with much fewer points. It is difficult to predict a precise box for objects (such as ④) with sparse raw clouds. However, pseudo clouds on the object are richer. Figure 6 shows different views of pseudo clouds on ③, demonstrating that pseudo clouds are qualified to provide supplementary information for raw clouds.

Background similar to Foreground Dense pseudo clouds not only benefit locating foreground but also help to distinguish the background from the foreground. Some background raw clouds are very similar to the foreground

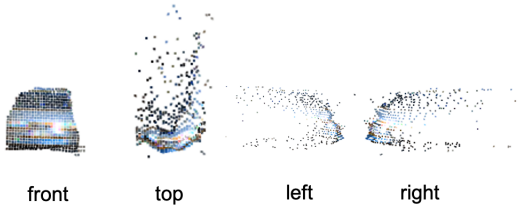

Figure 6. Different views of object ③ in Figure 5(b).

because of the sparsity of raw clouds, which may confuse detectors and cause a lot of false positives. As seen in Figure 5(c), Voxel-RCNN mistakes the fence for a car because raw clouds on the fence and car are similar. Nevertheless, pseudo clouds on them are very different, which helps our SFD to distinguish them.

5. Conclusion

In this paper, we propose a novel multi-modal framework SFD for high quality 3D detection. We design a new RoI fusion strategy 3D-GAF to fuse raw clouds and pseudo clouds in a more fine-grained manner. With the proposed SynAugment, our SFD can use data augmentation methods tailored to LiDAR-only methods. Besides, we design a CPConv to effectively and efficiently extract features of pseudo clouds. Experimental results demonstrate that our approach can significantly improve detection accuracy.

Acknowledgments This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, U1909203, 61936006, 61973271), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05).

References

[1] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Dragomir Anguelov. To the point: Efficient 3d object detection in the range image with graph convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2021. 1

[2] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017. 1, 2, 3, 6

[3] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point R-CNN. In *ICCV*, 2019. 1

[4] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv preprint arXiv:2012.15712*, 2020. 2, 4, 5, 6, 8

[5] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. *arXiv preprint arXiv:2103.10039*, 2021. 5

[6] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Real-time anchor-free single-stage 3d detection with iou-awareness. *arXiv preprint arXiv:2107.14342*, 2021. 1

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[8] Jiaqi Gu, Zhiyu Xiang, Yuwen Ye, and Lingxuan Wang. Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE Robotics and Automation Letters*, 6(2):1808–1815, 2021. 2

[9] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11078–11088, 2021. 2

[10] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3D object detection from point cloud. In *CVPR*, pages 11873–11882, 2020. 2, 6

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[12] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783*, 2021. 2, 5, 6

[13] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2020. 2, 3, 6, 7

[14] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021. 2, 6

[15] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3D proposal generation and object detection from view aggregation. *CoRR*, 2017. 1, 3, 6

[16] Jason Ku, Alex D Pon, Sean Walsh, and Steven L Waslander. Improving 3d object detection for pedestrians with virtual multi-view synthesis orientation estimation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3459–3466. IEEE, 2019. 2

[17] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 6

[18] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019. 3

[19] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7546–7555, 2021. 2

[20] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019. 1, 2, 3, 6

[21] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *ECCV*, 2018. 2, 6

[22] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11677–11684, 2020. 1

[23] Jiageng Mao, Minzhe Niu, Haoyue Bai, Xiaodan Liang, Hang Xu, and Chunjing Xu. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. *arXiv preprint arXiv:2109.02499*, 2021. 1, 2, 6

[24] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. *arXiv preprint arXiv:2109.02497*, 2021. 2, 6

[25] Su Pang, Daniel Morris, and Hayder Radha. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. *arXiv preprint arXiv:2009.00784*, 2020. 2, 6

[26] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2, 6, 7

[27] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 5

[28] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. *arXiv preprint arXiv:2108.10723*, 2021. 2, 6

[29] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-

voxel feature set abstraction for 3D object detection. In *CVPR*, pages 10529–10538, 2020. 1, 2, 6

[30] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 1, 7

[31] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 6, 7

[32] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019. 2

[33] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 7

[34] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 2, 4, 7

[35] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021. 4

[36] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian 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*, pages 8445–8453, 2019. 2

[37] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *arXiv preprint arXiv:1903.01864*, 2019. 2

[38] Liang Xie, Chao Xiang, Zhengxu Yu, Guodong Xu, Zheng Yang, Deng Cai, and Xiaofei He. PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module. In *AAAI*, pages 12460–12467, 2020. 2, 4, 6

[39] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018. 2

[40] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021. 2, 6

[41] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 4, 6, 7

[42] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-dense 3D object detector for point cloud. In *ICCV*, pages 1951–1960, 2019. 1

[43] Maosheng Ye, Shuangjie Xu, and Tongyi Cao. Hvnet: Hybrid voxel network for lidar based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[44] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, June 2021. 2

[45] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection. In *ECCV*, 2020. 2, 6, 7

[46] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019. 2

[47] Wenwei Zhang, Zhe Wang, and Chen Change Loy. Multi-modality cut and paste for 3d object detection. *arXiv preprint arXiv:2012.12741*, 2020. 1, 4

[48] Xin Zhao, Zhe Liu, Ruolan Hu, and Kaiqi Huang. 3d object detection using scale invariant and feature reweighting networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9267–9274, 2019. 2

[49] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. CIA-SSD: Confident IoU-aware single-stage object detector from point cloud. In *AAAI*, 2021. 1, 4, 6

[50] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021. 1, 2, 4, 6