VIN: Voxel-based Implicit Network for Joint 3D Object Detection and Segmentation for Lidars

Yuanxin Zhong
zyxin@umich.edu

Minghan Zhu
minghanz@umich.edu

Huei Peng
hpeng@umich.edu

Mechanical Engineering
University of Michigan
Ann Arbor, USA

Abstract

A unified neural network structure is presented for joint 3D object detection and point cloud segmentation in this paper. We leverage rich supervision from both detection and segmentation labels rather than using just one of them. In addition, an extension based on single-stage object detectors is proposed based on the implicit function widely used in 3D scene and object understanding. The extension branch takes the final feature map from the object detection module as input, and produces an implicit function that generates semantic distribution for each point for its corresponding voxel center. We demonstrated the performance of our structure on nuScenes-lidarseg, a large-scale outdoor dataset. Our solution achieves competitive results against state-of-the-art methods in both 3D object detection and point cloud segmentation with little additional computation load compared with object detection solutions. The capability of efficient weakly supervised semantic segmentation of the proposed method is also validated by experiments.

1 Introduction

3D object detection and scene understanding are two major perception tasks for 3D computer vision. The former task provides position and dimension information for dynamic objects of interest, while the latter task helps to understand the environment, which is usually accomplished by semantic segmentation of the sensor data. These tasks are important for autonomous driving and mapping. The detected 3D bounding boxes are useful for object behavior prediction, while the semantic information is useful for lane-keeping and static obstacle avoidance. Significant prior works exist using lidar (Light Detection and Ranging) sensors for these tasks thanks to their superior ranging precision and robustness to certain environmental factors such as low/high lighting.

During the past decade, techniques for object detection and segmentation have advanced significantly. Convolutional Neural Networks (CNN) have been widely used, including [13, 33, 36] for 2D object detection, [6, 16, 46] for 2D semantic segmentation, [31, 44, 49] for 3D object detection and [47, 51] for 3D semantic segmentation.

Few of these works combine the detection and segmentation tasks, several follow the panoptic segmentation scheme proposed by [18]. Examples include [39] on 2D images, and [51] on 3D point clouds. The simultaneous multi-task learning strategy is favorable for real-time applications, since it's usually computationally efficient to generate detection and segmentation results simultaneously, as some computations can be re-purposed.
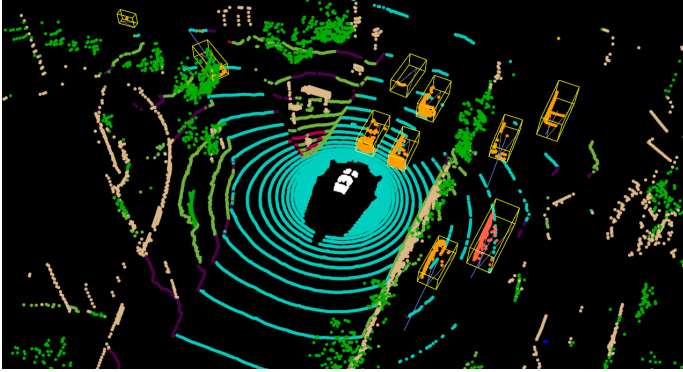

Figure 1: An example of simultaneous object detection and segmentation.
Legend of point clouds: ■ *Car* ■ *Truck* ■ *Pedestrian* ■ *Vegetation* ■ *Sidewalk* ■ *Drivable Surface* ■ *Terrain* ■ *Man-made* Best viewed in color and zoomed in.

It's worth remembering that 3D detection is different from 2D detection in the sense that a 2D bounding box is usually a tight bound of corresponding instance mask, while 3D detection (in outdoor scenes) provides object shape and orientation. Instance segmentation in 2D is also inherently different from 3D instance segmentation in terms of overlap, since there may exist occluded objects in 2D images where pixels of one object may locate inside the bounding box of another object, which is not the case in 3D point clouds, where bounding boxes should not overlap under normal situations. These differences make the design of panoptic segmentation methods different between 2D and 3D data.

In this paper, we propose a method called VIN (**V**oxel-based **I**mplicit **N**etwork), which takes a 3D lidar point cloud as input and reports both 3D object detection and semantic segmentation results as the outputs. To the best of our knowledge, this is one of the first papers that perform lidar-based 3D object detection and point cloud segmentation through a single network. Our main contributions include:

1. A semantic branch from a voxel-based 3D object detector which adds little computation overhead for the additional output. The semantic branch can be trained with weak supervision. The performance with just **0.1%** semantic labels, after some training, was found to be on par with a model trained with full supervision.

2. A strategy to fix inconsistency between bounding boxes and point-wise semantic labels, validated by our experiments.

3. Improved semantic and panoptic segmentation quality compared with state-of-the-art methods based on the results from the nuScenes-lidarseg dataset.

The remainder of the paper is organized as follows: Section 2 introduces related work in 3D object detection and segmentation, Section 3 describes the structure of the proposed network, Section 4 presents the settings and performance metrics of our experiments. Finally, Section 5 concludes this paper.

2 Related Work

2.1 3D Object Detection

Due to the wide availability of various datasets including [4, 12, 14], 3D object detection has become a hot topic in computer vision. Grouped by the modality of sensors used in the detection, the methods in literature can be categorized as image only [6, 7, 27, 40], point cloud only [31, 34, 42, 44, 49] and multi-modal fusion [26, 35]. Among the lidar-based detection algorithms, the approaches include feature extraction by points, voxels, projected images, and combinations [20, 34]. The point-based approaches [31] apply Multi-Layer Perceptron (MLP) and gather feature from points in the same group, while the voxel-based methods [35, 44, 49] first assign the point cloud into voxel grids, and then convolutional neural network (CNN) modules are applied to the voxels. The projection-based methods [26] project the point cloud into images in a perspective view and then features are extracted using 2D CNN models. After the feature extraction, these methods usually generate 3D bounding boxes in a single-staged or two-staged manner ([22] and [33] respectively).

2.2 Point Cloud Semantic Segmentation

Point cloud semantic segmentation is an emerging research field taking advantage of datasets including [2, 4]. Similar to 3D object detection using point clouds, the segmentation algorithms also can use different approaches. PointNet[29] and its successors [30, 37] directly operate on a point cloud array and report point-wise semantics. Voxel-based frameworks [1, 9, 11] extract features for voxel grid convolution and reports voxel-wise semantics. Due to the cubic growth of computational cost with the number of voxel grids, sparse convolution [9, 42] is widely applied for these methods. Voxels are usually sliced in the 3D Cartesian coordinate, while there are also methods [17, 48, 51] in the polar or cylindrical coordinate systems. Methods using projected images [41] for semantic segmentation take advantage of well-explored techniques developed for the 2D segmentation.

Different representations for point clouds have respective strengths and drawbacks. Point-wise feature operation preserves the granularity of the original point cloud, but suffers from heavy computation cost. Voxel-wise feature operations are fast, but suffer from lower precision introduced by the rasterization. Projection-based methods can take advantage of handy 2D convolution structures and their efficiency but need to learn to reconstruct the 3D object shape. In this paper, we enhance the voxel-based framework for its efficiency, by using the implicit representation introduced in Section 3, to enable precise point-wise predictions.

Recently, many researchers [15, 23, 24] work to advance the ability of point cloud semantic segmentation with fewer labels, namely weakly or semi-supervised semantic segmentation. Most of them depend on unsupervised algorithms [15, 23] to cluster the point cloud, or depends on consistency between data frames [24]. In this paper, we propose a method that can handle weak supervision with the help of object detection results.

2.3 Joint 3D Detection and Segmentation

Both 3D object detection and environment segmentation are important tasks for autonomous driving. Relatively few papers combine the two tasks to provide richer information for downstream modules. [26] achieved joint 3D object detection and point cloud segmentation by combining the features from the lidar point cloud and camera image and extrinsic projection

to collect additional point cloud features from the image feature map. [38] improved semantic segmentation performance by adding object detection as an auxiliary downstream stage for additional supervision, which shares a similar structure as this paper, but they started from a point-based semantic backbone. In addition, the two approaches cited above only produce results in a field of view limited by the camera. In this paper, we aim to develop detection and segmentation results without this limitation.

There are many other methods [17, 51] that produce panoptic segmentation results of a point cloud, which is different from joint 3D detection and segmentation as discussed in Section 1. [17] proposed a shifting network module to predict whether a point belongs to a certain instance or not by location regression. [51] proposed a panoptic segmentation framework based on a cylindrical representation of the point cloud augmented by asymmetrical convolutional modules. Compared with these methods, our method achieves panoptic segmentation by generating instance labels with predicted bounding boxes.

2.4   Implicit Representation

Many recent computer vision algorithms use 2D or 3D grids which result in loss of data granularity. To preserve precision, a possible solution is to operate on the raw points, and another way is to use implicit representation such as signed distance functions [25, 28]. The key insight behind the implicit representation is to learn a function to represent the input location, instead of learning the collection of per-location predictions given the feature map input. The prediction can be occupancy of the location as used in [25], or a direct semantic label as what we used in this paper. A major benefit of this approach is that we can predict continuously, i.e. prediction can be obtained for arbitrary query positions, whether it's aligned with the original data or not.

3   Methodology

Different from most methods mentioned in Section 2, our goal is to generate 3D bounding boxes and point-wise semantic labels simultaneously given a point cloud input. On top of these results, panoptic segmentation predictions can be generated easily. First, We will describe the backbone applicable to our algorithm in Section 3.1. Then the details of the proposed semantic branch will be elaborated in 3.2. The panoptic segmentation will be covered in Section 3.3. Figure 2 illustrates the structure of the proposed method.

3.1   Backbone Network

Our method can work with any voxel-based detectors that predict bounding boxes based on a feature map. To show the effectiveness of our semantic branch, we use CenterPoint[44] as our backbone. CenterPoint follows the VoxelNet[49] structure, consisting of voxel feature extraction (VFE) layers, a 3D convolutional backbone, a bird's eye view 2D convolutional backbone and several detection heads (see the upper branch of Figure 2. After object proposals are generated for each location, they are passed through a Non-Maximum Suppression (NMS) or Circular NMS module to collect final predictions.
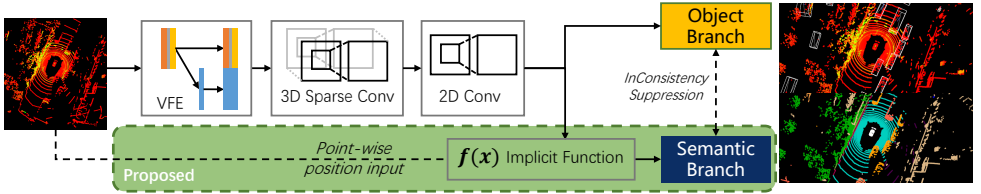
Figure 2: The overall structure of the proposed joint 3D object detection and semantic segmentation framework

3.2 Semantic Rendering Branch

Inspired by [19], we propose a semantic rendering branch that learns a continuous function to generate semantic distribution prediction for each input position and reduce rasterization error. The semantic branch prevents granularity loss by using an implicit function. This process is achieved by using a lightweight MLP module with inputs coming from both point positions and local voxel features. Different from PointRend which uses both coarse and fine-grained features, we only use the original global feature map from the convolutional backbone since our base framework is a single-stage detector instead of the two-stage MaskRCNN[16] used in [19]. In our experiment, the MLP of the semantic branch consists of 4 layers with 256, 128, 64, and 32 channels, respectively.

Given a query point $q = (x_q, y_q, z_q)$ from the point cloud input and backbone feature map $\mathbf{M} \in \mathbb{R}^{C \times D \times H \times W}$, we first find the grid position $(i, j, k) \in \mathbb{N}^{D \times H \times W}$ in the feature map where the query point lies, then the semantic distribution $s_q \in \mathbb{R}^S$ with S classes of the point are generated by the MLP module $f : \mathbb{R}^{3+C} \to \mathbb{R}^S$ formulated as

$$s_q = \text{softmax}(f([x_q - cx_i \ y_q - cy_j \ z_q - cz_k \ \mathbf{M}_{\cdot ijk}])) \tag{1}$$

where $(cx_i, cy_j, cz_k) \in \mathbb{R}^{D \times H \times W}$ represents the real-world center of the grid in the feature map indexed by (i, j, k). If the query point is outside of the voxel grid, then the voxel closest to the point is selected. This module f will be supervised by the point-wise semantic labels from the dataset.

Since only the position of the query point is fed into the function, unlike the method used in [51], the semantic branch can predict points that are not in the original point cloud. The decoupling between feature extraction and querying is beneficial when semantics are required to be extrapolated between points, which will be further discussed in Section 4.4. Note that in our backbone, the 3D voxel features are flattened and processed by a 2D convolutional network, therefore in our case $D = 1$. The proposed branch can be applied directly to a 3D voxel grid if the backbone framework produces a 3D tensor to the detection head.

With the network structure defined above, we define the loss function as

$$L = \alpha_{cls} L_{cls} + \alpha_{reg} L_{reg} + \alpha_{sem} L_{sem} \tag{2}$$

where we use focal loss with Gaussian kernels for the classification loss L_{cls} on the heatmap, and L_1 loss for the regression loss L_{reg} on box parameters, both of which are inherited from the backbone method we adopted from [44]. For semantic supervision, we use a combination of Lovász oss [3] and weighted cross-entropy loss for the semantic loss L_{sem}. In our experiment, the parameters for the weight loss are $\alpha_{cls} = \alpha_{sem} = 1, \alpha_{reg} = 0.25$.

3.3   Panoptic Post-processing

After bounding boxes and point-wise semantic labels are generated, panoptic segmentation results can be generated by assigning instance ids of the boxes to the points inside them.

Aside from panoptic label generation, object bounding boxes and semantic labels can be used to mutually recover the error in predictions. Inspired by the strategy introduced in [47], we developed a novel procedure named **InConsistency Suppression** (ICS) to first fix inconsistent labels of the bounding boxes using a estimated label from point-wise semantic outputs and then inconsistent points will be fixed. The algorithm is described in the pseudocode (Algorithm 1) below. The inputs for the procedure are the collection of predicted bounding boxes B and semantic point clouds P. Each box $b \in B$ has an object classification $K(b)$ with confidence score $s(b)$, and each point $p \in P$ has semantic classification $K(p)$ with confidence score $s(p)$. b also denotes the area inside the box. In the procedure description, K_{th} denotes the thing categories.

Algorithm 1 InConsistency Suppression

1: **procedure** ICS(B, P)   ▷ Fix inconsistent labels in-place
   B: labeled bounding boxes, P: labeled point cloud
   c_α, c_γ: tunable parameters with default value 1
   m_p: score margin for overriding point label

2:     Sort B descendingly by score
3:     **for** $i = 1 \ldots |B|$ **do**   ▷ loop for fixing box label
4:         $P_i \leftarrow \{p \in P \cap b_i | K(p) \neq K(b_j) \, \forall j < i, p \in b_j\}$ ▷ select inconsistent points
5:         **for** $k \in K_{th}$ **do**   ▷ for each semantic class k
6:             $P_i^k \leftarrow \{p \in P_i | K(p) = k\}$ ▷ select points with label k
7:             $\alpha_k \leftarrow |P_i^k|/|P_i|$ ▷ count criterion
8:             $\beta_k \leftarrow \Sigma_{p \in P_i^k} s(p)/|P_i^k|$ ▷ score criterion
9:             $\gamma_k \leftarrow 1 + s^{c_\gamma}(b_i) \cdot \mathbb{1}_{k=K(b_i)}$ ▷ correctness criterion
10:        $k^* \leftarrow \arg\max_{k \in K_{th}} \alpha_k \beta_k \gamma_k$ ▷ select the best class using the three criteria
11:        **if** $k^* \neq K(b_i)$ **then**   ▷ if the best class is not the predicted one
12:            **if** $\exists j > i$ s.t. $K(b_j) = k^*$ **then**
13:                Swap $K(b_i)$ and $K(b_j)$ ▷ swap label with a box with lower score
14:            **else**
15:                $K(b_i) \leftarrow k^*$ ▷ override the box label by the best class
16:    **for** $i = |B| \ldots 1$ **do**   ▷ loop for fixing point labels
17:        **for** $p \in \{p \in P \cap b_i | s(p) < s(b_i) - m_p\}$ **do**   ▷ for each point with low score
18:            **if** $\exists j < i$ s.t. $p \in b_i \cap b_j$ **then**
19:                **Continue**   ▷ ignore boxes with large overlap
20:            $K(p) \leftarrow K(b_i)$ ▷ override the point label by box label

---

4   Experiment Results

Experiments are conducted on the nuScenes[4] dataset with nuScenes-lidarseg extension. This dataset is selected because it provides labels for both 3D object detection and point cloud semantic segmentation. The nuScenes dataset contains various labels for different tasks, including 28k synchronized frames with multiple cameras, one lidar and multiple

Table 1: Comparison of semantic segmentation performance using the nuScenes validation dataset. For all metrics the higher the better, the best one is shown in boldface.

Method	Source	mIoU	fwIoU	Car	Truck	Bus	Trailer	Construction Vehicle	Pedestrian	Motorcycle	Bicycle	Traffic Cone	Barrier	Drivable Area	Other flat	Terrain	Man-made	Vegetation
SalsaNext [ ]	[ ]	58.8	82.8	81.0	65.7	77.1	38.3	18.4	52.8	47.5	4.7	43.5	56.6	94.2	60.0	**70.3**	81.2	80.5
MinkNet42 [ ]	[ ]	60.8	82.7	77.1	62.2	77.4	42.5	23.0	55.6	55.1	8.3	50.0	63.1	94.0	67.2	68.6	**83.7**	80.8
(AF)²-S3Net [ ]	[ ]	62.2	83.0	80.0	67.4	82.3	42.2	20.1	59.0	62.0	12.6	49.0	60.3	94.2	68.0	68.6	82.9	**82.4**
VIN (ours)	-	73.7	84.3	**87.0**	82.6	91.7	63.2	49.9	79.7	82.2	**46.2**	**59.4**	74.7	94.5	67.1	68.5	**83.7**	81.2
VIN + ICS (ours)	-	**73.8**	84.4	**87.0**	82.8	91.7	**63.7**	50.4	79.8	**82.5**	46.1	**59.4**	74.7	94.5	67.1	68.5	**83.7**	81.2
VIN (seg only)	-	72.0	**86.3**	86.6	82.0	**91.8**	61.8	45.9	**80.3**	68.6	30.7	45.2	**75.4**	**95.8**	**73.8**	72.6	**85.3**	83.3

radars for training, and 6k samples each for validation and testing. The nuScenes-lidarseg extension provides point-wise semantic labels of 15 categories in total. It's a challenging dataset with adverse environment scenarios including dark nights and rainy days.

Our backbone method is CenterPoint[44] with a voxel size of 0.1m and without CBGS[50] or other test-time augmentation due to our hardware limitation. Our model is trained with 2 NVidia V100 16G GPUs, using AdamW optimizer with cyclic learning rate scheduling starting from $1e^{-4}$ and weight decay of 0.01. Please refer to [44] and our code for details.

Selected qualitative results are shown in Figure 3. Our method is able to generate precise point-wise semantic labels while preserving the ability to predict accurate 3D bounding boxes. However, our method still suffers from the problem of ambiguity due to the sparsity of the lidar point cloud, common for all lidar-based detection or segmentation algorithms.

4.1 Quantitative Results

Semantic Segmentation Performance For segmentation performance, our method is compared against state-of-the-art methods on the nuScenes lidar segmentation track. The main metrics used for comparison are the intersection-over-union (IoU) for each category. Mean IoU (mIoU) and frequency-weighted IoU (fwIoU) numbers are also provided by the nuScenes benchmark. The results are presented in Table 1, compared with other methods, our approach performs significantly better in most "things" categories except for the traffic cones. Our backbone method does not preserve details in each voxel, thus it may not work well with small objects. On the other hand, existing methods outperform in many "stuff" categories, because our network needs to focus on objects to learn object detection well. Overall, our method outperforms the state-of-the-art methods by about 9.4 percentage points in mIoU.

3D Object Detection Performance: For the lidar-only object detection performance, we select a few other state-of-the-art methods on the nuScenes detection track. The overall performance is measured by mean average precision (mAP) and the NDS score proposed by nuScenes to capture not only precision performance, but also errors of other target properties. From the results shown in Table 2, we found that although our method suffers performance loss when adding a semantic branch, we still achieve good performance in most categories. This indicates that our method can produce semantic segmentation labels without much loss of detection performance. The most prominent gaps come from the "trailer", "bicycle" and "construction vehicle" categories, all of which are rare in the dataset. It's hard for the network to maintain the same performance when it learns to solve an additional segmentation task. Label balancing techniques can be applied in the future to solve the problem.

Panoptic Segmentation Performance Panoptic segmentation is a side product of our method, however, it can be used to evaluate the combined performance of detection and segmentation. The metrics proposed in [18] is leveraged in our experiment, which includes
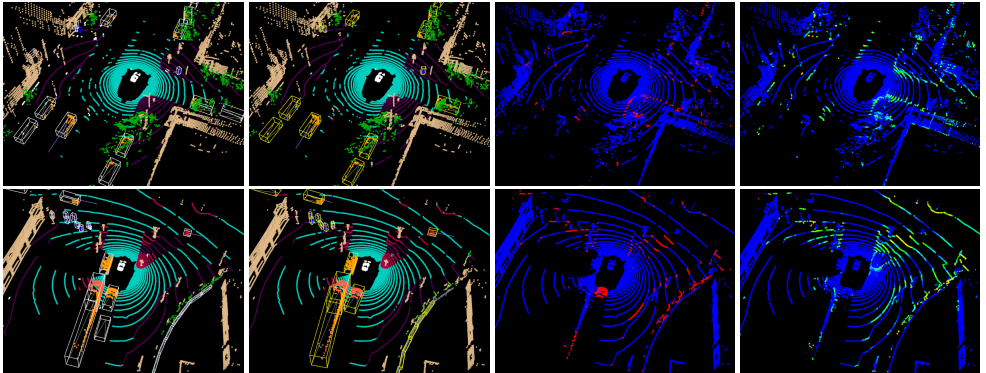
Figure 3: Qualitative results of VIN. First column: ground-truth semantic labels and bounding boxes; Second column: estimated semantic labels and bounding boxes; Third column: error points of semantic segmentation (labeled in red); Forth column: confidence score of the semantic segmentation (colors vary from light green to dark blue for scores from low to high). Best viewed in color and zoomed in.

Table 2: Comparison of (lidar-only) 3D object detection performance on the nuScenes validation dataset. All metrics are the higher the better, the best one is underlined.
Abbreviations: *CV - construction vehicle, TC - traffic cone, Ped - Pedestrian, Motor - Motorcycle, R - Reproduced*

Method	Source	mAP	NDS	Car	Truck	Bus	Trailer	CV	Ped	Motor	Bicycle	TC	Barrier
PointPillars [□]	[□]	28.2	46.8	75.5	31.6	44.9	23.7	4.0	49.6	14.6	0.4	8.0	30.0
3DSSD [□]	[□]	42.6	56.4	81.2	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
CenterPoint [□]	[□]	56.6	65.0	84.6	54.7	66.0	32.3	15.1	84.5	56.9	38.6	67.4	66.1
CenterPoint	R	50.7	60.2	82.6	50.3	63.8	30.8	12.0	79.9	43.9	22.3	60.8	60.4
VIN (Ours)	-	45.2	57.0	82.1	48.3	61.3	18.7	3.5	73.4	50.4	2.2	56.1	55.9
VIN + ICS (Ours)	-	46.3	57.6	82.1	48.1	61.1	22.7	7.2	74.3	50.4	4.1	56.6	56.4

Panoptic Quality (PQ), Segmentation Quality (SQ) and Recognition Quality (RQ). SQ and RQ are analog to point-averaged and instance-averaged IoU. We also adopt the replaced Panoptic Quality ($PQ^{\dagger}$) metric proposed by [□]. The performance of our algorithms is compared with three baseline methods in Table 3. Our algorithms outperform these baseline methods in most metrics except SQ. A possible fix is adding penalty terms for "stuff" points inside object boxes and "thing" points outside object boxes in future work, thus explicitly let the network distinguish between the environment and objects.

4.2   Ablation Study

Inconsistency Suppression The proposed ICS procedure is used to eliminate inconsistency between bounding boxes and semantic labels, which is beneficial for downstream perception modules. This is demonstrated in Table 1, 2 and 3. For fixing semantic and panoptic segmentation labels, we set $m_p = 0.1$ in Algorithm 1 and achieve slightly better results. On the other hand, there is a large effect by ICS on detection performance, especially in the 'barrier', 'construction vehicle' and 'bicycle' categories, which indicate that ICS is effective. The improvement comes from using point predictions to help determine the object classification.

Weakly supervised segmentation Weak supervision can help to reduce label efforts when building datasets. Thanks to the nature of the implicit function representation, it doesn't require the full point cloud for supervision. Our method can handle weakly su-

Table 3: Comparison of panoptic segmentation performance on nuScenes validation set. All metrics are the higher the better, the best ones are highlighted in boldface.

Method	Source	PQ	PQ†	RQ	SQ	PQth	RQth	SQth	PQst	RQst	SQst	mIoU
Cylinder3D[■] + SECOND[■]	[■]	40.1	48.4	47.3	**84.2**	29.0	33.6	**84.4**	58.5	70.1	83.7	58.5
Cylinder3D[■] + PointPillars[■]	[■]	36.0	44.5	43.0	83.3	23.3	27.0	83.7	57.2	69.6	82.7	52.3
DS-Net[■]	[■]	42.5	51.0	50.3	83.6	32.5	38.3	83.1	59.2	70.3	**84.4**	70.7
VIN (ours)	-	**51.7**	**57.4**	61.8	82.6	**45.7**	53.7	83.6	**61.8**	**75.4**	80.9	73.7
VIN + ICS (ours)	-	**51.7**	**57.5**	**61.9**	82.6	**45.7**	**53.8**	83.6	**61.8**	**75.4**	80.9	**73.8**

Table 4: Comparison of detection and segmentation performance on nuScenes validation set with different supervision levels. The label percentages denote the ratio of points used in training the semantic branch. All metrics are higher the better.

Method	mAP	NDS	mIoU	fwIoU	mIoUth	mIoUst	PQ	PQ†	PQth	PQst
Ours (full supervision)	45.2	57.0	73.7	84.3	71.3	78.3	51.7	57.4	45.7	61.8
Ours (10% label)	45.7	57.4	74.4	84.8	72.3	78.0	52.7	58.2	46.3	63.4
Ours (1% label)	44.7	56.6	72.6	84.6	69.5	77.9	50.4	55.9	42.8	63.1
Ours (0.1% label)	45.1	57.0	72.3	84.5	69.1	77.5	50.6	56.0	43.2	62.9
Ours (0.02% label)	44.5	56.6	70.6	83.8	67.1	76.5	49.0	54.5	41.3	61.8

pervised segmentation tasks by feeding fewer labels during the training process. A key difference between our proposed method and single-task segmentation methods is that our method only get auxiliary information from the bounding box labels. The efficacy of our method is illustrated in Table 4. It can be seen that our method achieves robust segmentation performance even with just 0.1% semantic labels. However, the performance is not increased monotonically with the amount of available labels, which is unexpected and requires further experiments to explain.

4.3 Inference Efficiency

A major motivation to combine detection and segmentation in a single network is to reduce inference time and achieve better real-time efficiency when deployed. We compare our method with baseline CenterPoint in terms of inference time. Measured on a single NVidia 2080Ti graphics card, the original CenterPoint network with 0.1m voxel size achieves 6.0 FPS, while our method with the same voxel size and backbone configuration achieves 5.9 FPS but with the additional semantic segmentation results. This is much better than having a separate segmentation implementation. For example, a state-of-the-art method [■] alone takes 0.27 seconds (reported on nuScenes leaderboard, on an NVidia Tesla V100). When deploying our method on a vehicle, further optimization, including network distillation, refactoring using a highly efficient inference framework (e.g. TensorRT) can be implemented.

4.4 Utilizing the Implicit Function

Thanks to the fact that our semantic branch is merely based on the feature map produced by the convolutional backbone, semantics queries can be conducted at arbitrary positions in the space, as mentioned in Section 3.2.

Here we demonstrate two use cases of the query ability, semantic prediction on downsampled point cloud and dense semantic map generation. The experiment results for the former case are reported in table 5. The down-sampling strategy is useful for reducing onboard latency or helping generate off-board labels for autonomous vehicles. In this experiment, only partial points (down-sampled from the original point cloud) are fed into the trained detection backbone and the semantics of the remainder of the original point cloud is
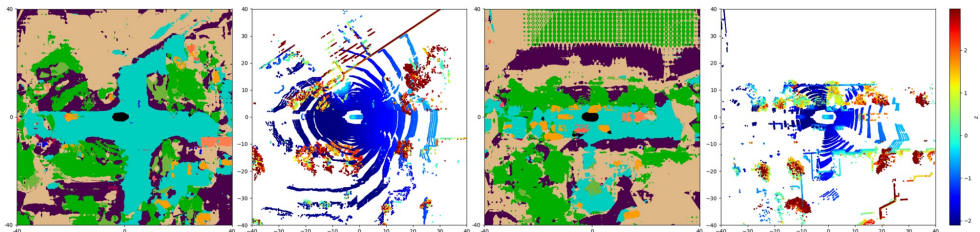
Figure 4: Two examples of generated dense semantic map on the Nuscenes dataset. The input point cloud is colored by the height. Please refer to Figure 3 for the color legend.

Table 5: Comparison of segmentation performance with partial point cloud input on Nuscenes. The best ones are highlighted in boldface.

Method	mIoU (random sample)			mIoU (beam sample)		
	100%	75%	50%	32 beams	24 beams	16 beams
Ours	73.7	**73.3**	**72.5**	73.7	**72.5**	**70.2**
Nearest Neighbor	73.7	73.1	72.2	73.7	64.6	55.9
(Inference time)	*169ms*	*133ms*	*104ms*	*169ms*	*125ms*	*96ms*

predicted by either the semantic branch or nearest neighbor querying. It can be concluded from the experiment that our model can better capture the semantics in region with no lidar measurements when the original point cloud is down-sampled to reduce the inference time.

On the other hand, it's also possible to generate a dense semantic map by querying semantics at grid points with the semantic branch. Figure 4 shows the semantic prediction of the scenes at certain height (the bilinearly interpolated height of the point cloud is used to create the figure). This dense map can be used as a bird's eye view semantic map, which standalone models have been proposed in the literature (e.g. [52]) to estimate. It's useful for finding the semantics boundary for different areas and better interpreting the performance of the deep learning model.

5   Conclusion

In this paper, a novel framework for joint 3D object detection and semantic segmentation using lidar point clouds is proposed, which is more efficient than two separate implementations. A semantic branch learning implicit representation of spatial semantic properties is proposed. This modification can be applied to any voxel-based 3D object detectors. Experiments using the nuScenes dataset confirm the efficacy and efficiency of our algorithm by comparing it with state-of-the-art methods on semantic and panoptic segmentation tasks.

Currently, our method still struggles to detect and segment partially occluded objects and small objects with few lidar reflections, which are the strengths of camera images, but lidar-based algorithms work better in dazzling and dark scenarios. Future work can incorporate image data to improve the accuracy of both detection and segmentation of the framework.

Acknowledgements

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[5] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[8] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. *arXiv preprint arXiv:2102.04530*, 2021.

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[10] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5828–5839, 2017.

[12] Yiqun Dong, Yuanxin Zhong, Wenbo Yu, Minghan Zhu, Pingping Lu, Yeyang Fang, Jiajun Hong, and Huei Peng. Mcity data collection for automated vehicles study. *arXiv preprint arXiv:1912.06258*, 2019.

[13] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

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

[15] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. *arXiv preprint arXiv:2102.08945*, 2021.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. *arXiv preprint arXiv:2011.11964*, 2020.

[18] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[19] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[20] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[21] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[23] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. *arXiv preprint arXiv:2104.02246*, 2021.

[24] Jilin Mei, Biao Gao, Donghao Xu, Wen Yao, Xijun Zhao, and Huijing Zhao. Semantic segmentation of 3d lidar data in dynamic scene using semi-supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2496–2509, 2019.

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[26] Gregory P Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[27] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017.

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[30] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

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

[32] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[34] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

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

[36] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[37] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[38] Ozan Unal, Luc Van Gool, and Dengxin Dai. Improving point cloud semantic segmentation by learning 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2950–2959, 2021.

[39] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2020.

[40] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[41] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

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

[43] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020.

[44] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *arXiv preprint arXiv:2006.11275*, 2020.

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

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[47] Yuanxin Zhong, Sijia Wang, Shichao Xie, Zhong Cao, Kun Jiang, and Diange Yang. 3d scene reconstruction with sparse lidar data and monocular image in single frame. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 11 (07-11-01-0005):48–56, 2017.

[48] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020.

[49] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[50] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.

[51] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. *arXiv preprint arXiv:2011.10033*, 2020.