SPARSE FUSE DENSE: TOWARDS HIGH QUALITY 3D DETECTION WITH DEPTH COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current LiDAR-only 3D detection methods inevitably suffer from the sparsity of point clouds. Sparse point clouds can confuse detectors as they lack sufficient geometric and semantic information. 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 new multi-modal framework named **SFD** (Sparse Fuse Dense) to tackle these issues. Specifically, we propose to enhance *sparse* point clouds generated from LiDAR with *dense* pseudo point clouds generated from depth completion. To make full use of information from different types of point clouds, we design a new RoI feature fusion method *3D-GAF (3D Grid-wise Attentive Fusion)*, which fuses 3D RoI features from the couple of point clouds in a grid-wise attentive way. In addition, we devise a *CPFE (Color Point Feature Extractor)* to extract both 3D geometric and 2D semantic features in pseudo point clouds. Moreover, we introduce a multi-modal data augmentation method named *SynAugment* to utilize all data augmentation approaches tailored to LiDAR-only methods. Our method holds the highest entry on the KITTI 3D object detection leaderboard*, demonstrating the effectiveness of SFD. Codes will be public.

1 INTRODUCTION

In recent years, the rise of deep learning and autonomous driving has led to a rapid development of 3D detection. Many excellent 3D detection methods have been proposed Yan et al. (2018); Shi et al. (2020a); Deng et al. (2020); Zheng et al. (2021b). Current 3D detection models are mainly based on raw LiDAR point clouds, while the sparsity of point clouds considerably limits their performances. The sparse LiDAR point clouds provide poor information in far and occluded regions, making it difficult to generate precise 3D boxes. To solve this problem, researchers typically resort to fusing visual features from RGB images. Nevertheless, with more data, more annotations and more time, current multi-modal methods perform less accurately than LiDAR-only methods. We summarize the main reasons into three points: *dimension gap*, *information loss*, *data augmentation*.

Dimension Gap There is an inherent dimension gap between images and point clouds, indicating that it is hard to fuse two-dimensional images and three-dimensional point clouds directly. Some methods Xu et al. (2018); Zhao et al. (2019) crop and reshape image RoI features to fuse with point cloud features, ignoring the correspondence between 2D pixels and 3D points, consequently leading to suboptimal performance.

Information Loss Some methods resolve the dimension gap by establishing correspondence between images and point clouds Liang et al. (2018; 2019); Vora et al. (2020); Xie et al. (2020); Huang et al. (2020). However, the sparse correspondence caused by sparse point clouds makes the extracted image features sparse, posing a lot of image information loss.

Data Augmentation The last serious issue is insufficient data augmentation in multi-modal methods. Complicated data augmentation approaches, such as gt-sampling Yan et al. (2018), random rotation and random scaling, are difficult to deploy in multi-modal methods because 2D image data cannot be operated like 3D LiDAR data. However, data augmentation is essential because it can largely improve the generalization ability of models.

*On the date of ICLR deadline, *i.e.*, Oct 6, 2021

To generally resolve the aforementioned problems, in this paper, we propose a novel multi-modal framework named SFD, which aims to enhance raw LiDAR point clouds with dense pseudo point clouds generate from depth completion. As shown in Figure 1, pseudo points on objects are much more than raw points in all distance ranges. Pseudo point clouds can provide sufficient information, especially for distant and occluded objects, as shown in Figure 2, demonstrating that enhancing raw LiDAR point clouds with pseudo point clouds is reasonable.

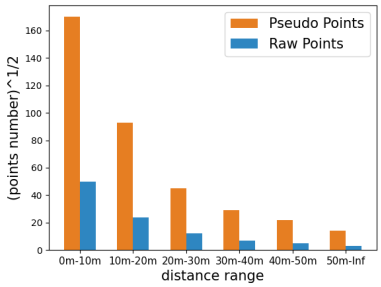

Figure 1: Average number of pseudo points and raw LiDAR points on objects in different distance ranges.

As for the three issues in multi-modal methods, we observe that pseudo point clouds have the same representation as raw LiDAR point clouds. Therefore the dimension gap issue can be eliminated naturally. Besides, pseudo point clouds carry all image information, allowing us to use all image information when fusing the couple of clouds, rather than rely on the sparse correspondence between images and raw LiDAR point clouds. For the data augmentation issue, we solve it by performing the same transformation as raw LiDAR point clouds on pseudo point clouds (see our SynAugment in Sec. 3.5).

In addition, aiming to fully fuse dense pseudo point clouds and raw LiDAR point clouds, we propose an effective RoI fusion method named 3D-GAF (3D Grid-wise Attentive Fusion), which fuses 3D RoI features from the couple of clouds in a grid-wise attentive way. Moreover, to explore both 2D semantic features and 3D geometric features carried by pseudo point clouds in 3D RoIs, we present a CPFE (Color Point Feature Extractor) that takes both 2D and 3D neighborhood relationships into account.

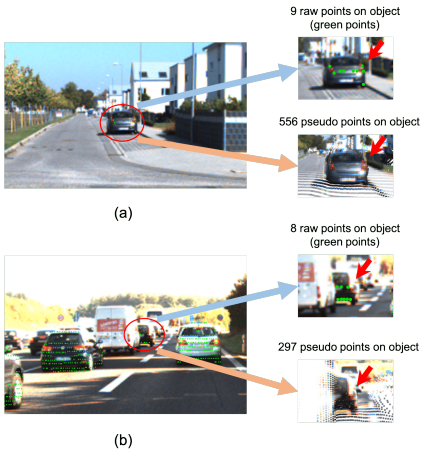

Figure 2: Comparison between raw LiDAR point clouds and pseudo point clouds.

Main Contributions ***Firstly***, we propose a new multi-modal framework named SFD (Sparse Fuse Dense), which leverages advantages of pseudo point clouds generated from depth completion to tackle the sparsity problem in LiDAR-only methods. Moreover, the dimension gap, information loss and data augmentation issues in multi-modal methods can be solved simultaneously. ***Secondly***, we design a new RoI feature fusion method *3D-GAF (3D Grid-wise Attentive Fusion)* to fully fuse raw LiDAR point clouds and pseudo point clouds. In order to further extract rich information in pseudo point clouds, we devise a *CPFE (Color Point Feature Extractor)*. In addition, we present a multi-modal data augmentation method *SynAugment (Synchronized Augmentation)*, which enables us to use data augmentation approaches designed for LiDAR-only methods. ***Finally***, we achieve the top performance on the KITTI 3D detection benchmark with our SFD.

2 Related Work

3D Detection Using Single-modal Data. Current 3d detection methods are mainly based on LiDAR data. SECOND Yan et al. (2018) proposes a sparse convolution operation to speed up 3D convolution. PV-RCNN Shi et al. (2020a) leverages the advantages of voxel-based methods and point-based methods to get more discriminative features. Voxel-RCNN Deng et al. (2020) points out that precise positioning of raw points is unnecessary. Recently, SE-SSD Zheng et al. (2021b) attains an excellent performance with self-ensembling.

3D Detection Using Multi-modal Data. Due to the sparsity of point clouds, researchers seek help from multi-modal methods which utilize both images and raw LiDAR point clouds. Many early methods use a cascading approach to use multi-modal data Qi et al. (2018); Wang & Jia (2019). More recent methods Liang et al. (2018); Xie et al. (2020); Huang et al. (2020) use LiDAR and image data in combination by establishing correspondence between point clouds and images and then indexing

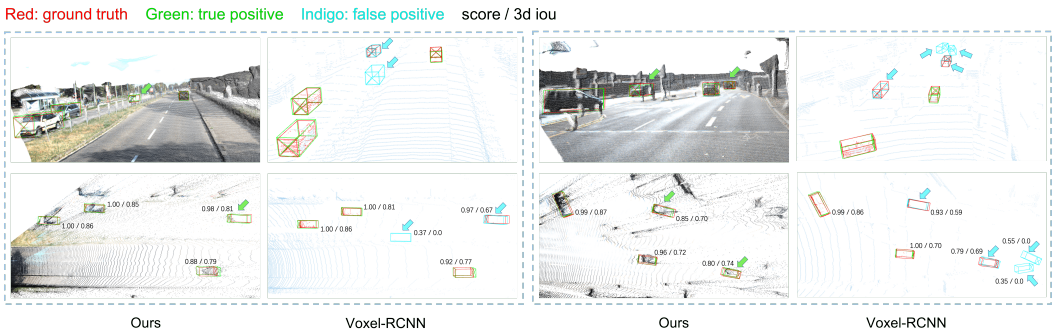

Figure 3: Comparison between SFD and Voxel-RCNN. For the visualization of SFD and Voxel-RCNN, we use pseudo clouds and raw clouds, respectively. We show true positives in green and false positives in indigo. The ground-truth boxes and raw LiDAR points inside the boxes are rendered in red. Green arrows represent that our predictions are more accurate, and indigo arrows represent false positives of Voxel-RCNN. The figures beside BEV boxes represent the score and 3d iou.

image features by point clouds. In other fields, there are some works Wang & Neumann (2018); He et al. (2021) also benefit from multi-modal data.

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 Hu et al. (2021); Imran et al. (2021); Gu et al. (2021). Although the primary purpose of the deep completion task is to serve downstream tasks, there are few methods using depth completion in 3D detection. In the image-based 3D object detection field, there are some works Wang et al. (2019); You et al. (2019) 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.

Related to MMF. MMF Liang et al. (2019) also utilizes depth completion and it benefits from multi-task multi-sensor fusion. In MMF, pseudo point clouds are used for point feature fusion, and depth completion feature maps are used for RoI feature fusion. In our method, we concentrate on finding a more effective RoI feature fusion method. When fusing RoI features, as shown in Figure 5(a), MMF concatenates reshaped *2D LiDAR RoI features* cropped from BEV LiDAR feature maps and *2D image RoI features* cropped from FOV image feature maps (i.e., depth completion feature maps). In contrast, as shown in Figure 5(b), our method fuses *3D raw LiDAR point clouds* and *3D pseudo point clouds*, which brings two benefits. ***Firstly***, 3D representation of images (pseudo point clouds) allows us to fuse RoI features from raw LiDAR point clouds and images in a more fine-grained manner (3D-GAF). We elaborate on three advantages of our 3D-GAF over previous RoI fusion methods (include MMF) in Sec. 3.3. ***Secondly***, with the 3D representation of images, our model can use all data augmentation approaches tailored to LiDAR-only methods (see our SynAugment in Sec. 3.5), while MMF cannot. Overall, our method makes fuller use of the advantages of pseudo point clouds than MMF, and pushes a new state-of-the-art.

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 $\mathcal{I}$ donate 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 $\mathcal{P}$. Moreover, 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).

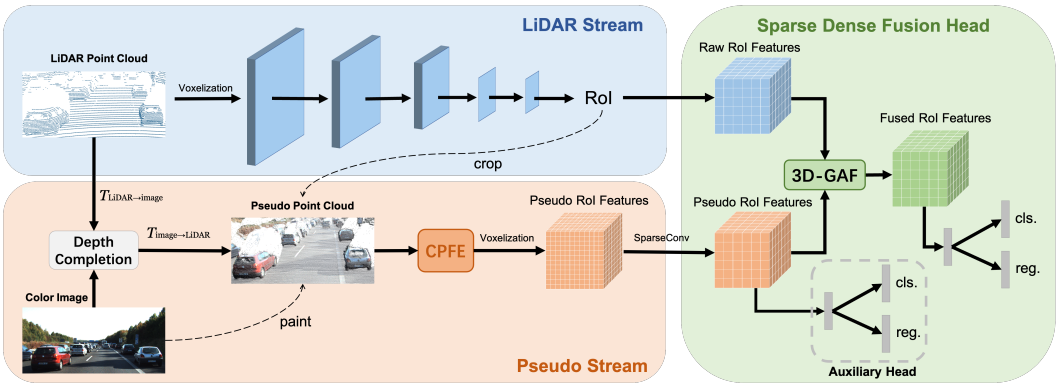

Figure 4: SFD mainly consists of three parts: *LiDAR Stream*, *Pseudo Stream* and *Sparse Dense Fusion Head*. (1)*LiDAR Stream* only uses raw clouds (generated by LiDAR) to predict 3D RoIs. Then RoIs are used to crop raw clouds and pseudo clouds (generated from depth completion). (2)*Pseudo Stream* uses raw clouds and images to generate pseudo clouds. Painting pseudo clouds with RGB, we get the colorful pseudo clouds. Then CPFE (see Figure 7) is 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 Figure 6), then the fused RoI features are used to predict class confidences and bounding boxes. In addition, an auxiliary head is employed to regularize our network. It can be detached at inference time.

3.2 Overview of Methods

We show our framework in Figure 4, 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 CPFE, 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, previous methods Chen et al. (2017); Ku et al. (2017); Liang et al. (2019), directly concatenate reshaped *2D LiDAR RoI features* cropped from BEV LiDAR feature maps and *2D image RoI feature* cropped from FOV image feature maps in a roi-wise way, which is coarse. In our method, 2D images are converted to 3D pseudo clouds, allowing us to fuse them in a more fine-grained manner, as shown in Figure 5. We propose a more effective RoI fusion named 3D-GAF, consisting 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 5(b). Previous methods use 2D RoI to crop image features, which involves features from other objects or backgrounds. It causes a lot of interference, especially for occluded objects, as shown in Figure 5(a). (2)***Grid-wise Fusion***. Because of the same representation of raw RoI features and pseudo RoI features, we can fuse each couple of grid features separately. It enables us to accurately enhance each part in an object with the corresponding pseudo grid feature instead of the whole pseudo RoI feature used in previous methods. (3)***Attentive Fusion***. We utilize attention to fuse each couple of grid features adaptively, as shown in Figure 6. For grids where raw clouds are sparse, pseudo features should be enhanced. For grids where pseudo clouds are inaccurate, pseudo features should be weakened. Table 7 provides ablation studies on 3D Fusion, Grid-wise Fusion and Attentive Fusion, validating their effectiveness.

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 ($6\times6\times6$ by default) is the total number of grids in a 3D RoI, and C is the grid feature channel. The i^{th} raw RoI grid feature and i^{th} pseudo RoI grid feature in $\mathbf{b}$ are denoted as F_i^{raw} and F_i^{pse}, respectively. As shown in Figure 6, given a couple of RoI grid features $(F_i^{\text{raw}}, F_i^{\text{pse}})$, we apply two fully connected layers on them and concatenate the outputs, getting compress feature

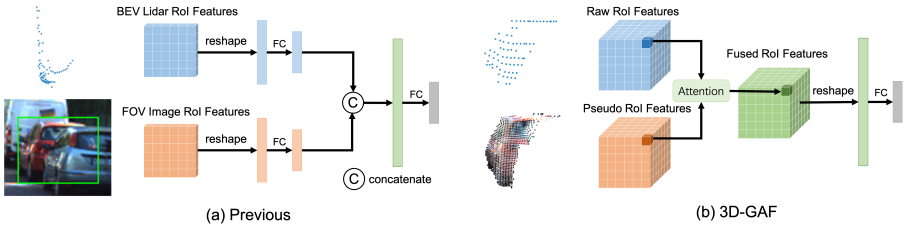

Figure 5: Comparison between previous methods and 3D-GAF.

embeddings E_i. Then a fully connected layer and a sigmoid operation are applied sequentially, resulting in the weight $w_i \in \mathbb{R}^2$, which consists of a weight $w_i^{\text{raw}} \in \mathbb{R}^1$ for the raw grid feature and a weight $w_i^{\text{pse}} \in \mathbb{R}^1$ for the pseudo grid feature:

$$E_i = \text{MLP}(\text{CONCAT}(F_i^{\text{raw}}, F_i^{\text{pse}})) \tag{1}$$

$$w_i = \sigma(\text{MLP}(E_i)) \tag{2}$$

In addition, we feed the couple of RoI grid features to two fully connected layers, resulting in a couple of transformed RoI grid features $T_i = (T_i^{\text{raw}}, T_i^{\text{pse}})$:

$$T_i^{\text{raw}} = \text{MLP}(F_i^{\text{raw}}), \quad T_i^{\text{pse}} = \text{MLP}(F_i^{\text{pse}}) \tag{3}$$

After that, we multiply T_i with weight w_i, getting a couple of attentive RoI grid features. Finally, they are concatenated and fed to a fully connected layer, resulting in a fused RoI grid feature F_i:

$$F_i = \text{MLP}(\text{CONCAT}(w_i^{\text{raw}} T_i^{\text{raw}}, w_i^{\text{pse}} T_i^{\text{pse}})) \tag{4}$$

In practice, all couples of RoI grid features in a batch can be processed in parallel, so our 3D-GAF is computationally efficient. There are some works Kaul et al. (2019); He et al. (2020b) that also use a dual fusion architecture, while the motivation and design of each work are different. We concatenate the pseudo grid feature and raw grid feature to produce a couple of weights for the couple of grid features, aiming to fuse raw clouds and pseudo clouds in different parts of an object adaptively.

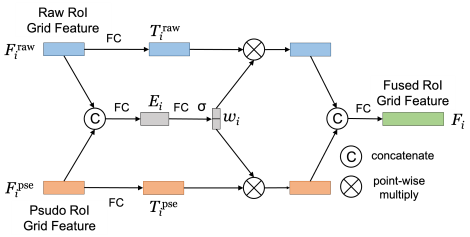

Figure 6: Illustration of Attentive Fusion.

3.4 Color Point Feature Extractor

A naive approach to exploit 3D-GAF is directly voxelizing pseudo clouds without any feature extraction on them, which obviously cannot fully explore rich information of pseudo clouds. Therefore, we propose to perform point-wise feature extraction on pseudo clouds before voxelization. PointNet++ Qi et al. (2017) 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 extract both 3D structural features and 2D semantic features efficiently.

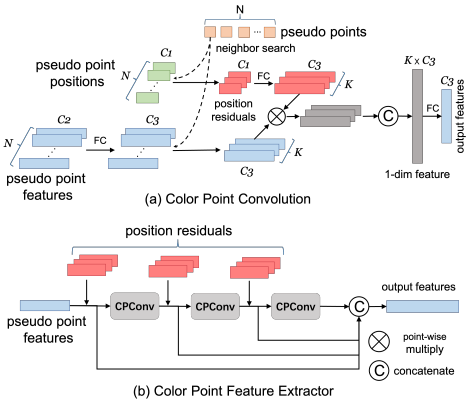

Figure 7: Illustration of CPConv and CPFE.

Color Point Convolution Based on the above two points, we propose a CPConv (Color Point Convolution), which searches neighbors on a regularly arranged image domain, as inspired by the voxel query Deng et al. (2020) and grid search Fan et al. (2021). 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. Specifically, we project pseudo points in a 3D RoI to an image according to their image coordinates (u, v). After projecting, each pseudo point can get its image neighbors easily. In addition, we search dilate neighbors instead of nearest neighbors for a larger receptive field, following Yu & Koltun (2015).

Pseudo Point Features For the i^{th} pseudo point p_i, we donate its feature as $f_i = (x_i, y_i, z_i, r_i, g_i, b_i)$, which consists of 3D geometric features and 2D semantic features. As motivated by Deng et al. (2020), we apply a fully connected layer on pseudo point features before performing the neighbor search to reduce complexity. Then the feature channel is raised to C_3, as shown in Figure 7(a).

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 $||\cdot||$ calculates the Euclidean distance between p_i and ${p_i}^k$.

Feature Aggregation For K ($K = 9$ by default) dilate 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 for alignment 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 Fan et al. (2021) instead of max-pooled Deng et al. (2020) for maximum information fidelity. Finally, a fully connected layer is applied to map aggregated feature channel back to C_3.

Multi-Level Feature Fusion To extract deeper features of pseudo clouds, we propose a Color Point Feature Extractor, which stacks three CPConvs, as illustrated in Figure 7(b). 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 features from multi-level to get a more comprehensive and discriminate feature representation for objects.

3.5 Synchronized Augmentation

With depth completion, 2D images can be converted into 3D pseudo clouds, allowing us to easily achieve all existing data augmentation approaches tailored to LiDAR-only methods. Specifically, we paint pseudo clouds with RGB before data augmentation. Then we only need to perform data augmentation on pseudo clouds synchronizing with raw clouds to achieve multi-modal data augmentation, as shown in Figure 8. We call this data augmentation method SynAugment.

MoCa Zhang et al. (2020) provides a common data augmentation pipeline for multi-modal methods by reversing point cloud transformations and replaying the image transformation. However, it is not suitable for our SFD. When using gt-sampling, MoCa needs to obtain image masks for all training samples in advance and perform additional occlusion detection on images, which is complicated. In addition, the feature mapping in MoCa relies on the sparse correspondence between images and point clouds, which introduces image information loss. By contrast, our method can use all image information in 3D RoIs.

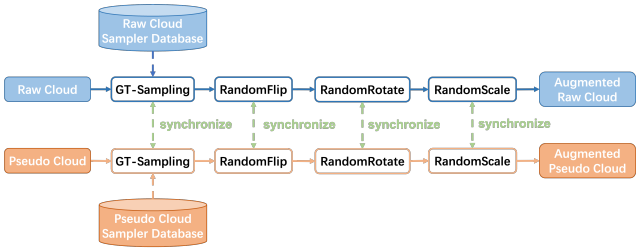

Figure 8: Illustration of SynAugment. We perform data augmentations on raw clouds and pseudo clouds synchronously.

3.6 LOSS FUNCTION

We follow the detection loss function of Deng et al. (2020), which is donated as $\mathcal{L}_{\text{det}}$. To prevent gradients from being dominated by *LiDAR Stream*, we add auxiliary loss $\mathcal{L}_{\text{aux}}$ on pseudo RoI features. $\mathcal{L}_{\text{aux}}$ is consistent with $\mathcal{L}_{\text{det}}$, including classification loss and regression loss. The depth completion network loss $\mathcal{L}_{\text{depth}}$ follows the definition of Hu et al. (2021). Then the total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda \mathcal{L}_{\text{aux}} + \beta \mathcal{L}_{\text{depth}} \tag{5}$$

where λ and β are the weight of $\mathcal{L}_{\text{aux}}$ and $\mathcal{L}_{\text{depth}}$ ($\lambda = 1$, $\beta = 1$ by default).

4 EXPERIMENTS

4.1 DATASET AND EVALUATION METRICS

We evaluate our model on the KITTI 3D and BEV object detection benchmark Geiger et al. (2013). 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. The average precision (AP) on *val* set obtained from the 40-point and 11-point precision-recall (PR) are all provided. The average precision (AP) on the *test* set is obtained from the 40-point precision-recall (PR). For the reason that Waymo and NuScenes datasets do not provide depth completion labels, we do not conduct experiments on these two datasets.

4.2 IMPLEMENTATION DETAILS

The *LiDAR Stream* of SFD is based on Deng et al. (2020), an excellent 3d detector that only uses raw clouds. For the depth completion, we use Hu et al. (2021), a precise and efficient depth completion network. Actually, our method is not sensitive to the depth completion network. SFD with TWISE Imran et al. (2021) can also achieve a comparable result. We replace Smooth-L1 Loss with 3D GIoU Loss Zhou et al. (2019) in the regression head. The inference speed of SFD is 10.2 HZ on an NVIDIA RTX 2080 Ti GPU. We follow the data augmentation approaches mentioned in Deng et al. (2020) and Zheng et al. (2021a). 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.

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

Method	Modality	AP_{3D}		
		Easy	Mod	Hard
SECOND Yan et al. (2018)	LiDAR	83.34	72.55	65.82
PointPillars Lang et al. (2019)	LiDAR	82.58	74.31	68.99
PointRCNN Shi et al. (2019)	LiDAR	86.96	75.64	70.70
Part-A^2 Shi et al. (2020b)	LiDAR	87.81	78.49	73.51
SA-SSD He et al. (2020a)	LiDAR	88.75	79.79	74.16
STD Yang et al. (2019)	LiDAR	87.95	79.71	75.09
CIA-SSD Zheng et al. (2021a)	LiDAR	89.59	80.28	72.87
PV-RCNN Shi et al. (2020a)	LiDAR	90.25	81.43	76.82
Voxel-RCNN Chen et al. (2019)	LiDAR	90.90	81.62	77.06
CT3D Sheng et al. (2021)	LiDAR	87.83	81.77	77.16
SE-SSD Zheng et al. (2021b)	LiDAR	**91.49**	82.54	77.15
MV3D Chen et al. (2017)	LiDAR+RGB	74.97	63.63	54.00
ContFuse Liang et al. (2018)	LiDAR+RGB	83.68	68.78	61.67
F-PointNet Qi et al. (2018)	LiDAR+RGB	82.19	69.79	60.59
AVOD Ku et al. (2017)	LiDAR+RGB	83.07	71.76	65.73
PI-RCNN Xie et al. (2020)	LiDAR+RGB	84.37	74.82	70.03
UberATG-MMF Liang et al. (2019)	LiDAR+RGB	88.40	77.43	70.22
EPNet Liang et al. (2019)	LiDAR+RGB	89.81	79.28	74.59
3D-CVF Yoo et al. (2020)	LiDAR+RGB	89.20	80.05	73.11
CLOCs PVCas Pang et al. (2020)	LiDAR+RGB	88.94	80.67	77.15
SFD (ours)	LiDAR+RGB	90.83	**83.96**	**77.47**

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 surpasses all state-of-the-art multi-modal methods by a large margin. For LiDAR-only methods, SFD outperforms previous best method SE-SSD Zheng et al. (2021b) by 1.42% on the moderate level. We also provide a comparison on the KITTI val set, as seen in Table 3, and our SFD also achieves a good performance. In addition, BEV detection results are provided in Table 2, and Figure 3 shows the prediction visualization of SFD and Voxel-RCNN.

Table 2: Comparison between Voxel-RCNN and SFD on the KITTI *val* set with BEV AP calculated by 40 recall positions for car class.

Method	AP_{BEV}		
	Easy	Moderate	Hard
Voxel-RCNN	95.52	91.25	88.99
SFD (ours)	**96.42**	**92.27**	**91.49**

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

Method	Modality	$3D_{R11}$			$3D_{R40}$		
		Easy	Mod	Hard	Easy	Mod	Hard
Fast PointRCNN Chen et al. (2019)	LiDAR	89.12	79.00	77.48	-	-	-
PV-RCNN Shi et al. (2020a)	LiDAR	89.35	83.69	78.70	92.57	84.83	82.69
Pyramid-PV Mao et al. (2021)	LiDAR	89.37	84.38	78.84	-	-	-
Voxel-RCNN Deng et al. (2020)	LiDAR	89.41	84.52	78.93	92.38	85.29	82.86
SE-SSD Zheng et al. (2021b)	LiDAR	-	85.71	-	93.19	86.12	83.31
UberATG-MMF Liang et al. (2019)	LiDAR+RGB	88.40	77.43	70.22	-	-	-
3D-CVF Yoo et al. (2020)	LiDAR+RGB	-	-	-	89.67	79.88	78.47
EPNet Huang et al. (2020)	LiDAR+RGB	-	-	-	92.28	82.59	80.14
CLOCs PVCas Pang et al. (2020)	LiDAR+RGB	-	-	-	92.78	85.94	83.25
SFD (ours)	LiDAR+RGB	**89.74**	**87.12**	**85.20**	**95.47**	**88.56**	**85.74**

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

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

Table 5: Ablation study on SynAugment. When SynAugment is not used, we remove gt-sampling, random scaling and random rotation. The results are evaluated with the AP calculated by 40 recall positions for car class.

Experiment	SynAugment	AP_{3D}		
		Easy	Moderate	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

4.4 ABLATION STUDY

Table 4 and Table 5 detail how each proposed module influences the accuracy of our SFD. Experiment (a) is our baseline which is modified on Voxel-RCNN Deng et al. (2020). It only uses raw clouds as input. Experiments (b),(c) and (d) are all equipped with multi-modal data augmentation (SynAugment) for a fair comparison with experiment (a), which is equipped with single-modal data augmentation.

Effect of 3D Grid-wise Attentive Fusion In Table 4, experiment (b) uses 3D-GAF to fuse raw RoI features and pseudo RoI features, making a 0.61%, 1.10% and 2.32% improvement on easy, moderate and hard levels, respectively. It demonstrates the effectiveness of our 3D-GAF.

Effect of Color Point Feature Extractor In Table 4, experiment (c) exploit our CPFE to extract rich features of pseudo clouds based on experiment (b), yielding a moderate AP of 88.56% with 1.99% improvement. It also demonstrates that with 3D-GAF and CPFE combined, our SFD outperforms the baseline by 3.09% on the moderate level.

Effect of Synchronized Augmentation Our SynAugment enables our multi-modal framework to utilize data augmentation approaches designed for LiDAR-only methods. We take off those data augmentation approaches designed for LiDAR-only methods from experiments (a) and (b) in Table 4, resulting in experiments (a) and (b) in Table 5. As shown in Table 5, without SynAugment, the

performance of our multi-modal method drops drastically, which proves the importance of sufficient data augmentation for multi-modal methods.

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 Shi et al. (2019), Part-A^2 Shi et al. (2020b) and SECOND Yan et al. (2018) implemented by OpenPCDet Team (2020). As shown in Table 6, our method can improve different detectors significantly. For the one-stage detector SECOND, we use the same architecture as *Pseudo Stream* (CPFE with sparse convolutions) to extract features of raw clouds in the 3D RoI. The raw clouds are also painted with RGB information to be consistent with pseudo clouds.

Table 6: Cooperating with different detectors. The average precisions are calculated by 40 recall positions for car class on easy, moderate and hard levels.

Methods	with SFD	AP_{3D}		
		Easy	Moderate	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*

Ablation Study on 3D Grid-wise Attentive Fusion We conduct experiments to verify the effectiveness of each part of 3D-GAF, as shown in Table 7. Experiment (a) directly concatenates raw RoI features and pseudo RoI features cropped by 2D RoI, which we call *2D RoI-wise Concat Fusion*. Experiment (b) concatenates raw RoI features and pseudo RoI features cropped by 3D RoI, which we call *3D RoI-wise Concat Fusion*. Experiment (c) fuses a couple of RoI features in a grid-wise manner based on experiment (b), and experiment (d) extends (c) with Attentive Fusion. Results show that each part of 3D-GAF can improve our SFD. Moreover, we can find that the contribution of Grid-wise Fusion and Attentive Fusion mainly lie on the moderate level and easy level, respectively.

Table 7: 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.

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**

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.

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.

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*

5 CONCLUSION

We propose a new multi-modal framework SFD for high quality 3D detection. With SFD, we overcome the dilemmas in LiDAR-only methods and multi-modal methods. We propose a new RoI fusion method 3D-GAF, which fuses raw clouds and pseudo clouds in a more fined-grained manner. To fully explore informative pseudo clouds, we design a CPFE, which efficiently and effectively extract both 3D features and 2D features in pseudo clouds. With SynAugment, our SFD can use all existing data augmentation approaches tailored to LiDAR-only methods. Experimental results on the KITTI dataset demonstrate that our approach can significantly improve detection accuracy and outperform other start-of-the-arts, including single-modal and multi-modal methods.

REFERENCES

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

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

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.

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.

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.

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.

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

Qingdong He, Zhengning Wang, Hao Zeng, Yijun Liu, Shuaicheng Liu, and Bing Zeng. Stereo rgb and deeper lidar based network for 3d object detection. *arXiv preprint arXiv:2006.05187*, 2020b.

Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3003–3013, 2021.

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.

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*, pp. 35–52. Springer, 2020.

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*, pp. 2583–2592, 2021.

Chaitanya Kaul, Nick Pears, and Suresh Manandhar. Sawnet: A spatially aware deep neural network for 3d point cloud processing. *arXiv preprint arXiv:1905.07650*, 2019.

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

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*, pp. 12697–12705, 2019.

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

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*, pp. 7345–7353, 2019.

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.

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

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*, pp. 5099–5108, 2017.

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*, pp. 918–927, 2018.

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.

Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *CVPR*, pp. 770–779, 2019.

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*, pp. 10529–10538, 2020a.

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*, 2020b.

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

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*, pp. 4604–4612, 2020.

Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 135–150, 2018.

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*, pp. 8445–8453, 2019.

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.

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*, pp. 12460–12467, 2020.

D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

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

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.

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.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

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

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, pp. 9267–9274, 2019.

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*, 2021a.

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*, pp. 14494–14503, 2021b.

Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pp. 85–94. IEEE, 2019.

Appendix

A More Ablation Study

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 9, SFD can consistently improve the state-of-the-art Voxel-RCNN Deng et al. (2020) on all classes and all evaluation metrics.

Table 9: Performance of SFD on the KITTI *val* set with AP calculated by 40 recall positions.

Class	with SFD	AP_{3D}			AP_{BEV}		
		Easy	Moderate	Hard	Easy	Moderate	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*

Robustness Analysis on Depth Completion Network PENet Hu et al. (2021) and TWISE Imran et al. (2021) are recently proposed deep completion networks. We train our SFD with them and evaluate the results on both the KITTI *val set* and *test set*. Although TWISE performs worse than PENet on the KITTI depth completion benchmark, SFD is not sensitive to this, as shown in Table 10. The performances of SFD with different depth completion networks are comparable, which manifests the robustness of our method.

Table 10: Robustness experiment with different depth completion networks. The results evaluated on the KITTI *val* set and *test* set are all provided, with AP calculated by 40 recall positions for car class.

Evaluation Set	PENet	TWISE	AP_{3D}			AP_{BEV}		
			Easy	Moderate	Hard	Easy	Moderate	Hard
val set	√		**95.47**	88.56	85.74	**96.42**	**92.27**	91.49
		√	95.41	**88.58**	**86.01**	96.28	92.25	**91.69**
test set	√		**90.83**	**83.96**	**77.47**	94.76	91.04	**86.31**
		√	90.62	83.60	77.10	**94.84**	**91.23**	86.26

Comparison between PointNet++ and CPConv Because of the huge number of pseudo points (see Figure 1), it is impossible to perform PointNet++ Qi et al. (2017) on all pseudo points, and it is inevitable to down-sample pseudo clouds. In our experiments, we sample 1024 pseudo points in each 3D RoI. As shown in Table 11, with more inference time, PointNet++ performs much worse than CPConv. We summarize the reasons as the following two points. *Firstly*, PointNet++ cannot make use of 2D semantic features in pseudo clouds because of the ball query. *Secondly*, down-sampling used by PointNet++ causes a lot of information loss, while in our CPConv, we can keep all pseudo points in 3D RoIs thanks to the fast neighbor search.

Table 11: Comparison between PointNet++ and CPConv. The results are evaluated with the average precision calculated by 40 recall positions for the car class.

Method	Inference Time	AP_{3D}		
		Easy	Moderate	Hard
PointNet++	95ms	92.25	85.61	83.36
CPConv(Ours)	**12ms**	**95.47**	**88.56**	**85.74**

Inference Speed We test the inference speed of SFD on an NVIDIA RTX 2080 Ti GPU with batch size 1. Our SFD runs at 15.2 HZ, excluding the latency of the depth completion network. With Imran et al. (2021) or Hu et al. (2021) as our depth completion network, the speed of SFD is 11.4 HZ or 10.2 HZ, respectively.

B MORE QUALITATIVE ANALYSES

Figure 9: Comparison between SFD and Voxel-RCNN. For the visualization of SFD and Voxel-RCNN, we use pseudo clouds and raw clouds, respectively. We show true positives, false positives and ground-truth boxes in green, indigo and red, respectively. The raw LiDAR points inside prediction boxes are rendered in blue. Green arrows represent that our predictions are more accurate, and indigo arrows represent false positives of Voxel-RCNN.

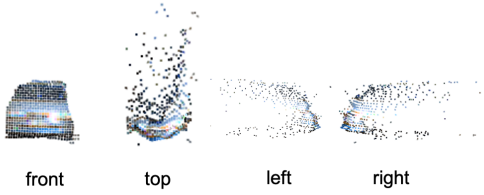

Figure 10: Different views of pseudo clouds on object ③ in Figure 9(b).

Figure 9 provides three cases, corresponding to three situations where SFD improves Voxel-RCNN Deng et al. (2020).

Occlusion Occlusion is a challenging problem in the scenario of autonomous driving, as shown in Figure 9(a). Object ① is heavily occluded by the black car in front, making its raw clouds insufficient (see ②). However, pseudo clouds can alleviate this by providing sufficient 3D geometric information and additional 2D semantic information.

Long Distance Figure 9(b) shows another common scene. Due to the limited resolution of LiDAR, faraway objects are with much fewer points. As shown in object ④, it is difficult to predict a precise box with insufficient raw clouds. However, pseudo clouds on the object are richer. Figure 10 shows different views of pseudo clouds on ③, demonstrating the rationality of our hypothesis. The pseudo clouds are qualified to provide supplementary information for raw clouds.

Background Similar to Foreground Dense pseudo clouds not only benefit location of foreground, but also help to distinguish background from foreground, as seen in Figure 9(c). In the autonomous driving scene, some background raw clouds are similar to the foreground due to the sparsity of raw clouds, which may confuse detectors and cause many false positives. In Figure 9(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 method to distinguish them.