SPV-SSD: An Anchor-free 3D Single-Stage Detector with Supervised-PointRendering and Visibility Representation*

Lingmei Yin[a,1], Wei Tian[a,*,1], Ling Wang[a], Zhiang Wang[a], Zhuoping Yu[a] and Dengcheng Liu[a,b]

[a]Tongji University, 201804 Shanghai, China
[b]Nanchang Automotive Institute of Intelligence and New Energy, Jiangling Group New Energy Vehicle Co. LTD (JMEV), 330200 Jiangxi, China

ARTICLE INFO

Keywords:
3D object detection
sequential fusion
spatial visibility of LiDAR
anchor-free detection

ABSTRACT

Nowadays 3D object detection based on multi-modal sensor fusion has been widely adopted in autonomous driving and robotics, e.g. the fusion of RGB camera semantic segmentation information and light detection and ranging (LiDAR) geometric information to detect 3D objects, as neither of the single modal sensor is able to acquire all the necessary signals. Many state-of-the-art methods fuse the signals sequentially for simplicity. By sequentially we mean using the image semantic signals as auxiliary input for LiDAR-based object detector, the overall performance would heavily rely on the semantic signals and the error introduced by these signals may lead to detection errors. To remedy this dilemma, we propose an approach coined supervised-PointRendering to rectify the potential errors in the image semantic segmentation results by training auxiliary tasks on the fused feature map of the image semantic feature, the laser point geometry feature and a novel laser visibility feature. The laser visibility feature is obtained through the raycasting algorithm and is adopted to constrain the spatial distribution of fore- and background objects. Furthermore, we build an efficient anchor-free Single Stage Detector (SSD) powered by an advanced global-optimal label assignment to achieve a better time-accuracy balance. The new detection framework is evaluated on the widely used KITTI and nuScenes datasets, manifesting the highest inference speed and at the same time outperforming most of the existing single-stage detectors in terms of average precision.

1. Introduction

Accurate and real-time 3D object detection plays an important role in the autonomous driving. Some methods use monocular or stereo cameras to conduct 3D object detection [28]. Despite images can provide fine-grained semantic context, the performance of such methods is limited due to the lack of depth information. On the other hand, methods based on LiDARs [13, 29] can obtain much higher precision with accurate depth information, but are unfriendly to small objects with low resolution and sparse texture information. Thus, the LiDAR-camera fusion scheme offers an opportunity to boost the 3D detection performance. The current fusion dilemma lies in that most of 3D detectors extract features on the bird's-eye view (BEV), which is difficult to align with the front view of camera. The new fusion style at the input stage can avoid this problem. A typical method is the PointPainting [30], which projects the point cloud into the image semantic segmentation results and appends the semantic class scores to each point. This method enables easily missed small objects to be detected due to semantic information, but decreases the precision of large objects which can be detected accurately by LiDAR-only methods (Table 1). From the results reported on the KITTI benchmark [11], it can be seen that the PointPainting contributes to significant improvements on average precision of pedestrians and cyclists, while deteriorating the performance of car detection. The reason is that the sequential fusion makes subsequent detection network heavily rely on the quality of semantic segmentation. As Fig. 1 shows, many background points are misclassified as cars, inevitably leading to more false positives.

Table 1
Deterioration of car detection performance by sequential appending image semantic segmentation results to point cloud. Test results of Painted PointPillars are obtained on validation set using the official implementation while the test results of Painted PointRCNN are from the KITTI server.

Method	Car (%)			Pedestrian (%)			Cyclist (%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [30]	87.22	76.95	73.52	57.75	52.29	47.91	82.29	63.26	59.82
Painted PointPillars	86.26	76.77	70.25	61.50	56.15	50.03	79.12	64.18	60.79
Delta	-0.96	-0.18	-3.27	+3.75	+3.86	+2.12	-3.17	+0.92	+0.97
PointRCNN[25]	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
Painted PointRCNN [30]	82.11	71.70	67.08	50.32	40.97	37.87	77.63	63.78	55.89
Delta	-4.85	-3.94	-3.62	+2.34	+1.6	+1.86	+2.67	+4.96	+3.36

Current 3D object detectors often ignore the fact that measuring the 3D environment as (x,y,z) points destroys the hidden spatial distribution information. According to the physical characteristics of raycasting, "visibility" ensures that there are no obstacles between the LiDAR origin and the detected object, and everything behind the detected object along its line-of-sight is occluded. With the visibility constraint, it can estimate the free area distribution in the 3D space and can provide context information for object

*Project supported by the National Natural Science Foundation of China (No.52002285), the Shanghai Pujiang Program (No.2020PJD075), the Natural Science Foundation of Shanghai (No.21ZR1467400) and the Perspective Study Funding of Nanchang Automotive Institute of Intelligence and New Energy (TPD-TC202110-03).

*Corresponding author
2033616@tongi.edu.cn (Lingmei Yin); tian_wei@tongi.edu.cn (Wei Tian); 1951504@tongi.edu.cn (L. Wang); wangzhiang@tongi.edu.cn (Z. Wang); yuzhuoping@tongi.edu.cn (Z. Yu); liudengcheng@naiine.com (D. Liu)

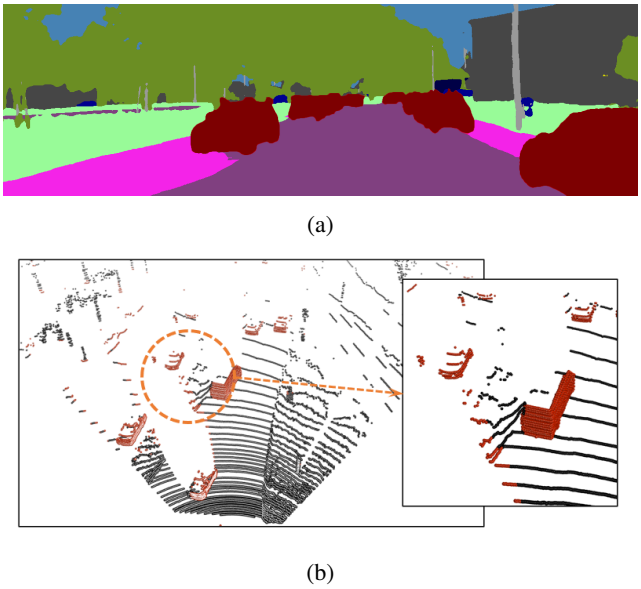
[1]Equal contribution

detection. Moreover, the database sampling is one of the general augmentation strategies, which randomly copies and pastes virtual objects into point cloud scenes. However, such method often inserts objects behind walls or buildings, ignoring their visibility and authenticity. In this paper, we adopt visibility feature to assist both object detection and data augmentation.

Existing voxel-based detectors prefer anchor-based detection head. Anchor-based methods generate dense anchor boxes to guarantee the high recall. However, for complex scenes in nuScenes [3], anchor-based methods will generate dozens of anchors in consideration of the number of categories and orientation bins in each grid, resulting in cubic-level growth of parameters and slowing down the inference. In contrast, anchor-free detection head directly regresses bounding boxes in each grid, capable to run in real-time. Yet the huge solution space brings too many false positives. Recent study [10] proves that a suitable label assignment can boost the performance of anchor-free detection head significantly, which enlightens its application in the 3-D detection.

To address above issues, in this work we present a new anchor-free 3D single-stage object detector with supervised-PointRendering and visibility representation. The supervised-PointRendering decorates points by appending image semantic segmentation results to points, with the additional point-wise supervision task to rectify incorrect semantic segmentation results. We use the raycasting algorithm to reconstruct the spatial visibility features of laser, and fuse them with the point cloud features and image semantic features. We adapt the Sim-OTA label assignment strategy [10], which can select the global assignments with high confidence for all ground truths, to our anchor-free detection head with modifications such as adding orientation to the algorithm and enlarging center prior region of small objects to make it more suitable for 3D detection framework. With such an implementation, our anchor-free model can boost the inference speed while achieving a higher precision. Compared with existing approaches, the main contributions of our work are:

- We propose a novel supervised-PointRendering to eliminate the effects of incorrect image semantics on object detection, significantly improving the detection precision of all categories.
- We introduce laser visibility features into the sequential fusion-based 3D object detection, which supplements more scene context to boost the 3D object detection.
- We creatively design an anchor-free detection head powered by the modified Sim-OTA label assignment in the voxel-based methods, achieving a decent balance between precision and speed. At present, our model has achieved comparable precision with the state-of-the-art single-stage methods on the KITTI and nuScenes datasets with an ultra-high inference speed, showing the generality and efficiency of our model in different traffic scenarios.



(a)

(b)

Figure 1: (a) Semantic segmentation results by DeeplabV3+ [5]; (b) Painted point cloud with the 2D segmentation results. Cars are denoted in red. The zoomed figure depicts the misclassified ground area.

2. Related Work

2.1. 3D Object Detection with Point Cloud

Generally, LiDAR-based 3D object detectors can be divided into two categories depending on their architecture: single-stage and two-stage detectors.

The single-stage detectors directly output class scores and bounding boxes in one stage. For instance, the VoxelNet [36] first voxelizes the point cloud and extracts voxel feature by a tinny PointNet. The SECOND [32] exploits the sparse convolution and submanifold sparse convolution to accelerate 3D convolution. Point-GNN [27] extracts features through a novel graph neural network and achieves comparable performance with state-of-the-art two-stage detectors. SASSD [12] employs an auxiliary network for box center regression and segmentation, aiming to learn the structure and localization information. This inspires us to exploit point-level supervision to eliminate the boundary-blurring effect in semantic segmentation.

Two-stage detectors usually generate region proposals in the first stage, then refine them in the second stage by re-using the point cloud with full resolution. The Voxel-RCNN [8] generates regions of interest (RoIs) by an efficient voxel-based CNN. The PV-RCNN [24] leverages the advantages of 3D voxel CNN and PointNet-based set abstraction to learn more representative features. The Part-A^2 [26] enriches the RoI features by predicting intra-object

part locations in the first stage, reducing the ambiguity of bounding boxes.

It's worth noting that most voxel-based models use anchor-based detection head. However, this mechanism has several drawbacks. First, to achieve the optimal detection performance, one needs to cluster a set of optimal anchors based on training data which requires heuristic tuning. Second, the anchor mechanism greatly increases both the number of predictions and the design complexity of detection heads. On the contrast, the anchor-free mechanism reduces the parameter number of detection head significantly, making the training and decoding phase of detector much simpler. Anchor-free detectors [18, 22, 35] have developed rapidly in 2D detection in recent years and can exceed anchor-based detectors with suitable label assignments.

Given the high efficiency of one-stage network and great potential of anchor-free mechanism in 3D object detection, we focus on developing a novel anchor-free single-stage detector, which has attained comparable performance with the state-of-the-art single-stage detectors with a high inference speed.

2.2. Multi-modal Fusion

Object detection frameworks based on LiDAR-camera fusion have developed rapidly in recent years. According to [30], they can be grouped into following categories: object-centric fusion, continuous feature fusion, detection seeding and sequential fusion.

The representatives of object-centric fusion are MV3D [6] and AVOD [16]. They extract RoI features from the image and point cloud projection view, and then perform deep feature fusion on the BEV or the front view. However, the projection process loses spatial information, which decreases the detection accuracy greatly.

The continuous feature fusion, pioneered by ContFuse [19], fuses features from the image and LiDAR backbone networks at different scales. These methods often calculate a mapping to convert the point cloud to the image plane. The core problem lies in that each BEV feature vector of point cloud corresponds to multiple pixels in the 2D image, resulting in fuzzy feature align and thus limited performance.

The detection seeding methods like Frustrum PointNet [21] and ConvNet [31] utilize the 2D detections to limit the frustum search space to seed the 3D proposal. These detectors rely heavily on the performance of 2D detectors and impose an upper bound on recall.

The sequential fusion is a simple, general yet effective strategy compared with other fusion methods. The LRPD [9] and PointPainting [30] both use the output of image semantic segmentation network to assist object detection. PointPainting [30] projects LiDAR points into the semantic segmentation results and feed the painted points to 3D object detector. However, the "boundary-blurring effect" occurs in image semantic segmentation due to the relative low resolution of the high-level feature map. This effect becomes even worse after reprojecting them into the point cloud.

2.3. Visibility Representation

Most researches on the spatial visibility representation are carried out in the robotic mapping. Buhmann et al. [2] estimate a 2D probabilistic occupancy map based on sonar readings to navigate the mobile robots. Hornung et al. [14] propose a general 3D occupancy map to describe the space state, indicating the occupied, free and unknown area. The visibility representation through raycasting algorithm is the core of constructing such occupancy maps.

Despite the popularity, the visibility reasoning has not been widely researched in the 3D object detection. Richter et al. [23] integrate the occupancy grid map into the probabilistic framework to detect objects with known surface. It is worth noting that Hu et al. [15] propose the concept that LiDAR point clouds are not real "3D" but "2.5D". They reconstruct the spatial visibility state through raycasting algorithm and convert 3D spatial features to 2D multi-channel feature maps, and then integrate the 2D feature maps into the PointPillars [17]. Such visibility features can be directly concatenated with voxelized point cloud and thus bring better data alignment.

3. Proposed Method

In this section, we introduce the anchor-free 3D single-stage object detector with supervised-PointRendering and LiDAR visibility. Sec. 3.1 introduces our backbone and detection networks. Sec. 3.2 introduces the supervised-PointRendering to enhance point cloud feature. Sec. 3.3 describes the extraction of spatial visibility feature of laser to provide scene representation. Sec. 3.4 introduces the anchor-free detection head with improved label assignment. Sec. 3.5 presents the loss functions for training.

3.1. Network Architecture

Our framework consists of data processing, backbone network, and detection head.

In the data processing, we adopt an off-the-shelf image segmentor to output the per pixel semantic class scores, which are then appended to the points. We reconstruct the spatial visibility feature though raycasting algorithm and concatenate them with the voxelized point cloud. The painted point clouds with visibility features are then fed into the backbone network.

Our backbone network contains two modules: a typical 3D convolution backbone network followed by a Spatial-Semantic Feature Aggregation module (SSFA) [33] to extract features and an auxiliary network to exploit point-wise supervisions. As shown in Fig. 2, the backbone network is composed of four convolution blocks, each of them contains submanifold convolutions with a kernel size of 3. The last three blocks downsample the features with a stride of 2. The features from the backbone are then concatenated along the height dimension to produce the BEV feature maps as inputs to the SSFA module. The SSFA module adaptively fuses

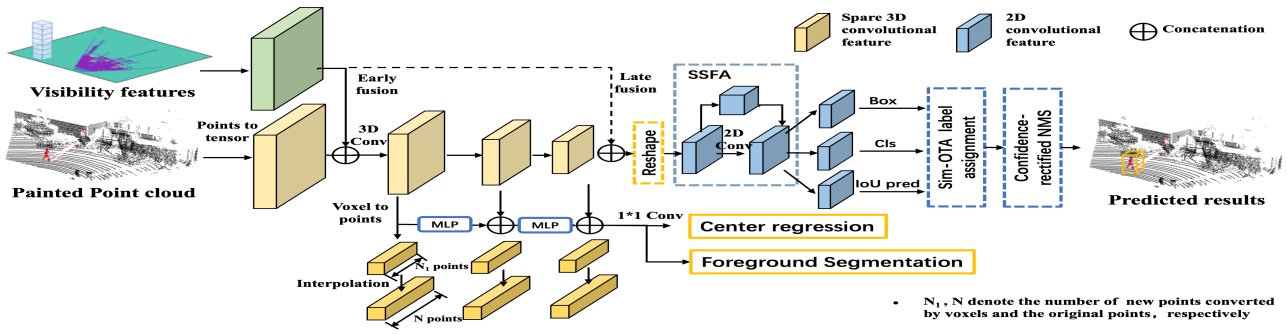


Figure 2: The pipeline of our proposed anchor-free 3D single-stage object detector with supervised-PointRendering and spatial visibility representation.

high-level semantic features and low-level spatial features. The auxiliary network scatters the convolution features back to points, and performs foreground segmentation and center regression tasks.

The anchor-free detection head has three tasks: classification, localization and intersection-over-union (IoU) prediction with a confidence function to rectify the confidence used in NMS post-processing [1]. In the end, we use a global optimal label assignment to find the best label assigning at minimal global costs.

3.2. Supervised-PointRendering

3.2.1. PointRendering

The image semantic segmentation network takes in an input image and outputs semantic class scores per pixel. These scores serve as high-level features of an image. We transform LiDAR points into the camera coordinate system, project them into the image, and append the segmentation scores of corresponding pixels to the LiDAR point features. We choose segmentation scores as image feature encoding because scores implicitly contain uncertainty information, which requires the network to distinguish the semantic category itself, thus reducing the influence of incorrect semantic segmentation on subsequent networks.

3.2.2. Point-wise Supervision Task

As shown in Fig. 2, we utilize the point-wise foreground segmentation to rectify the incorrect segmentation results. The voxel feature is converted to the world coordinate system through inverse voxelization. The feature from N_1 points are propagated to N points by interpolation, where N_1 and N denote the number of new points converted by voxels and the original points, respectively. In the interpolation approach, the feature vector $f(\boldsymbol{x}_j)$ at each coordinate of original points $\boldsymbol{x}_j$ can be calculated by a weighted average of the features of its k nearest neighbors $\boldsymbol{x}'_i$ among the new points (in Eq. 1, k is empirically set to 3), interpreted by

$$f(\boldsymbol{x}_j) = \frac{\sum_{i=1}^{k} w_{ij}(\boldsymbol{x}'_i) f(\boldsymbol{x}'_i)}{\sum_{i=1}^{k} w(\boldsymbol{x}'_i)}, j = 1, \ldots, C, \tag{1}$$

where the weight $w_{ij}(\boldsymbol{x}'_i)$ is the inverse square of distance between the original point $\boldsymbol{x}_j$ and the new point $\boldsymbol{x}'_i$:

$$w_{ij}(\boldsymbol{x}'_i) = \frac{1}{\|\boldsymbol{x}_j - \boldsymbol{x}'_i\|^2}. \tag{2}$$

The obtained point-wise features are further processed by a shared multi-layer perceptron (MLP) to yield point feature encoding $h\left(x_i\right)$, which is then concatenated with next-stage features. We apply 1×1 convolutions to generate predictions for the foreground segmentation task. Besides, we add a center regression to guide the backbone network to learn object structure information. These tasks are detachable in the inference stage, introducing no extra computational cost.

The point-wise supervision loss includes the foreground segmentation loss and the center regression loss, which are interpreted by the focal loss and the smooth-L1 loss, respectively. The foreground segmentation label is a binary value with 1 to indicate that the point is within the ground-truth bounding box. The center regression targets are the offsets from foreground points to corresponding object centers.

3.3. Visibility Representation

3.3.1. Ray Casting Algorithm

The raycasting algorithm in a horizontal plane is depicted in Fig. 3. Given the center coordinates of current voxel (x, y), (t_{mx}, t_{my}) are the time intervals from current position to the boundary of the adjacent voxel along the ray in the x and y direction, respectively. We traverse along the ray to reach the next voxel, update (x, y) and (t_{mx}, t_{my}), and mark the traversed grid. Such a step iterates until it reaches the

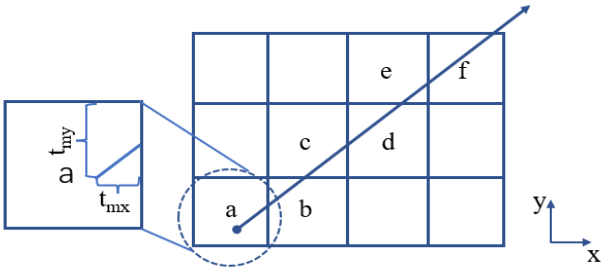


Figure 3: Ray traversal on the 2D grids.

Algorithm 1 Vanilla Voxel Traversal Algorithm

Require: sensor origin $\boldsymbol{s}$, original point set $\mathcal{P}$, voxelized painted points $\mathcal{P}_{vox}$, ending voxel v_p.
Ensure: occupancy grids $\boldsymbol{O}$, voxelized points with visibility states $\mathcal{P}_{vis}$

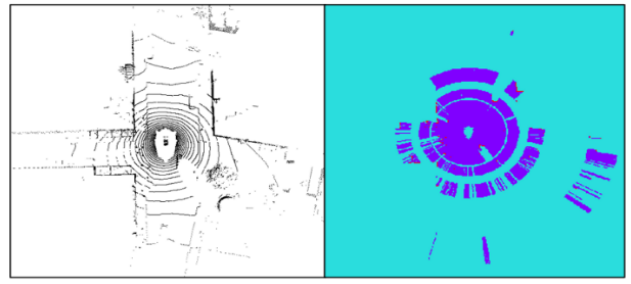
$\boldsymbol{O}$[:]←UNKNOWN;
for $\boldsymbol{p}$ in $\mathcal{P}$ **do**
  v ← $get\_voxel(\boldsymbol{p})$;
  while v ≠ v_p **do**
    v ← $next\_voxel(\mathrm{v}, \boldsymbol{p}-\boldsymbol{s})$;
    if v = v_p **then**
      $\boldsymbol{O}$[v] ← OCCUPIED;
      break;
    else
      $\boldsymbol{O}$[v] ← FREE;
    end if
  end while
  $\mathcal{P}_{vis}$ ← $concat(\mathcal{P}_{vox}, \boldsymbol{O})$;
end for

end of the ray or the last grid. The algorithm can be easily extended to 3D space by adding corresponding variables along the z axis.

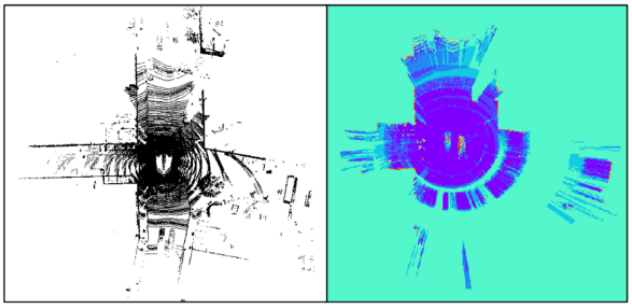
3.3.2. Spatial Visibility States

A grid can be assigned with one of the three visibility states: unknown, occupied and free space, which are represented by specific numerical values. Here we first initialize all voxels as unknown, then execute the raycasting algorithm for each laser ray. If the voxel is located at the end of the ray or at the last grid, it is marked as occupied and the algorithm stops. Otherwise, the voxel is marked as free. Details about this procedure can be referred to Alg. 1.

Fig. 4 depicts the visibility representation of laser rays. It can be seen that the spatial visibility information is consistent with the original point cloud, providing strong support for the quantitative results obtained in the experiment.



(a)



(b)

Figure 4: (a) A single LiDAR sweep and its instantaneous visibility. The denotation of three visibility states are: occupied (red), unknown (light blue), and free (dark blue). (b) The aggregated LiDAR sweeps and their superimposed temporal occupancy states. Color with increased saturation indicates a greater probability of the corresponding voxel occupancy state.

3.4. Anchor-free Detection Head with Improved Label Assignment

3.4.1. Improved Label Assignment for 3D Object Detection

In order to expand the positive area of the small objects, we regard the predictions from the grids which are within a fixed region, e.g., 1m×1m around the center of ground truth boxes, as positive candidates. Then we use the simOTA strategy [10] to finely select the global-optimal positive samples from these positive candidates. The steps are depicted in Fig. 5 and introduced as follows.

- Calculating the cost matrix: its element indicates the pair-wise matching cost between a positive candidate box and a ground truth. The pair-wise cost c_{ij} is composed of the classification loss L_{ij}^{cls}, the box regression loss L_{ij}^{r} and the BEV rotated-IoU loss L_{ij}^{IoU} between a positive candidate i and the ground truth box j, which is calculated as:

$$c_{ij} = L_{ij}^{cls} + L_{ij}^{r} + \lambda L_{ij}^{IoU}. \tag{3}$$

- Calculating the dynamic k: For each ground truth, we calculate its IoU (intersection over union) score with all positive candidates. We select the top N box predictions and sum their IoU scores as k, which is

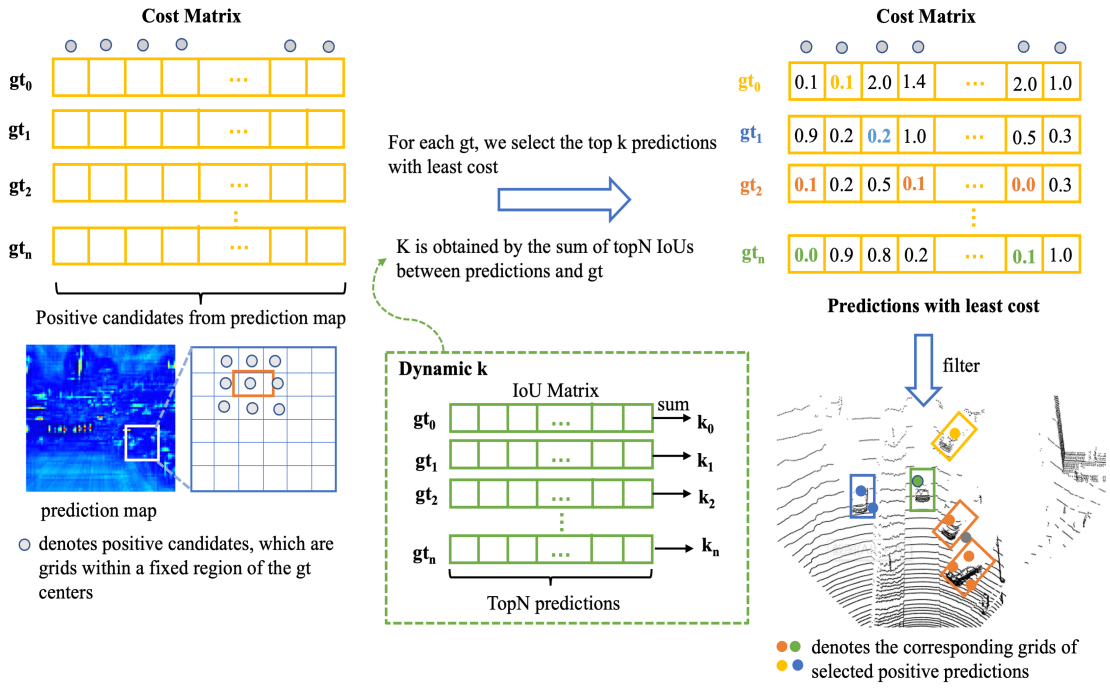


Figure 5: sim-OTA label assignment in 3D object detection

further rounded by the floor operation. Here we set $N = 10$. k is thus the number of positive samples assigned to each ground truth box.

- Selecting the top k predictions with the least cost.
- Filtering the repeated predictions: In the case that the same prediction matches multiple ground truth boxes, we assign the prediction to the ground truth box with the least cost.

Finally, the corresponding grids of those positive predictions are assigned as positives, while the rest grids are considered as negatives.

3.4.2. Anchor-free Detection Head

To achieve real-time performance without decreasing precision, we build an anchor-free detection head with sim-OTA label assignment.

In this detection head, each location directly predicts its offsets to the instance center, as well as the size and orientation of the candidate box. Given that the orientation regression is difficult without prior information, we apply a hybrid formulation of classification and regression to predict the orientation. Specifically, we split 2π into N_a bins. The network predicts both the angle bin and the residuals relative to the bin. In the experiment, we set $N_a = 12$.

To alleviate the misalignment between localization accuracy and classification confidence, we add an IoU prediction branch to rectify the confidence in the non-maximum-suppression (NMS) process. The confidence is calculated by $C = c \cdot i^{\beta}$, where C is the confidence in NMS process, i denotes the predicted IoU, c denotes the classification score, and β is a hyper-parameter set to 4.

3.5. Loss Function

The overall loss L includes the detection head loss L_{head} and point-wise supervision loss for foreground segmentation L_{seg} and center regression L_{ctr}:

$$L = L_{head} + \omega L_{seg} + \mu L_{ctr}. \tag{4}$$

Here we set ω=0.9, μ=2.0 to balance the point-wise supervision task from the main task. The detection head loss includes the classification loss L_c, the box regression loss L_r, the 3D D-IoU loss L_{IoU} and the IoU prediction loss L_{conf}, it is calculated as follows :

$$\begin{aligned} L_{head} = & \frac{1}{N_c}\sum_i L_c(s_i, u_i) + \lambda_1 \frac{1}{N_p}\sum_i [u_i > 0] L_r \\ & + \lambda_2 \frac{1}{N_p} L_{IoU} + \lambda_3 \frac{1}{N_p} L_{conf} \end{aligned} . \tag{5}$$

Among them, λ_i is the weight of each task. N_c and N_p respectively denote the number of total proposals and positive proposals selected by the sim-OTA. s_i is the classification

score for each grid while u_i denotes the corresponding class label. Here, we set $\lambda_1 = 1.0$, $\lambda_2 = 5.0$ and $\lambda_3 = 1.0$. The 3D D-IoU loss is based on [34] and L_{conf} is calculated by the smooth-L1 loss. For classification, we choose the focal loss, which is:

$$\mathcal{L}_{cls} = -\alpha(1 - p^a)^\gamma \log p^a, \tag{6}$$

where p^a is the estimated probability. α and γ are hyper-parameters and are set to 0.25 and 2 following the original paper [20].

The regression loss L_r includes the center regression loss L_{dist}, the size regression loss L_{size} and the angle regression loss L_{angle}. The L_{dist} regularizes offsets from positive grid location to their corresponding instance centers. The targets for L_{size} are the offsets between the real object size and the average size of its category. We adopt the smooth-L1 loss for both L_{dist} and L_{size}. Since the angle localization loss cannot distinguish flipped boxes, our angle loss includes the orientation classification loss L_{ori_cls} and the corresponding residual prediction loss L_{ori_reg}:

$$L_{angle} = L_{ori\_cls}(d_c^a, t_c^a) + L_{ori\_reg}(d_r^a, t_r^a), \tag{7}$$

where d_c^a and d_r^a are the predicted orientation bin and its residual while t_c^a and t_r^a are their ground truths.

4. Experimental Setup

4.1. Datasets

We evaluate our method on the widely used KITTI [11] and nuScenes [3] datasets.

The KITTI detection dataset contains 7,481 training samples and 7,518 testing samples. The training data are further divided into a training set (3712 frames) and a validation set (3769 frames). We evaluate our method on all three classes, i.e., car, pedestrian and cyclist. We choose the average precision (AP) with 40 recall positions as criterion, which follows the IoU threshold of 0.7 for Car and 0.5 for Pedestrian and Cyclist. Also, the dataset is divided into three difficulty levels: easy, moderate and hard, based on the object size, occlusion state, and truncation level.

The nuScenes dataset [3] contains 1,000 scenes, each consisting of a video sequence. For each video, only the key frames (every 0.5s) are annotated with a 360-degree view. The dataset has been officially divided into subsets for training, validation and test, which include 700 scenes (28,130 frames), 150 scenes (6019 frames), 150 scenes (6008 frames), respectively. The annotations include ten classes. The corresponding RGB images that cover the 360-degree field-of-view are also provided for each key frame. The main metrics are the mean Average Precision (mAP) and the nuScenes detection score (NDS). mAP calculates the center offsets on the BEV plane as the evaluation standard instead of the intersection ratio (IoU). NDS is a weighted sum of mAP, and the mean average errors of size (mASE), location (mATE), orientation (mAOE), attribute (mAAE) and velocity (mAVE).

4.2. Setup of Supervised-PointRendering

4.2.1. Setup on KITTI Dataset

We use the DeeplabV3+ [5] as the image semantic segmentation network which is pre-trained on the Cityscapes [7] dataset. Note that in the Cityscapes semantic segmentation benchmark, the “rider" (considered as “pedestrian") and the “bicycle" are two different categories, while in the KITTI object detection they are grouped into one category “cyclist". We solve this problem by mapping a “bicycle" and a “pedestrian", which is within a distance of 1m to the bike, into a “cyclist", and classifying the “bicycle" without people around it as the “background". After PointRendering, the input dimension of point cloud has changed from 4 to 8, namely $(x, y, z, r, s_{bg}, s_{car}, s_{pede}, s_{cyc})$, where r is the intensity and s represents the segmentation score of each class.

4.2.2. Setup on nuScenes Dataset

We choose the HTCNet [4] as image semantic segmentation network due to its outstanding performance and pretrain it on the nuImage[1] dataset. It’s worth noting that the homogeneous transform on nuScenes is a little bit complex because LiDAR and camera work at different frequencies, we adopt the transform procedure from [3]. During the image projection, there are laser points projecting on two images simultaneously if the field of view of two cameras overlap. We use the average segmentation score of the overlapped images to decorate the points. For sweeps that do not have synchronous images, we assign them with images which are adjacent to the capture time of these sweeps.

4.3. Setup of Visibility Representation

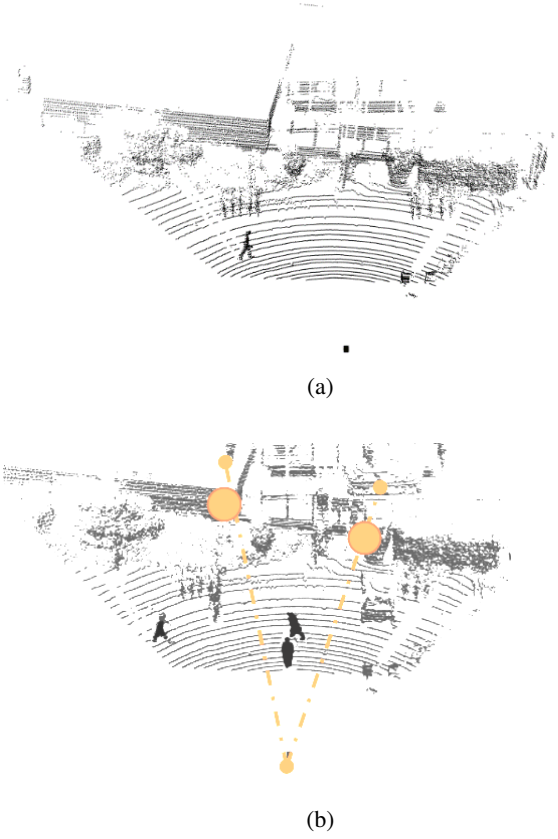
4.3.1. Visibility Consistency in Data Augmentation

To keep the consistency of spatial visibility state after the database sampling of data augmentation, we adopt the“drilling" strategy [15], which allows the rays between the scene origin and added targets to pass through “occupied" voxels in the original scene. As shown in Fig. 6, in order to keep the authenticity of the scene, part of the wall is “drilled" by removing the orange points of the wall.

4.3.2. Spatial Visibility Feature Extraction

Numerical values of the three states are empirically defined as: unknown (0.5), occupied (0.7) and free space (0.4) according to [15]. Given one point cloud frame as input, the LiDAR origin is set to (0, 0, 0). If the input consists of aggregated LiDAR sweeps, the LiDAR origin is changed according to the position of the current frame relative to the previous frame, and Bayesian filtering is applied to accumulate these temporal visibility states as a 3D probability occupancy map.

[1] https://www.nuscenes.org/nuimages



(a)

(b)

Figure 6: The "drilling" strategy in data augmentation through visibility reasoning. (a) The raw point cloud. (b) The point cloud after ground truth database sampling. The orange part will be removed after "drilling".

4.4. Implementation and Training

For the KITTI dataset, the detection range on x, y and z axis is set to [0, 70.4]m, [-40, 40]m and [-3, 1]m , respectively. The input voxel size is set as $0.05 \times 0.05 \times 0.1\text{m}^3$. The network is trained by the ADAM optimizer with an initial learning rate 0.003. We train the network with a batch size of 56 on 8 RTX 2080 Ti GPUs. The learning rate is decayed by 10 at 35 epochs and 40 epochs.

For the nuScenes dataset, the detection range on x, y and z axis is set to [-51.2, 51.2]m, [-51.2, 51.2]m, [-5, 3]m, respectively. The input voxel size is set as $0.1 \times 0.1 \times 0.2\text{m}^3$. We train the entire network with a batch size 48 for 20 epochs on the same computer platform.

Here we decouple the semantic segmentation network and the spatial visibility calculation from the detection framework for a better storage management and inference speed measurement.

5. Experimental Results And Analysis

5.1. Comparison with State-of-the-arts

5.1.1. Results on Kitti Test Set

As shown in the Table 2, our model outperforms other advanced methods on the "moderate" and "hard" difficulty level of car and cyclists by a large margin. Specifically, our method outperforms PointPillars, SECOND and VoxelNet respectively by 6.03%, 4.38% and 15.23% on the moderate AP. Compared to the very recent method SA-SSD [12] and CIA-SSD [33], our model achieves a gain by about 1.24% and 2.53% on the hard AP, showing the effectiveness of our strategy.

The results manifest that our model deals quite well with the difficult objects, which are with more occlusion or sparser points. Moreover, the accurate image semantic segmentation results provided by the supervised-PointRendering effectively eliminate the false negative predictions caused by sparse points and occlusions. Also, the inference speed of our model outperforms all state-of-the-art voxel-based single stage detectors. The high efficiency of our model is mainly due to the anchor-free detection head, which reduces a large number of parameters and computation cost.

However, our mAp on pedestrian is lower than the state-of-the-art methods on the test set. One of the causes is assumed that the number of pedestrians in KITTI is much smaller than that of cars. Even with database sampling in data augmentation, the variety of pedestrians is still lacking. Additionally, the voxel resolution of a pedestrian on the BEV is much worse than that of a cyclist or a car (see Fig.7). Due to above issues, to learn a pedestrian detector well with voxelized features is difficult on the KITTI dataset. Similar problem can also be seen with the VoxelNet [8].

5.1.2. Results on KITTI Validation Set

We compare our method with other advanced single-stage methods in Table 3. It can be seen that the direct PointPainting decreases the precision of car detection, while our supervised-PointRendering successfully solves this problem, with gains of (1.73%, 6.49%, 10.39%) on three difficulty levels on the car category compared with the SECOND baseline. Our method also surpasses the SA-SSD and CIA-SSD by 3.06% and 3.16% on the "moderate" difficulty level of car, which is remarkable.

5.1.3. Results on nuScenes Validation Set

Given the fact that the nuScenes dataset provides more classes in different scales and its point clouds are sparser, a modified version of our method is presented. In this version, we set those categories into 5 groups and adjust our detection head to predict the specific group of objects [37], so that objects of similar shapes or sizes could benefit from each other. The results (Table 4) on the validation set show that both our original network and the modified network achieve

Table 2
The 3D detection AP of compared methods on the KITTI test data.

Method	Car (%)			Pedestrian (%)			Cyclist (%)			FPS
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
Voxelnet [36]	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37	-
SECOND [32]	84.65	75.96	68.71	45.31	35.52	33.14	75.83	60.82	53.67	20
PointPillars [17]	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92	42
Point-GNN [27]	88.33	79.47	72.29	**51.92**	**43.77**	**40.14**	**78.60**	63.48	57.08	-
SA-SSD [12]	88.75	79.79	74.16	N/A	N/A	N/A	N/A	N/A	N/A	25
CIA-SSD [33]	**89.59**	80.28	72.87	N/A	N/A	N/A	N/A	N/A	N/A	32
Ours	87.22	**80.34**	**75.40**	45.83	38.45	36.03	78.36	**64.40**	**56.92**	**33**

Table 3
The 3D detection AP of compared single-stage methods on the KITTI validation set. The AP is calculated with 11 recall points.

Method	Car (%)			Pedestrian (%)			Cyclist (%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [17]	87.22	76.95	73.52	57.75	52.29	47.91	82.29	63.26	59.82
Painted PointPillars [30]	86.26	76.77	70.25	**61.50**	**56.15**	50.03	79.12	64.18	60.79
Delta	-0.96	-0.18	-3.27	+3.75	+3.86	+2.12	-3.17	+0.92	+0.97
VoxelNet [8]	81.97	65.46	62.85	57.86	53.42	48.87	67.17	47.65	45.11
SECOND [32]	87.43	76.48	69.10	N/A	N/A	N/A	N/A	N/A	N/A
SA-SSD [12]	**90.15**	79.91	78.78	N/A	N/A	N/A	N/A	N/A	N/A
CIA-SSD [33]	90.04	79.81	78.80	N/A	N/A	N/A	N/A	N/A	N/A
Ours	89.16	**82.97**	**79.49**	60.75	55.67	**50.20**	**83.67**	**69.59**	**64.21**

remarkable performance, validating the generality of our model on the different driving platforms. Our original model achieves an improvement of 5.27% and 8.66% on the NDS and mAP metrics over the SECOND baseline. Specifically, it improves the AP on bus, truck and trailer by almost 2 times. Besides, our modified model surpasses the MEGVII [37] by 9.99%, 17.03%, 12.46% and 14.38% on car, bus, pedestrian and motorcycle (Moto.), respectively. Notably, our model significantly improves the detection precision of difficult object classes such as constructions (Cons) and bicycles.

5.2. Ablation Study

In this section, we first study the effect of anchor-free detection head with the sim-OTA label assignment. Then we conduct a comprehensive analysis of the effect of supervised-PointRendering and spatial visibility feature. Experiments are conducted on the KITTI validation set.

5.2.1. Anchor-free Detection Head with Sim-OTA Label Assignment

The Table 5 shows that our anchor-free head with the sim-OTA label assignment in 3D object detection boosts the moderate AP of car by 2.08%. The combination of anchor-free head and suitable label assignment strategy not only reduce the parameter amount to $\frac{1}{N_{class} \times N_{ori}}$, but also yields better performance compared to anchor-based approaches such as SA-SSD [12] and CIA-SSD [33].

In addition, we adopt the 3D D-IoU loss and an IoU prediction branch to further improve the AP of each category. The results manifest that these strategies contribute a lot on large objects such as cars. The baseline for the subsequent ablation experiments is the modified SECOND network with above strategies.

Table 4
Comparison with state-of-the-art methods on the nuScenes validation dataset. The performance of PointPillars, SECOND, and MEGVII was reproduced using their official implementation. Here M.G.head represents multi-group head.

Method	NDS (%)	mAP (%)	AP (%)									
			Car	Pedestrian	Bus	Barrier	T.C.	Truck	Trailer	Moto.	Cons	Bicycle
PointPillars [17]	45.32	29.57	70.51	59.93	34.49	33.21	29.62	25.04	20.01	16.75	4.50	1.67
SECOND [32]	48.40	27.12	75.53	59.86	29.04	32.21	22.49	21.88	12.96	16.89	0.36	0.00
MEGVII [37]	44.15	37.68	71.61	65.28	50.29	48.62	45.65	35.77	20.19	28.20	10.56	0.63
Ours	53.67	35.78	79.57	68.07	53.77	33.93	29.05	34.71	23.67	22.65	11.27	1.04
Ours (M.G.head)	**68.41**	**50.81**	**81.60**	**77.74**	**67.32**	**59.21**	**57.43**	**51.70**	**37.50**	**42.58**	**15.76**	**17.34**

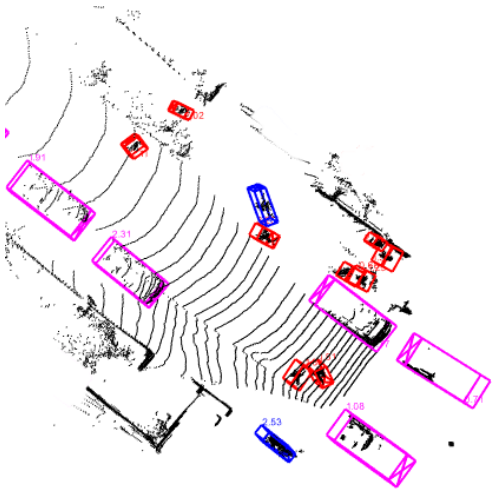


Figure 7: Qualitative results on the KITTI dataset. The pink, blue, and red boxes represent cars, cyclists and pedestrians, respectively.

Table 5
Ablation study of different strategies for the detection head. The AP is calculated with 11 recall points.

anchor-free	3D D-IoU	IoU pred.	Car (%) Mod.	Pedestrian (%) Mod.	Cyclist (%) Mod.
			76.48	51.14	66.74
✓			78.56	**54.55**	67.65
✓	✓	✓	**81.85**	53.70	**68.13**

5.2.2. *Supervised-PointRendering*

This experiment compares different semantic encodings such as the numerical number of object category, segmentation score, one-hot encoding and VoxelRendering. The VoxelRendering is to voxelize the segmentation scores, extract semantic features through 3D sparse convolution, and fuse them with point cloud features before downsampling. Among them, segmentation score encoding attains the largest gain in AP, which is 0.9% higher than that with numerical number encoding on the car category. We argue that the segmentation score implies the classification confidence information, guiding the model to distinguish the category itself.

It can be seen in Table 6 that the AP on car drops after pure PointRendering. After utilizing point-wise supervision, our model boosts the moderate AP by 1.86%, indicating that the foreground segmentation task corrects the segmentation errors successfully. For pedestrian and cyclist, supervised-PointRendering even improves their performance by 3.94% and 0.04%, showing the importance of accurate image semantics in small object detection.

In Fig. 5, we also present qualitative examples to illustrate the benefit of our supervised-PointRendering. It can be seen that the SECOND with PointRendering alone misses more cars, while our method avoids this phenomenon and achieves the best detection performance.

Table 6
Ablation study of supervised-PontPainting. The class_id, one-hot, VR, seg. score and SUPV denote the numerical number of object category, one-hot encoding, VoxelRendering, segmentation score and supervised-PointRendering, respectively. The AP is calculated with 11 recall points.

class_id	one-hot	VR	seg. score	SUPV	Car (%) Mod.	Ped. (%) Mod.	Cyc. (%) Mod.
					81.85	53.70	68.13
✓					78.45	57.36	64.48
	✓				77.92	55.59	65.12
		✓			79.10	54.19	64.25
			✓		79.35	55.10	67.20
			✓	✓	**83.21**	**57.64**	**68.17**

Table 7
Ablation study of spatial visibility fusion. vis.(early) indicates visibility feature in early fusion while vis.(late) is for late fusion. The AP is calculated with 11 recall points.

vis.(early)	vis.(late)	Superv.	Car (%) Mod.	Ped (%) Mod.	Cyc. (%) Mod.
			81.85	53.70	68.13
✓			79.33	**57.98**	69.54
	✓		78.26	55.91	69.03
✓		✓	**82.97**	55.67	**69.59**

5.2.3. *Spatial Visibility Fusion*

Here we compare fusion of spatial visibility feature at different stages, as shown in Fig. 2. Results in Table 7 show that the early fusion performs better than late fusion, indicating that the early fusion is more conducive to data alignment. After supplementing the spatial visibility information, the detection performance of small objects improves significantly. The moderate APs of pedestrian and cyclist increase by 3.07% and 1.41%, respectively. Since the “Easy" and “Moderate" difficulty level in KITTI represent the situation that can be completely observed or less occluded, it is consistent with the meaning of “visibility" in principle.

However, by employing both spatial visibility and supervised point-pointing, the AP on pedestrian drops a little. Since the spatial visibility is calculated based on voxels, we assume that it is the similar problem as in Section 5.1.1. Therefore, the learning of spatial visibility negatively impacts the performance of supervised point-pointing on pedestrian detection.

6. Conclusion

In this work, we present an anchor-free detection framework with a novel sensor fusion method and LiDAR visibility representation. We propose the supervised-PointRendering, which uses point-wise supervision to eliminate the influence of erroneous boundary segmentation on large object detection, improving the precision of all categories by a large margin. Current 3D object detectors process 3D point data with their coordinates yet ignore their hidden information

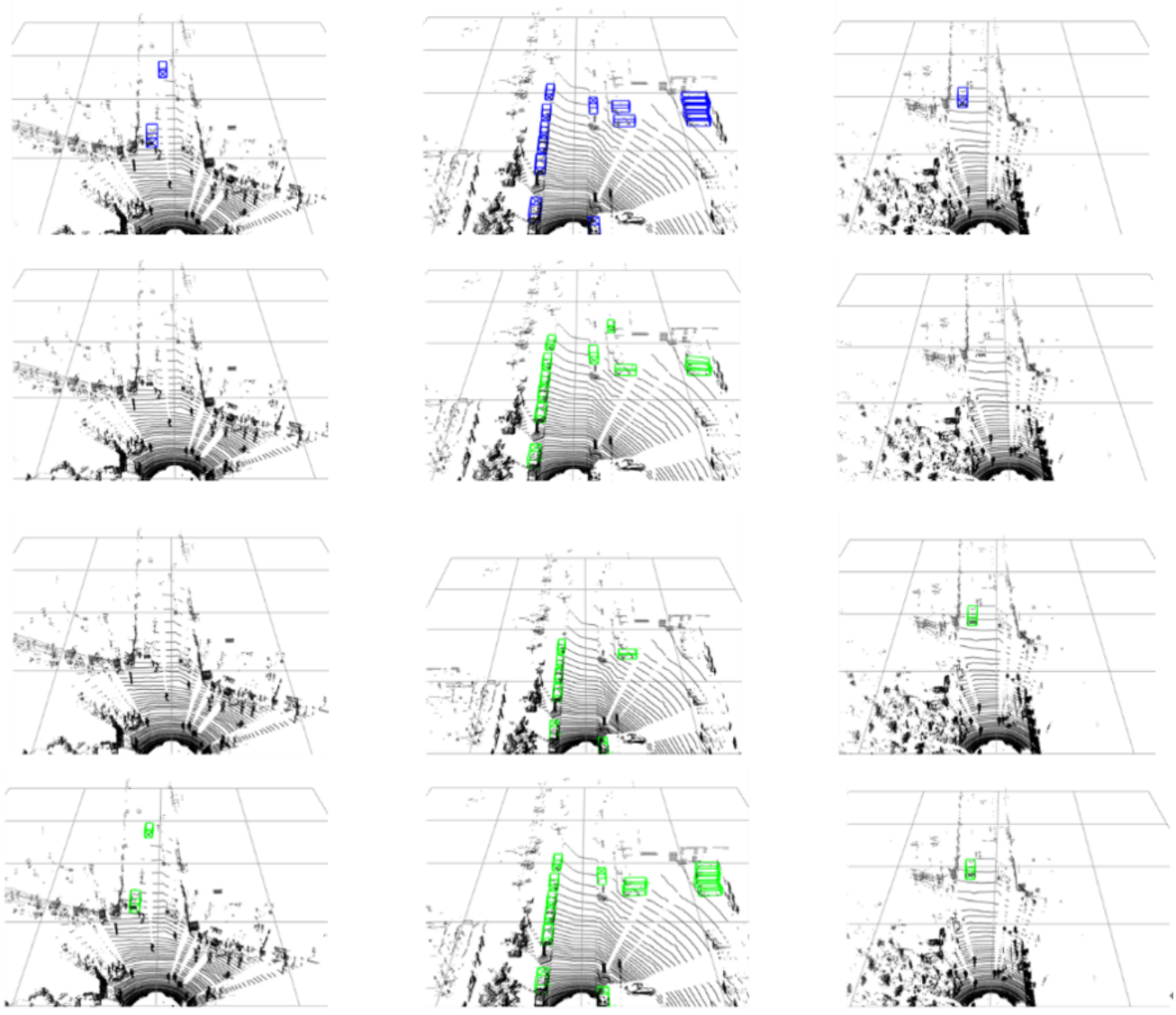


Figure 8: From top to bottom are ground truths, detection results of SECOND, painted SECOND and our SPV-SSD on KITTI dataset. The ground truths and predictions are labeled in blue and green respectively.

about the free space. Here we introduce spatial visibility features of LiDAR to provide more spatial information for object detection. Our detector solves the bottleneck of the inference time by combining anchor-free head with suitable label assignment. By the experiment results on the KITTI and nuScenes datasets, we demonstrate that our model achieves the comparable performance with state-of-the-art single-stage methods with an ultra-high inference speed. The test results on the datasets also show that our detector is with strong robustness against different traffic scenarios.

CRediT authorship contribution statement

Lingmei Yin: Methodology, Software, Validation, Writing - original draft, Formal analysis. **Wei Tian:** Conceptualization, Methodology, Experimental Design, Writing - Review & Editing, Project administration, Funding acquisition. **Ling Wang:** Software, Data Curation, Writing - Original Draft, Visualization. **Zhiang Wang:** Writing - review & editing. **Zhuoping Yu:** Supervision. **Dengcheng Liu:** Resources.

Declaration of Competing Interest

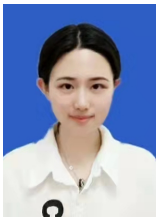
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Bodla, N., Singh, B., Chellappa, R., Davis, L.S., 2017. Soft-nms — improving object detection with one line of code. IEEE International Conference on Computer Vision (ICCV) , 5562–5570.

[2] Buhmann, J.M., Burgard, W., Cremers, A.B., Fox, D., Hofmann, T., Schneider, F.E., Strikos, J., Thrun, S., 1995. The mobile robot rhino, in: SNN Symposium on Neural Networks.

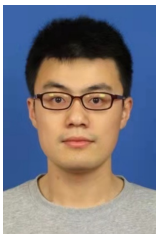
[3] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 11618–11628.
[4] Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D., 2019. Hybrid task cascade for instance segmentation. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 4969–4978.
[5] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. ArXiv abs/1802.02611.
[6] Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3d object detection network for autonomous driving. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 6526–6534.
[7] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 3213–3223.
[8] Deng, J., Shi, S., Li, P., gang Zhou, W., Zhang, Y., Li, H., 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection, in: AAAI.
[9] Fürst, M., Wasenmüller, O., Stricker, D., 2020. Lrpd: Long range 3d pedestrian detection leveraging specific strengths of lidar and rgb. IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) , 1–7.
[10] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. Yolox: Exceeding yolo series in 2021. ArXiv abs/2107.08430.
[11] Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. IEEE Conference on Computer Vision and Pattern Recognition , 3354–3361.
[12] He, C.H., Zeng, H., Huang, J., Hua, X., Zhang, L., 2020. Structure aware single-stage 3d object detection from point cloud. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 11870–11879.
[13] He, Y., Xia, G., Luo, Y., Su, L., Zhang, Z., Li, W., Wang, P., 2021. Dvfenet: Dual-branch voxel feature extraction network for 3d object detection. Neurocomputing 459, 201–211.
[14] Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. Octomap: an efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots 34, 189–206.
[15] Hu, P., Ziglar, J., Held, D., Ramanan, D., 2020. What you see is what you get: Exploiting visibility for 3d object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 10998–11006.
[16] Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L., 2018. Joint 3d proposal generation and object detection from view aggregation. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , 1–8.
[17] Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. Pointpillars: Fast encoders for object detection from point clouds. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 12689–12697.
[18] Law, H., Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. ArXiv abs/1808.01244.
[19] Liang, M., Yang, B., Wang, S., Urtasun, R., 2018. Deep continuous fusion for multi-sensor 3d object detection, in: ECCV.
[20] Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P., 2020. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 42, 318–327.
[21] Qi, C., Liu, W., Wu, C., Su, H., Guibas, L.J., 2018. Frustum pointnets for 3d object detection from rgb-d data. IEEE/CVF Conference on Computer Vision and Pattern Recognition , 918–927.
[22] Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 6517–6525.
[23] Richter, S., Wirges, S., Königshof, H., Stiller, C., 2019. Fusion of range measurements and semantic estimates in an evidential framework / fusion von distanzmessungen und semantischen größen im rahmen der evidenztheorie. tm - Technisches Messen 86, 102 – 106.
[24] Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H., 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 10526–10535.
[25] Shi, S., Wang, X., Li, H., 2019a. Pointrcnn: 3d object proposal generation and detection from point cloud. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 770–779.
[26] Shi, S., Wang, Z., Wang, X., Li, H., 2019b. Part-a2 net: 3d part-aware and aggregation neural network for object detection from point cloud. ArXiv abs/1907.03670.
[27] Shi, W., Rajkumar, R.R., 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 1708–1716.
[28] Shi, Y., Mi, Z., Guo, Y., 2022. Stereo centernet based 3d object detection for autonomous driving. Neurocomputing 471, 219–229.
[29] Tong, G., Peng, H., Shao, Y., Yin, Q., Li, Z., 2021. Ascnet:3d object detection from point cloud based on adaptive spatial context features. Neurocomputing .
[30] Vora, S., Lang, A.H., Helou, B., Beijbom, O., 2020. Pointpainting: Sequential fusion for 3d object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 4603–4611.
[31] Wang, Z., Jia, K., 2019. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , 1742–1749.
[32] Yan, Y., Mao, Y., Li, B., 2018. Second: Sparsely embedded convolutional detection. Sensors (Basel, Switzerland) 18.
[33] Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.W., 2021. Cia-ssd: Confident iou-aware single-stage object detector from point cloud, in: AAAI.
[34] Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R., 2019a. Iou loss for 2d/3d object detection. International Conference on 3D Vision (3DV) , 85–94.
[35] Zhou, X., Wang, D., Krähenbühl, P., 2019b. Objects as points. ArXiv abs/1904.07850.
[36] Zhou, Y., Tuzel, O., 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition , 4490–4499.
[37] Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G., 2019. Class-balanced grouping and sampling for point cloud 3d object detection. ArXiv abs/1908.09492.



Lingmei Yin received the B.Eng degree in Vehicle engineering from Tongji University, Shanghai, China, in 2020 and is pursuing the M.S. degree in Vehicle Engineering in Tongji University. She is currently working on computer vision and machine learning with research areas of 3-D object detection.

Wei Tian received the B.Sc degree in mechatronics engineering from Tongji University, Shanghai, China, in 2010, and received the M.Sc. degree in electrical engineering and information technology at KIT, Karlsruhe, Germany, in 2013. From 2013, he was with the institute of measurement and control systems at KIT and received the Ph.D. degree in 2019. Afterwards, he continued his post-doctoral research at KIT. From 2020, he is a leader of comprehensive perception research group at School of Automotive Studies, Tongji University. He is currently working on computer vision and machine learning with research areas of robust object detection and trajectory prediction.

Ling Wang is currently pursuing the B.Eng degree in Software Engineering in Tongji University, Shanghai, China, expected in 2023. His research interest is computer vision.



Zhiang Wang received the B.Eng. and M.Eng. degrees in Vehicle Engineering from the School of Automotive Studies and Sino-German College, Tongji University, Shanghai, China, in 2017 and 2021, respectively and received the M.Sc. degree in Mechanical Engineering at the KIT Department of Mechanical Engineering, Karlsruhe, Germany, in 2021. His research interests include autonomous driving and computer vision.

Zhuoping Yu received the B.Sc degree in mechatronics engineering from Tongji University, Shanghai, China, in 1982, the M.Sc. degree in engineering mechanics from Tongji University in 1985, and a Ph.D. degree in automotive design and manufacturing from Tsinghua University, Beijing, China, in 1996. He began to teach at Tongji University in 1985 and went to the Braunschweig Automotive Research Institute in Germany, Volkswagen R&D Department, and Darmstadt University Automotive Research Institute for research work. He is the former Dean of the School of Automobile of Tongji University. He is currently working on clean energy automotive engineering.

Dengcheng Liu received the Ph.D. degree in Vehicle Engineering from the School of Automotive Studies, Tongji University, Shanghai, China, in 2020. Now, he's a Industrial Strategy Researcher in Nanchang Automotive Institute of Intelligence and New Energy. At the same time, he also work as a postdoctoral fellow at the School of Electronic and Information Engineering, Tongji University & Jiangling Group New Energy Vehicle Co. LTD (JMEV). His research interests include new energy vehicle and fault diagnosis.