Structure Guided Proposal Completion for 3D Object Detection

Chao Shi[1], Chongyang Zhang[2], and Yan Luo[1]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2] Shanghai Key Laboratory of Digital Media Processing and Transmission, Shanghai 200240, China
{shichaostone,sunny_zhang,luoyan_bb}@sjtu.edu.cn

Abstract. 3D object detection from point clouds is one of the key components in autonomous driving. Current two-stage detectors generate a small number of proposals, and then refine them in the second RCNN procedure. However, due to the inherent sparsity of point clouds, the first stage may predict some low quality proposals with incomplete structure and inaccurate localization. These low quality proposals fail to obtain adequate and precise proposal features which are essential for the following refinement, inevitably degrading the overall detection performance. To alleviate this problem, we propose Structure guided Proposal Completion (SPC) for 3D object detection from point clouds. Specifically, two completion strategies are developed to obtain high quality proposals: one is Structure Completion, in which a group of structural proposals are obtained by traversing most structures, and thus at least one proposal with ground truth similar structure can be guaranteed. The other is RoI Feature Completion, which is used to fill the empty area of proposals with virtual points under structure-aware manner. With the proposed SPC, high quality proposals with clearer structure and more precise localization can be obtained, and further promote the RCNN to perceive adequate proposal features. Extensive experiments on KITTI benchmark demonstrate the effectiveness of our proposed method, especially for hard setting objects with fewer LiDAR points.

Keywords: 3D Object Detection · Point Cloud · Proposal Completion.

1 Introduction

3D object detection is one of the core computer vision tasks, since it benefits wide applications in various fields, such as autonomous driving, virtual reality and robot perception [1,5]. Point clouds from LiDAR sensors are often adopted to detect objects from 3D space for its less sensitivity to weather and time of the day. Hence, point clouds based 3D detectors [3,6,20,21,29,30] are one of the main research topics in both academic and industrial area.

Despite the great success of the above methods, their performance is still limited by the sparsity of point clouds. It means that the LiDAR sensors can solely

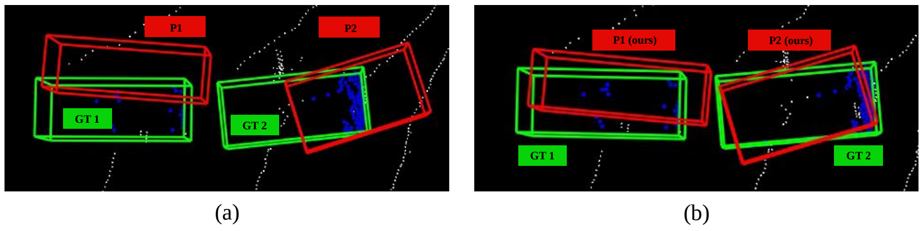

Fig. 1. Examples of RPN predicted proposals from sparse point clouds: (a) proposals from the first stage of PointRCNN [22], denoted as P1 and P2; and (b) augmented proposals by our Structure Completion module, denoted as P1(ours) and P2(ours). The object points, ground-truth boxes, and the predicted proposals are shown in blue, green and red colors, respectively. After applying Structure Completion, we can obtain a set of augmented proposals and we only show the most precisely localized proposals for better view.

capture small portion of object point clouds, especially in distant or occluded regions, making it difficult to generate precise 3D bounding boxes. Concretely, we analyzed the statistical results of point clouds in KITTI dataset [4], and found that: more than 20.2% objects have less than 30 points, including 10.8% objects have less than 10 points, and even worse for pedestrian and cyclist in KITTI dataset. Most of them are incomplete with some important parts missing, such as tyre of a bicycle or arm of a man.

To alleviate the sparsity of point clouds, some methods [2,24,26] combine the data of multi sensors with the usage of complementary information. MV3D [2] directly merges the image features and birds-eye-view (BEV) features. PointPainting [24] first paints points with semantic segmentation score, then uses the painted points to point-based 3D detectors. However, these approaches are complex with multiple fusion modules. Pyramid R-CNN [15] designs RoI-grid Pyramid to gather more points of interest outside proposals for accurate object recognition, but the background points may also be gathered. ImpDet [19] uses evenly distributed virtual points to learn richer context features, but ignores the structural information of each object. In addition to the widely studied issues above, we have further discovered that the sparsity also brings the difficulty to localize object accurately. As shown in Figure 1(a), region proposal network (RPN) may yield low quality proposals with inaccurate localization due to the sparsity of points, which brings difficulties for further regression. Moreover, constrained by sparse points and features, it is hard for the second refinement stage [22,30] to generate adequate proposal features.

To overcome the above limitations, we propose a novel two-stage 3D detection framework, namely **S**tructure guided **P**roposal **C**ompletion (SPC). To mitigate the problem of low quality proposals, we innovatively add a **Structure Completion** module after RPN, to augment the proposals containing insufficient points and features in structure guided manner. This module is inspired by the observation that for objects with fewer points, it is difficult for the network to predict where to regress and which part of the object these points belong to due to the insufficient context information. To this end, based on the predicted proposals and object structure, we traverse the missing structure of proposals with limited points and obtain a set of augmented proposals, in which at least one accurately localized proposal can be guaranteed for second refinement stage. As shown in Figure 1(b), after Structure Completion, our model can yield high quality proposals with precise localization. Subsequentially, we propose a **RoI Feature Completion** module, aiming to obtain more adequate proposal features by virtual points under structure-aware manner. Firstly, we use 2D CNN to extract BEV feature from pseudo BEV map. Then we fill the empty area in each proposal with virtual points which are evenly distributed in the proposal. The proposal features are obtained by adaptively aggregating the virtual point features and raw point features with attention mechanism. Combining all the above components, our approach can accurately detect objects with a few points and effectively improve the performance of 3D object detection. Eventually, for the 3D detection under challenging settings, our method presents outstanding performance compared with the state-of-the-art methods.

We summarize our contributions as follows: (1) We propose a Structure guided Proposal Completion 3D object detector (SPC), which mitigates the low quality proposal and inadequate proposal feature problems caused by the sparsity issue. (2) We propose two simple yet effective completion strategies. One is Structure Completion, by traversing the missing structures of object to obtain accurately localized proposal, which can be served as a plug-in to enhance point-based 3D detection models. The other is RoI Feature Completion, by filling the empty area of proposals with virtual points to generate adequate proposal features. (3) We conduct extensive experiments on KITTI dateset and demonstrate the effectiveness of our approach, especially for hard setting objects.

2 Related Work

In generally, LiDAR-based 3D object detectors can be categorized into two streams: (i) single-stage detectors predict object bounding boxes and scores directly in one stage, which usually run effectively due to simpler network structures; and (ii) two-stage detectors usually generate some coarse proposals in the first stage, then these proposals and corresponding features are fed into second stage for refinement, which help detectors attain higher precision.

2.1 Single-stage Object Detectors.

VoxelNet [35] first encodes point clouds as voxels, then proposes voxel feature encoding layer to extract voxel-wise feature. But it is computationally expensive due to the 3D convolution operation. SECOND [28] adopts the sparse convolution to accelerate voxel feature extraction process. PointPillar [8] collapses the points in vertical columns (pillars) instead of voxels for effective encoding process, then uses pseudo-image for feature learning and object detection. 3D-SSD [29] introduces a novel sampling strategy named Feature-FPS for better classification by combining feature-based and point-based sampling distances. SA-SSD [6] proposes an auxiliary network and losses on the basis of 3D voxel CNN to preserve structure information. SE-SSD [34] utilizes teacher SSD and student SSD to get more training data, meanwhile it also consumes more time to train the model.

2.2 Two-stage Object Detectors.

PointRCNN [22] first uses PointNet [17] to segment foreground objects and generate 3D proposals, then refines them with semantic features. Based on PointRCNN, Part-A^2 [23] introduces an intra-object part supervision to improve the feature representation. STD [30] proposes PointsPool operation for RoI refinement, which converts sparse feature to dense feature representation. PV-RCNN [21] combines point-based and voxel-based network to extract features from keypoints and voxels, then aggregates them by RoI-grid pooling. Voxel R-CNN [3] proposes Voxel RoI Pooling to extract RoI feature from voxels for refinement. Pyramid R-CNN [15] alleviates the sparsity and imbalanced distribution problems of points by RoI-grid pyramid and density-aware radius prediction. CT3D [20] uses channel-wise attention to reweight the proposal features for refinement. SFD [25] is multi-modality method with image and LiDAR as inputs for 3D detection. Different from the previous methods, we add a Structure Completion after RPN to generate more accurately localized proposals.

2.3 Point Cloud Augmentation.

Point cloud augmentation aims to generate denser points representation from sparse LiDAR points or RGB images. LiDAR-based methods like PUNet [32] reconstructs multiple upsampled points from high level feature vectors. Image-based methods [11,31] first perform depth completion and then convert to point clouds. MVP [31] predicts object depth in image space and further generates dense points. ImpDet [19] uniformly places virtual points around candidate point, then randomly chooses some points for boundary generation. In contrast, we generate virtual points in structure-aware manner and obtain their features from BEV feature, which is more accurate for refinement stage.

3 Methodology

The overall framework of Structure guided Proposal Completion (SPC) is shown in Figure 2. Given the input raw points, the first RPN predicts a number of proposals, and the following Structure Completion augments the sampled proposals by traversing most structures. We further introduce the RoI Feature Completion module for more adequate proposal features generation, which will be fed into the final detection head. Details are shown in the following sections.

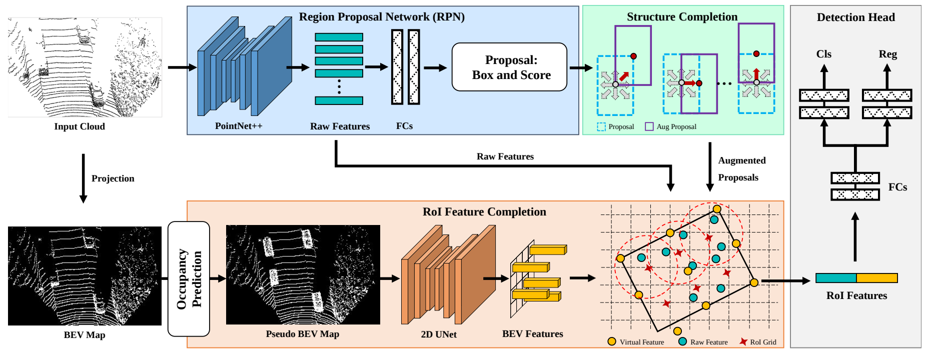

Fig. 2. Framework overview. We first generate 3D proposals from raw points with RPN, then a set of more accurately localized proposals will be generated by Structure Completion module. Subsequently, RoI Feature Completion module is introduced to generate more adequate proposal features by attentively aggregating raw point features and virtual point features. The final one is detection head for box refinement.

3.1 RPN for Proposals Generation

For each 3D scene, let $\mathcal{P} = \{(x, y, z, r)_n, n = 1, \ldots, N\}$ be a set of raw point clouds, where (x_n, y_n, z_n) means 3D location in LiDAR coordinate system and r_n means the reflectance. RPN takes $\mathcal{P}$ as input and generates a set of 3D bounding boxes $\mathcal{B} = \{B_1, B_2, \cdots, B_K\} \in \mathbb{R}^{K\times 8}$ that represent the detected objects, where K denotes the number of objects in each scene. Each 3D bounding box B_k is represented as $(x_k, y_k, z_k, h_k, w_k, l_k, \theta_k, c_k)$, where (x_k, y_k, z_k) is object center, (h_k, w_k, l_k) is object size, θ_k is object orientation, and c_k is classification score. In this paper, we choose the first stage of PointRCNN [22] as our default RPN to generate 3D proposals due to its effectiveness and accuracy. It is worth noting that PointRCNN can be replaced by other high quality RPN. During the first stage, RPN outputs a set of 3D proposals and extracts corresponding point features fed to the next stage.

3.2 Structure Completion Module

Due to distance and different forms of occlusion, the number of point clouds in certain objects may be quite limited. When further integrated with the indispensable point sampling strategy, some essential foreground points are discarded, which makes the sparsity issue worse. Hence, it is difficult for RPN to predict where to regress and which portion these points belong to, resulting in low quality proposal with inaccurate localization. This motivates us to generate precisely localized proposals. Accordingly, we propose **Structure Completion** module to augment the existing predicted proposals.

Preliminary. For each object, we divide the captured points into four parts (front-left, front-right, behind-left and behind-right, respectively) in 3D space.

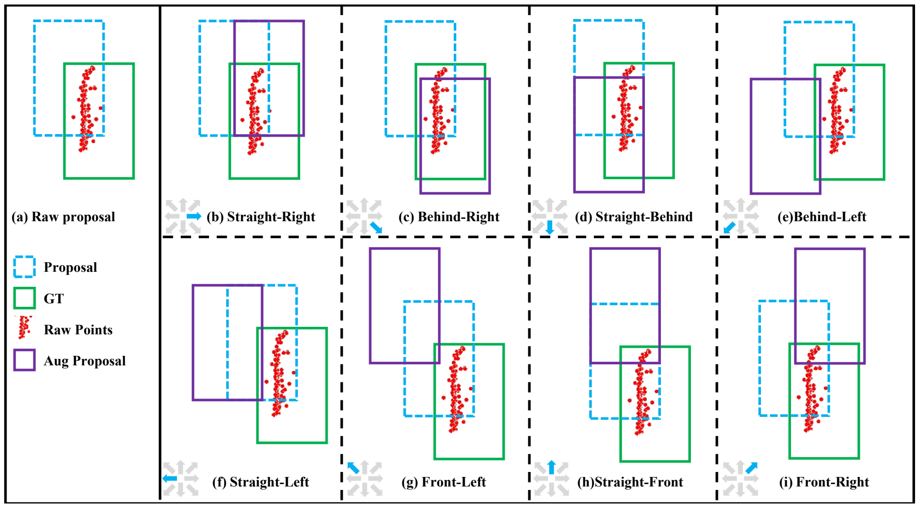

Fig. 3. The illustration of Structure Completion module. We show the captured points, ground truth, predicted proposal and augmented proposals in red, green, blue and purple, respectively. (a) shows the captured points, ground truth box and raw proposal. (b)-(i) represent the Structure Completion process from raw proposal to augmented proposals according to the missing structures. Specifically, we take proposal center as origin and shift the proposal by $l/2$ or $w/2$ in eight directions. For example, we shift the raw proposal by $w/2$ in straight-right direction and obtain the augmented proposal as shown in the (b). Particularly, we can obtain at least one more precisely localized proposal after Structure Completion, as shown in the (c).

The origin is located at the bottom center of the object, and the front direction is same with the heading direction. Inspired by [14], we find that the low quality proposals are mainly caused by localization error in 3D space, so our Structure Completion module focuses on improving the accuracy of localization prediction.

Structure Completion. Since proposals with fewer points suffer from structure missing and inaccurate localization, our mechanism pays more attention on these proposals, denoted as $\mathcal{B}_s = \{B_1, B_2, \cdots, B_s\}$. In details, given the full-set proposals generated by RPN, we first count the number of points in each proposal, and then sample the proposals with small number of points for subsequent completion. Considering that large objects like cars naturally have more points than small objects like pedestrians and cyclists, it is unfair to directly select proposals with the smaller number of points among all proposals. So we sample the proposals with fewer points in the same class objects instead.

As shown in the Figure 3(a), LiDAR captures points belonging to front-left part of the object, denoted by red points, and ground truth is drawn in green box. We show them in BEV map for better visualization. The subsequent down-sampling operation will worsen this problem and make their structure incom-

Algorithm 1 Pseudocode of Structure Completion Module

```
Input:
    proposals $\mathcal{B} \subseteq \mathbb{R}^{K\times 8}$ , $B_k = (x_k, y_k, z_k, h_k, w_k, l_k, \theta_k, c_k)$
    $H$ points number threshold
    $L$ number of augmented proposals
Output:
    augmented proposals $\mathcal{B}_H \subseteq \mathbb{R}^{(H\times L)\times 8}$

def structure_completion(proposals, $H$, $L$):
    # sample proposals based on points number threshold
    samples=proposals[: $H$, :]
    template=array(([1,1], [1,-1], [-1,-1], [-1,1], [1,0], [0,-1], [-1,0], [0,1]))
    center=samples[:, 3:5] $\times$ template
    # transform from local coordinate system to LiDAR coordinate system
    transform_center=transform(center, samples[:, 6]) + samples[:, 0:2]
    for i in range($L$):
        samples[:, 0] = transform_center[i, 0]
        samples[:, 1] = transform_center[i, 1]
    return samples
```

plete, then RPN may make a wrong prediction with inaccurate localization. It means the predicted proposal may be distributed around the object with relative small IoU. For example, RPN incorrectly predicts these points as behind-right structure and generates a proposal on the front-left direction, as shown in the Figure 3(a), which is hard to refine. In order to obtain accurately localized proposals, we propose to augment these predictions by traversing object structure. Concretely, for each selected proposal B_s, we take the predicted center (x, y, z) as the origin, then shift the proposal by $w/2$, $l/2$, $(w/2, l/2)$ in the eight different directions, including straight-right, behind-right, straight-behind, behind-left, straight-left, front-left, straight-front and front-right directions, respectively, as shown in the Figure 3 (b)-(i). All structures of the object, *i.e.*, the front-right, behind-left or behind-right structure, can be included through this method. Finally, for each selected proposal, we will get eight new proposals. Since we have traversed possible missing structures, at least one accurately localized proposal can be obtained for refinement stage, as shown in the Figure 3(c).

The pseudocode of Structure Completion module is presented in Algorithm 1. Given $\mathcal{B} = \{B_1, B_2, \cdots, B_K\}$ proposals generated from the first stage, we first sample $\mathcal{B}_s$ proposals based on the points number threshold H. For each sampled proposal, we will generate L new augmented proposals. Finally, we will obtain $\mathcal{B}_{aug} = \{K + L \times H\}$ proposals. By default, we set $L = 8$. Despite its simplicity, this module makes a significant contribution and we will conduct ablation experiments in the Experiments Section to analyze the effects.

3.3 RoI Feature Completion Module

On top of high quality proposals generated by our Structure Completion, we further attempt to refine these proposals including localization and heading. However, previous methods [10,22] may fail to obtain complete proposal features because certain proposals have fewer points and features. This motivates us to generate adequate proposal feature for each proposal. Specifically, we propose RoI Feature Completion module, which consists of BEV feature learning, virtual points generation, and feature aggregation.

BEV Feature Learning. BEV image has several advantages compared to front view image. Such as, objects preserve origin physical size and no occlusion, which make BEV image more feasible in 3D object detection, especially for localization prediction. So we utilize BEV features for further virtual point features generation. Concretely, by projecting the points, we can obtain BEV image $H \times W \times C$, in which H and W are height and width of BEV image. However, using 2D CNN to extract raw BEV features directly may be inappropriate due to the sparsity, which makes objects on BEV image incomplete. So we first use the occupancy prediction [27] to generate more complete pseudo BEV map, then use 2D CNN to extract BEV features f^{bev}.

Virtual Points Generation. For each generated 3D proposal B_k, we enlarge its width and length to contain more points and features. Nevertheless, as distance increases, points become sparser and fewer points can be captured. To this end, we propose structure-aware virtual points generation module, which generates uniformly distributed virtual points to enrich original proposal points and features. This module is inspired by the observation that 3D objects have relatively fixed structures and resemble shape prototypes.

Concretely, we first generate one type of template from original dataset for each category, and then uniformly place M virtual points in each template, denoted as $\mathcal{V} = \{(x, y, z, r)_m, m = 1, \ldots, M\}$. These points encode the structural information of the object. Then we select corresponding template and place them in the canonical coordinate system. The heading and the center of the template remain the same as the original proposal. We project $\mathcal{V}$ to BEV map and apply bilinear interpolation to get virtual points feature f_k^v from BEV feature f^{bev} due to its larger receptive fields. After that, for each proposal B_k, we have raw points p_k and corresponding features f_k^r, and we also have virtual points v_k and corresponding features f_k^v. Hence, the original proposal features f_k^p are enriched by concatenating virtual point features.

$$f_k^p = [f_k^r; f_k^v], \text{for } k = 1, 2, \cdots, K \tag{1}$$

where $[*; *]$ denotes the concatenation operation.

Feature Aggregation. After virtual points generation, original proposal feature is enriched by virtual points. However, the proposal region may contain background points due to inaccurate localization. Intuitively, it is inappropriate to utilize foreground and background points equally, because foreground points and virtual points indicated object boundary should make more contributions to the refinement stage, while background points should contribute less. To adaptively aggregate raw point features and virtual point features, we adopt the attention-based aggregation module to re-weight above features. In practice, we apply point-wise attention and channel-wise attention to strengthen the features, respectively. The point-wise attention can be represented as:

$$R_k = W_2 \delta \left(W_1 e_k\right), e_k = \mathrm{POOL}\left(f_k^p\right) \tag{2}$$

where e_k represents the pooled features across the channel-wise dimensions, POOL represents the average-pooling, W_1, W_2 are the weight parameters of two fully-connected layers, and δ is the ReLU activation function. Similar to the point-wise attention, the channel-wise attention can also be computed and we will get $S_k = W_2' \delta \left(W_1' \left(g^k\right)^T\right)$, where g^k represents the pooled features across the point-wise dimensions.

By element-wise multiply, we obtain the attention matrix $A_k = \sigma\left(R_k \bigodot S_k\right)$, where σ is sigmoid function and $\bigodot$ is element-wise multiply operation. After that, the re-weighting features can be formulated as:

$$\widetilde{f_k^p} = f_k^p A_k \tag{3}$$

Eventually, the re-weighting features are fed into the network [18] to obtain proposal features for the final confidence classification and box refinement.

3.4 Detection Head and Loss Function

Detection Head. The detection head takes proposal features as input for refinement stage. Specifically, we first transform the proposal features into feature vectors by a shared two layers MLP. Then, the feature vectors are fed into two branches for bounding box regression and confidence prediction. In practice, our detection head adopts the same architecture as that in [22] with several fully connected layers.

Loss Function. We use multi-task loss to train our model. Specifically, the total loss is composed of region proposal loss $\mathcal{L}_{rpn}$ and refinement loss $\mathcal{L}_{rcnn}$ as

$$\mathcal{L}_{total} = \mathcal{L}_{rpn} + \mathcal{L}_{rcnn} \tag{4}$$

For region proposal loss $\mathcal{L}_{rpn}$, we adopt the same loss function with [22] as

$$\mathcal{L}_{rpn} = \mathcal{L}_{focal} + \mathcal{L}_{reg} \tag{5}$$

where classification loss $\mathcal{L}_{focal}$ are focal loss and we keep the default parameters $\alpha_t = 0.25$ and $\gamma = 2$, and smooth-L1 loss is utilized for regression loss $\mathcal{L}_{reg}$.

The refinement loss $\mathcal{L}_{rcnn}$ consists of two parts, including the smooth L1 loss for box refinement $\mathcal{L}_{box}$, and the multi-class cross entropy loss for box class prediction $\mathcal{L}_{ce}$,

$$\mathcal{L}_{rcnn} = \mathcal{L}_{box} + \mathcal{L}_{ce} \tag{6}$$

Table 1. Comparison with state-of-the-art methods on KITTI *val* set, the results are evaluated by the 3D Average Precision with 40 recall positions. The 3D APs with 11 recall positions are also reported under moderate setting. "-" means that results are not reported in the published version. Best in bold.

Type	Method	Modality	Car 3D AP_{R40}			Ped. 3D AP_{R40}			Cyc. 3D AP_{R40}			3D AP_{R11}		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Car	Ped.	Cyc.
1-stage	SECOND [28]	L	90.97	79.94	77.09	58.01	51.88	47.05	78.50	56.74	52.83	76.48	52.98	67.15
	PointPillars [8]	L	87.75	78.35	75.18	57.30	51.41	46.87	81.57	62.94	58.98	77.28	52.29	62.68
	SA-SSD [6]	L	92.23	84.30	81.36	-	-	-	-	-	-	79.91	-	-
2-stage	PointRCNN [22]	L	89.47	80.32	77.92	62.82	53.66	47.01	91.62	71.34	66.77	78.61	53.85	71.62
	PV-RCNN [21]	L	92.57	84.43	82.69	64.26	56.67	**51.91**	88.88	71.95	66.79	83.24	57.37	69.48
	Voxel R-CNN [3]	L	92.38	85.29	82.86	-	-	-	-	-	-	84.52	-	-
	Ours(SPC)	L	**92.71**	**85.75**	**83.35**	**66.18**	**58.92**	51.38	**93.66**	**75.55**	**70.97**	**85.85**	**59.63**	**74.72**
	Improvement	-	*+0.14*	*+0.46*	*+0.49*	*+1.92*	*+2.25*	*-0.53*	*+2.04*	*+3.60*	*+4.18*	*+1.33*	*+2.26*	*+3.10*

4 Experiments

We evaluate the effectiveness of our proposed method on the KITTI dataset [4], which consists of 7481 training samples and 7518 testing samples in 3D object detection task. The training data is divided into a *train* set with 3712 samples and a *val* set with 3769 samples [2]. We report our results on *val* set and *test* set for *Car*, *Pedestrian* and *Cyclist* categories. The 3D Average Precision (3D AP) is utilized as the evaluation metric. We adopt the official evaluation protocol for fair comparisons. Specifically, the IoU threshold is set to 0.7 for *Car* and 0.5 for *Pedestrian* and *Cyclist*. APs are computed by recalling 11 and 40 positions on the *val* and *test* splits respectively.

4.1 Experimental Setup

Network Architecture. We randomly choose 16384 points from the entire point clouds per scene, and the detection range is limited to [0,70.4]m for the x axis, [-40,40]m for the y axis and [-3,1]m for the z axis. We adopt PointR-CNN [22] as our default RPN due to its efficiency. And we use 2D UNet [9] to extract pseudo BEV map features. For the virtual points generation strategy, we empirically generate 27 virtual points for each proposals, and the corresponding virtual point feature dimension is 128. For the box refinement module in SPC, we follow the usual RoI pooling strategies in PointRCNN [22].

Training and Inference Details. The SPC model is end-to-end optimized for 80 epochs with the ADAM [7] optimizer with the batch size 4. The learning rate is initialized as 0.01 and is decayed 10x at 50 and 70 epochs. We conduct data augmentation at training stage following strategies in [3], including gt-sampling, random rotation, random flipping and random scaling.

4.2 Comparison with State-of-the-Arts

We compare our model with the state-of-the-art competitors on *Car*, *Pedestrian*, *Cyclist* using AP under 40 recall positions (AP_{R40}), as shown in Table 1. The

Table 2. Comparison with state-of-the-art methods on the KITTI *test* set, with average precision of 40 recall positions evaluated on the KITTI server. "-" means that the results are not reported in the published version. "L" means LiDAR-only methods, and "L+R" means the detector makes the use of both LiDAR and RGB modality. Best in bold.

Type	Method	Modality	**Car 3D** AP_{R40}			**Ped. 3D** AP_{R40}			**Cyc. 3D** AP_{R40}		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
1-stage	VoxelNet [35]	L	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
	ContFuse [12]	L+R	83.68	68.78	61.67	-	-	-	-	-	-
	SECOND [28]	L	84.65	75.96	68.71	45.31	35.52	33.14	75.83	60.82	53.67
	PointPillars [8]	L	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
	TANet [13]	L	84.39	75.94	68.82	**53.72**	**44.34**	**40.49**	75.70	59.44	52.53
	SA-SSD [6]	L	88.75	79.79	74.16	-	-	-	-	-	-
	CIA-SSD [33]	L	89.59	80.28	72.87	-	-	-	-	-	-
2-stage	MV3D [2]	L+R	74.97	63.63	54.00	-	-	-	-	-	-
	F-PointNet [16]	L+R	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
	PointRCNN [22]	L	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
	STD [30]	L	87.95	79.71	75.09	53.29	42.47	38.35	78.69	61.59	55.30
	MMF [11]	L+R	88.40	77.43	70.22	-	-	-	-	-	-
	PV-RCNN [21]	L	**90.25**	81.43	76.82	52.17	43.29	40.29	78.60	**63.71**	**57.65**
	PI-RCNN [26]	L+R	84.37	74.82	70.03	-	-	-	-	-	-
	CT3D [20]	L	87.83	81.77	77.16	-	-	-	-	-	-
	Ours	L	87.69	**81.80**	**77.22**	48.37	39.88	37.21	**80.66**	63.04	56.88

AP_{R40} of SA-SSD [6], PV-RCNN [21] and Voxel R-CNN [3] come from published papers, and the AP_{R40} of SECOND [28], PointPillars [8] and PointRCNN [22] come from the results of the officially released code. In addition, we also report the 3D APs under 11 recall positions (AP_{R11}) under moderate setting. Our method achieves the best performance among all competitors. Specifically, our SPC outperforms other methods by 1.33%, 2.26% and 3.10% 3D AP_{R11} on car, pedestrian, cyclist class under moderate setting. Figure 4 shows some predicted results and we project them onto color images for better visualization. As observed, our SPC can produce high quality 3D bounding box in different kinds of scenes.

We also compare our model with state-of-the-art methods on the KITTI *test* set by submitting our results to KITTI online test server. As Table 2 displays, compared with all the models, SPC surpasses them on moderate and hard setting of car class. SPC also shows competitive results on the pedestrians and cyclists class. Those methods include the models that take LiDAR and RGB images as inputs and the ones taking LiDAR input only.

4.3 Ablation Study

Here we provide extensive experiments to analyze the effectiveness of our method. All of the experiments in this section are conducted on the KITTI *val* set.

Component Analysis. We individually evaluate the contributions of Structure Completion and RoI Feature Completion module in our model. We choose

Table 3. Component analysis of our SPC model on KITTI *val* set. **Structure** means the Structure Completion, while the **Feature** means the RoI Feature Completion module. Performance comparisons of different components with 3D AP_{R40}.

Method	**Structure**	**Feature**	**Car**			**Ped.**			**Cyc.**		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Baseline			89.47	80.32	77.92	62.82	53.66	47.01	91.62	71.34	66.77
SPC	✓		91.87	82.68	80.38	65.74	58.21	51.18	93.52	75.48	70.78
SPC		✓	92.62	83.69	81.27	64.17	55.24	48.21	91.67	73.99	69.15
SPC	✓	✓	**92.71**	**85.75**	**83.35**	**66.18**	**58.92**	**51.38**	**93.66**	**75.55**	**70.97**

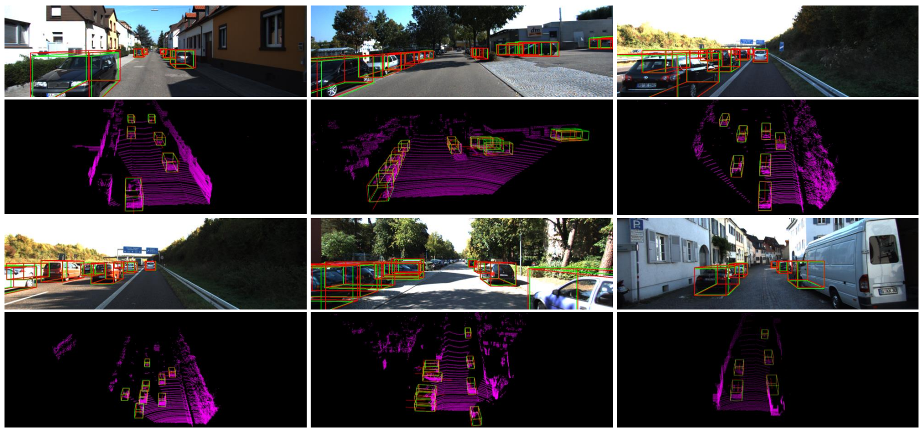

Fig. 4. Qualitative results on KITTI *val* set. The predicted bounding boxes and ground-truth bounding boxes are shown in red and green, respectively, and we project them back onto the color images for better visualization.

PointRCNN [22] as our baseline and all hyperparameters and training procedures are the same, whose mAP for car, pedestrian and cyclist are 81.70%, 56.31% and 74.43%, respectively. Table 3 shows the importance of each component of our SPC model. Simply adding Structure Completion after RPN, the process boosts the performance beyond the baseline, improving the AP_{R40} by 2.41%, 3.88% and 3.34% for three categories, respectively. It clearly demonstrates the effectiveness of Structure Completion module, which helps to alleviate the low quality proposal generation problem of RPN, especially for pedestrian and cyclist under moderate and hard setting. The main reason that leads to this phenomenon, is the inaccurate localization caused by structure missing has bigger impact for pedestrian and cyclist. Compared with car class, pedestrian and cyclist are relative smaller, then the same localization error has a more severe influence on the IoU between the predicted bounding boxes and the ground truth of the small object. Our RoI Feature Completion module contributes an improvement of 3.39%, 1.37% and 1.69% on the three categories, respectively. Finally, we combine the two module and obtain the full model of SPC.

Table 4. Effects of Structure Completion module in different LiDAR-based detection paradigms on KITTI *val* set with 3D AP_{R40}. **Structure** means the Structure Completion module.

Method	Car			Ped.			Cyc.		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointRCNN	89.47	80.32	77.92	62.82	53.66	47.01	91.62	71.34	66.77
PointRCNN+**Structure**	91.87	82.68	80.38	65.74	58.21	51.18	93.52	75.48	70.78
Improvement	*+2.40*	*+2.36*	*+2.36*	*+2.92*	*+4.55*	*+4.17*	*+1.90*	*+4.14*	*+4.01*
PV-RCNN	92.57	84.43	82.69	64.26	56.67	51.91	88.88	71.95	66.79
PV-RCNN+**Structure**	92.65	85.57	83.12	67.43	58.46	51.30	90.42	72.68	67.12
Improvement	*+0.08*	*+1.14*	*+0.43*	*+3.17*	*+1.79*	*-0.61*	*+1.54*	*+0.73*	*+0.33*

Table 5. Ablation on Structure Completion module. Performance comparisons of different Structure Completion strategies with 3D AP_{R40}. H means the points number threshold to sample proposals. L means the number of augmented proposals.

Method	Parameter	Car			Ped.			Cyc.		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Baseline		89.47	80.32	77.92	62.82	53.66	47.01	91.62	71.34	66.77
H	20	91.42	50.54	78.10	62.65	55.15	48.54	90.90	73.62	69.16
	40	**91.87**	**82.68**	**80.38**	65.74	**58.21**	**51.18**	**93.52**	**75.48**	**70.78**
	60	91.01	82.12	79.70	65.32	56.52	49.74	92.95	74.42	69.89
L	4	89.67	80.63	78.21	**66.09**	58.03	50.71	92.20	71.65	67.04
	6	89.01	78.87	77.90	64.83	55.93	48.70	92.07	73.15	68.82
	8	**91.87**	**82.68**	**80.38**	65.74	**58.21**	**51.18**	**93.52**	**75.48**	**70.78**

Effects of Structure Completion. The Structure Completion is easily extended to LiDAR-based 3D detectors. To verify it can play a plug-in to other models, we select PointRCNN [22] and PV-RCNN [21] to test on KITTI *val* set. As shown in Table 4, the Structure Completion module can bring $+0.08\% \sim +4.55\%$ AP_{R40} to the original 3D detector. We also test our model with different Structure Completion strategies. We first analyze the effect of hyperparameter H, as shown in Table 5. When we augment the proposals with fewer points, SPC shows better capacity than the baseline, especially at moderate and hard setting. When proposals with enough points ($H = 60$) are selected to augment, the performance slightly decrease. The reason is that for objects with sufficient points, they are conducive to network's inference of the localization and size of the objects. Augmenting these relatively accurate localization proposals may yield some proposals that are easily misclassified as positive, resulting in performance degradation. The best result comes from the $H = 40$ setting and we select as our default setting. Besides, we further analyze the effect of hyperparameter L, as displayed in Table 5. We get the best result by setting $L = 8$, that means each augmented proposal may make contributions to the finally result, because it is hard to identify which structure is missing.

Table 6. Ablation on RoI Feature Completion module. Performance comparisons of different number of virtual points with 3D AP_{R40}.

Number	**Car**			**Ped.**			**Cyc.**		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
8	92.65	85.57	83.11	64.95	56.07	49.44	91.29	72.46	68.66
27	**92.71**	**85.75**	**83.35**	**66.18**	**58.92**	**51.38**	**93.66**	**75.55**	**70.97**
64	92.54	85.52	83.11	66.00	56.39	48.60	93.23	75.30	70.60

Table 7. Speed and Accuracy comparisons on KITTI *val* set. The inference speed is tested under single RTX 3090 GPU with batch size 1. PR means PointRCNN [22], Structure means Structure Completion and Feature means RoI Feature Completion.

Inference Speed			AP_{R40} **for Car detection**		
PR	PR + Structure	PR + Feature	PR	PR + Structure	PR + Feature
71.1ms	72.0ms	90.4ms	80.32	82.68	83.69

Effects of RoI Feature Completion. We further validate the effectiveness of RoI Feature Completion module. As Table 6 displays, if we place too many virtual points in each proposal, the number of virtual points may be greater than the raw points and degrade the overall performance, because it cannot help to fit a bounding box well. Moreover, the more virtual points we generate, the higher computation costs. We choose the optimal value when the model achieves the best performance, *i.e.*, M =27.

Efficiency Analysis. We analyze the efficiency of our proposed methods from inference speed and accuracy, as shown in Table 7. The proposed method only adds little latency, including 0.9ms for Structure Completion and 19.3ms for RoI Feature Completion module. And we achieve 2.36% and 3.37% accuracy improvements on moderate setting for car detection than the baseline.

5 Conclusion

We present a two-stage 3D object detection framework SPC for outdoor point clouds. It aims to mitigate the problems of low quality proposal generation in RPN and insufficient proposal features in refinement stage. To this end, we introduce a Structure Completion strategy that generates at least one proposal structure similar to the ground truth boxes by traversing most structures. Moreover, we propose a RoI Feature Completion module that helps to obtain adequate proposal features by attentively aggregating raw point features and virtual point features. Our approach achieves competitive performance compared with state-of-the-art methods on KITTI dataset, especially on the hard setting objects.

Acknowledgments. This work was partly funded by NSFC(No.61971281), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and STCSM(18DZ2270700).

References

1. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018)
2. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
3. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. arXiv preprint arXiv:2012.15712 **1**(2), 4 (2020)
4. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
5. Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. Journal of Field Robotics **37**(3), 362–386 (2020)
6. He, C., Zeng, H., Huang, J., Hua, X.S., Zhang, L.: Structure aware single-stage 3d object detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11873–11882 (2020)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
9. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. IEEE transactions on medical imaging **37**(12), 2663–2674 (2018)
10. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7546–7555 (2021)
11. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: 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)
12. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 641–656 (2018)
13. Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., Bai, X.: Tanet: Robust 3d object detection from point clouds with triple attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 11677–11684 (2020)
14. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4721–4730 (2021)
15. Mao, J., Niu, M., Bai, H., Liang, X., Xu, H., Xu, C.: Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2723–2732 (2021)
16. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: 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)
17. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)

18. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
19. Qian, X., Wang, L., Zhu, Y., Zhang, L., Fu, Y., Xue, X.: Impdet: Exploring implicit fields for 3d object detection. arXiv preprint arXiv:2203.17240 (2022)
20. Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.S., Zhao, M.J.: Improving 3d object detection with channel-wise transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2743–2752 (2021)
21. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
22. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
23. Shi, S., Wang, Z., Wang, X., Li, H.: Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. arXiv preprint arXiv:1907.03670 **2**(3) (2019)
24. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4604–4612 (2020)
25. Wu, X., Peng, L., Yang, H., Xie, L., Huang, C., Deng, C., Liu, H., Cai, D.: Sparse fuse dense: Towards high quality 3d detection with depth completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5418–5427 (2022)
26. Xie, L., Xiang, C., Yu, Z., Xu, G., Yang, Z., Cai, D., He, X.: Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 12460–12467 (2020)
27. Xu, Q., Zhong, Y., Neumann, U.: Behind the curtain: Learning occluded shapes for 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2893–2901 (2022)
28. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
29. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11040–11048 (2020)
30. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1951–1960 (2019)
31. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. Advances in Neural Information Processing Systems **34** (2021)
32. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-net: Point cloud upsampling network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2790–2799 (2018)
33. Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.W.: Cia-ssd: Confident iou-aware single-stage object detector from point cloud. arXiv preprint arXiv:2012.03015 (2020)
34. Zheng, W., Tang, W., Jiang, L., Fu, C.W.: 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 (2021)

35. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)