A Versatile Multi-View Framework for LiDAR-based 3D Object Detection with Guidance from Panoptic Segmentation

Hamidreza Fazlali   Yixuan Xu   Yuan Ren   Bingbing Liu*

Abstract

3D object detection using LiDAR data is an indispensable component for autonomous driving systems. Yet, only a few LiDAR-based 3D object detection methods leverage segmentation information to further guide the detection process. In this paper, we propose a novel multi-task framework that jointly performs 3D object detection and panoptic segmentation. In our method, the 3D object detection backbone in Bird's-Eye-View (BEV) plane is augmented by the injection of Range-View (RV) feature maps from the 3D panoptic segmentation backbone. This enables the detection backbone to leverage multi-view information to address the shortcomings of each projection view. Furthermore, foreground semantic information is incorporated to ease the detection task by highlighting the locations of each object class in the feature maps. Finally, a new center density heatmap generated based on the instance-level information further guides the detection backbone by suggesting possible box center locations for objects. Our method works with any BEV-based 3D object detection method and, based on experiments on the nuScenes dataset, it provides significant performance gains. Notably, the proposed method based on a single-stage CenterPoint 3D object detection network achieve state-of-the-art performance on nuScenes 3D Detection Benchmark with 67.3 NDS.

1. Introduction

Over the past few years, there has been remarkable progress in autonomous vehicles (AVs) vision systems for understanding complex 3D environments [8, 9, 31]. 3D object detection is one of the core computer vision tasks that empowers AVs for robust decision-making. In this task, each foreground object, such as a car, a pedestrian, etc., needs to be accurately classified and localized by a 3D bounding box with 7 degrees of freedom (DOF), including the 3D box center location (x, y, z), size (l, w, h), and yaw angle (α).

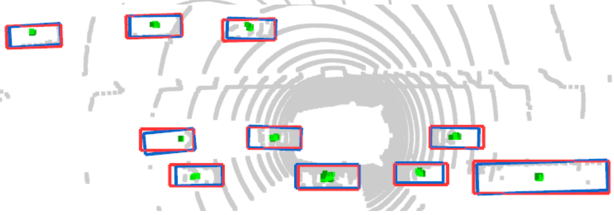

Figure 1. The proposed center density heatmap (colored in green) guides the detection head toward possible box center regions on the BEV plane. The blue and red boxes represent predictions and ground truth, respectively. Best viewed in color.

LiDAR-based 3D object detection methods rely on different strategies for 3D point cloud data representation. Some of these detection methods [13, 17, 28] are categorized as point-based methods. These methods directly process the raw point cloud to extract useful information. While they usually achieve high accuracy, their computational cost is significant. The other subset of LiDAR-based 3D object detection methods are known as grid-based methods [9, 18, 26, 31]. These methods transform the unordered point cloud into regular 3D volumetric grids, *i.e.*, voxels or pillars, and extract discriminative features from the points inside each gird cell. The extracted features are further processed by 2D or 3D Convolutional Neural Networks (CNN). Although point sub-sampling helps grid-based methods to be computationally efficient, some information is lost during projection and discretization [27].

Many of the points in each LiDAR scan represent the background region, including drivable surface, sidewalk, vegetation, etc. Feeding all this information to a 3D object detection algorithm without providing extra clues, *e.g.*, semantic information, makes the recognition and localization process challenging. Several works [17, 22, 28] have exploited a binary or semantic segmentation model for filtering background points or providing extra semantic features that can guide the proposal generation or detection process.

Inspired by this notion of providing guidance, we leverage 3D panoptic segmentation as an auxiliary task for guiding and further improving the performance of Bird's-Eye-View (BEV) based 3D object detection algorithms. A 3D panoptic segmentation method predicts the semantic class

[*] All authors are with Huawei Noah's Ark Lab, Canada at the time of writing. {hamidreza.fazlali1, richard.xu2, yuan.ren3, liu.bingbing}@huawei.com

label and performs instance-level segmentation for each point in the 3D space, both of which are useful as guidance signals for detecting objects. In addition, guiding a BEV-based detection model with features learned from a Range-View (RV) based network can reduce the sparsity of features representation in BEV projection. We validate these ideas by training a BEV-based 3D object detection method in conjunction with an RV-based 3D panoptic segmentation method. More specifically, the BEV-based detection backbone is supplemented with additional RV features extracted from the panoptic segmentation backbone, providing a rich set of multi-view information to aid the detection. Moreover, we exploit the semantic labels of the foreground objects estimated by the panoptic segmentation network to refine the 3D object detection backbone features. Finally, a center density heatmap in the BEV plane is designed based on the instance-level information obtained from the panoptic segmentation, highlighting regions that contain box centers of objects. In conjunction, the augmented backbone features, foreground semantic labels, and the center density heatmap guide the detection model towards a more accurate 3D box recognition and localization. We will describe our multi-task framework based on the single-stage CenterPoint 3D object detection method [30], but later in the experimental results section, we will quantitatively demonstrate that our approach can help any existing BEV-based 3D object detection method.

Our contributions can be summarized into four-fold. (1) We propose a multi-task framework that jointly learns 3D panoptic segmentation and 3D object detection for improving the 3D object recognition and localization. To the best of our knowledge, this is the first framework that leverages both the semantic- and instance-level information concurrently for improving 3D object detection. (2) The framework is also designed to be easily attached to any BEV-based object detection method as a plug-and-play solution to boost its detection performance. (3) With experiments conducted on the nuScenes dataset [2], which includes both the panoptic and 3D box information, we validate the effectiveness of our method with different BEV-based 3D object detection methods. (4) We conduct ablation studies to further examine the usefulness of each added component for performance improvement.

2. Related Work

3D Object Detection with Point-based Methods. The point-based methods take in the unordered sets of 3D point cloud and rely on PointNet [14] or PointNet++ [15] for feature extraction. FPointNet [13] uses the 2D object proposals from camera images to filter the point cloud and then uses PointNet for 3D object detection based on the proposal regions. Its performance suffers as proposal regions generated from RGB images lack accurate 3D information. PointRCNN [17] addresses this problem by first segmenting the foreground points using PointNet++ and then refining the proposals using the segmentation features. STD [28] uses PointNet++ for proposal generation and then further densifies the point features within each proposal using a pooling strategy. Generally, the point-based methods have a larger receptive field compared to the grid-based methods; however, their computational complexity is very high. [16].

3D Object Detection with Grid-based Methods. These methods divide the 3D space into volumetric grids known as voxels, so they can be processed by 2D or 3D CNN. Earlier methods encode each voxel with some hand-crafted features. PIXOR [27] encodes each voxel based on the occupancy and reflectance of points. Complex-YOLO [19] encodes each grid cell with the maximum height, maximum intensity, and normalized point density. In order to extract a more useful and richer set of features, the authors of VoxelNet [31] designed a Voxel Feature Encoder (VFE) layer to leverage the power of deep learning for voxel feature learning and then used a 3D CNN for its det1ection backbone. PointPillars [9] reduces the number of voxels to one along the height dimension, improving both the inference time and detection accuracy. In [1], 3D object detection is done in RV using a range-conditioned dilation (RCD) layer to address the problem of object scale change in the range image. To take advantage of both BEV and RV point cloud representations, CVCNet [3] uses Hybrid-Cylindrical-Spherical (HCS) voxels. RSN [22] is another multi-view fusion method that first performs binary segmentation on the RV and then applies sparse convolutions on the foreground voxels with the learned RV features to detect objects. To address the imbalance of voxel sparsity between object classes, [4] considers a limited number of non-empty object voxels as hotspots and the detection head predicts these hot spots and the corresponding boxes. Following the success of CenterNet [6] in 2D object detection, CenterPoint [30] uses an anchor-free 3D object detection head (center-head) that first detects the center point of each object in the BEV plane and subsequently regresses the bounding box dimensions.

3. Preliminaries

Cluster-free 3D Panoptic Segmentation (CPSeg). CPSeg [10] uses a dual-decoder architecture to conduct panoptic segmentation without generating object proposals or using clustering algorithms. The backbone takes in the RV representation of the LiDAR point cloud and provides multi-scale feature maps. The semantic decoder provides semantic labels, while the instance decoder predicts the object mass centroid as instance embedding for each point. Different from the semantic decoder, which only utilizes encoded feature maps, the instance decoder benefits from the additional information of computed surface normals.

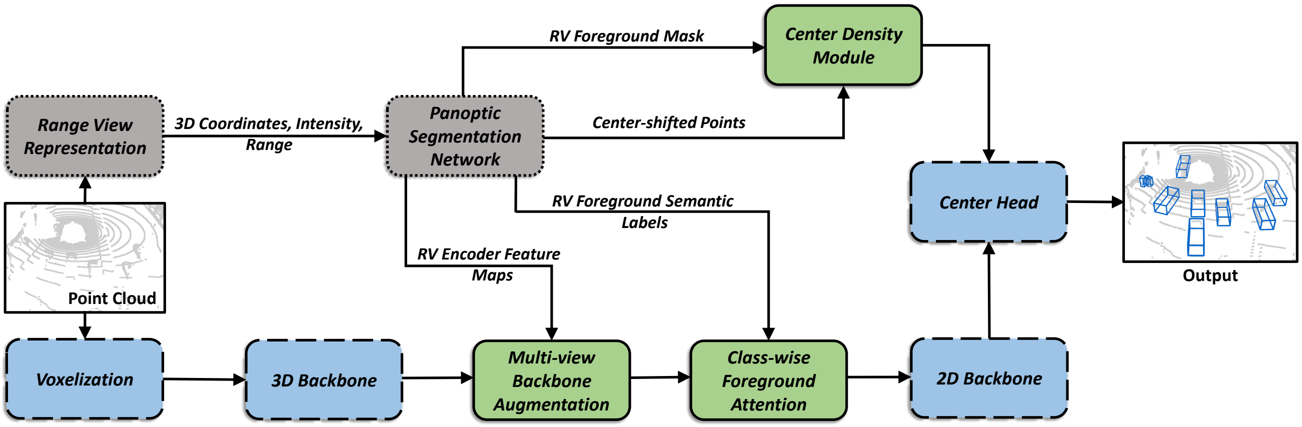
Range View Representation
3D Coordinates, Intensity, Range
Panoptic Segmentation Network
RV Foreground Mask
Center Density Module
Center-shifted Points
RV Foreground Semantic Labels
RV Encoder Feature Maps
Point Cloud
Center Head
Output
Voxelization
3D Backbone
Multi-view Backbone Augmentation
Class-wise Foreground Attention
2D Backbone

Figure 2. Block diagram of the proposed method. The **gray** blocks (dotted border) represent the CPSeg model [10] and the **blue** blocks (dashed border) represent the single-stage CenterPoint model [30]. The **green** blocks (solid border) are the proposed modules for combining the 3D panoptic segmentation and 3D object detection under a multi-task framework. Best viewed in color.

Then, the cluster-free instance segmentation head dynamically groups points with similar instance embedding as pillars in BEV, and objects are segmented by calculating the connectivity between pillars through a pairwise embedding comparison matrix.

CenterPoint 3D Object Detection. Originally, CenterPoint is a two-stage 3D object detection method, where bounding boxes are regressed in the first stage and further refined in the second stage. Specifically, the second stage takes in the backbone feature maps in the BEV plane and considers information at locations where four sides of the first-stage bounding box are located. During training, the model estimates a center heatmap for each object class and other regression targets, including the box center offset, size, angle, and velocity. In the estimated center heatmap for each class, the local-maxima values represent the confidence scores. Their locations on the map are used to estimate the other regression targets. Overall, CenterPoint removes the need for anchor boxes in 3D object detection, which are originally inherited from the 2D object detection and are challenging to fit in the 3D space.

4. Proposed Approach

The block diagram of the proposed multi-task framework is shown in Figure 2. This method receives raw LiDAR point cloud data as input and outputs both the 3D panoptic segmentation and object detection results. For panoptic segmentation, we use CPSeg for its state-of-the-art and real-time performance. We made some architectural modifications to the CPSeg model for speeding up the proposed multi-task framework. The details of these changes are described in the supplementary materials. Furthermore, instead of predicting mass-center offsets as in the original CPSeg, the instance segmentation head in our modified CPSeg provides the 3D box-center offsets, useful for guiding the 3D object detection. Other than center offsets, CPSeg also provides its encoder feature maps and generated foreground semantic predictions to aid the detection model, as shown in Figure 2.

For 3D object detection, we chose CenterPoint for its performance superiority compared to the anchor-based 3D object detection methods. We use the single-stage CenterPoint method based on VoxelNet backbone [6]. The CenterPoint consists of two main components: the backbone and the detection head. As shown in Figure 2, the detection backbone consists of the voxelization module that divides a point cloud into volumetric voxel grids, the 3D CNN backbone that learns 3D structural features, and a 2D CNN backbone that further processes the learned features in BEV. The detection head includes a group of class-specific center heads that predict center heatmap and other bounding box regression targets. We remove the second stage refinement process within the CenterPoint in our multi-task framework as information from panoptic segmentation is found to be sufficient in guiding the detection head to accurate detections. Moreover, in the Experiments section, we will show that our multi-task framework can easily work with any existing BEV-based 3D object detection method by simply replacing the detection backbone and head.

The 3D object detection and panoptic segmentation methods in our framework are trained jointly. First, the backbone of the detection network is augmented with the addition of a rich set of RV features from the panoptic segmentation encoder. Furthermore, foreground semantic labels and instance box center offsets estimated by the panoptic segmentation network are also injected to guide the 2D backbone and detection head to attend to the locations of potential objects and their centers, respectively. As illustrated in Figure 2, the integration takes place in the multi-view backbone augmentation, class-wise foreground atten-

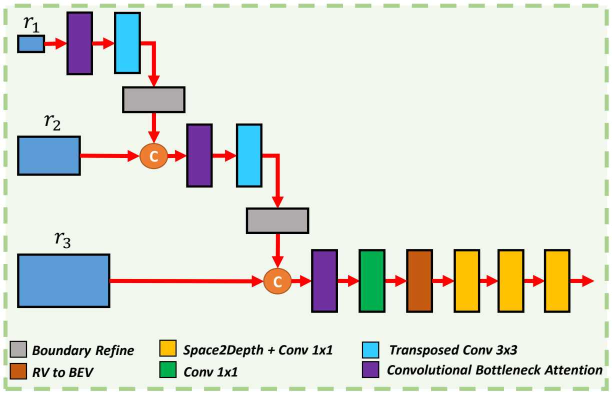

Figure 3. Cascade feature fusion module. The selected RV feature maps are fused and the projected to BEV plane and then downsampled to match the detection backbone feature map resolution. Best viewed in color.

tion, and center density heatmap modules. More details about these blocks are covered in the following subsections.

4.1. Multi-View Backbone Augmentation

RV and BEV representations of point clouds enable the design of efficient 3D perception models. Feature extractors of state-of-the-art panoptic segmentation models commonly rely on range view [11, 20], while most well-known 3D detection methods operate on the BEV plane [4, 9, 26, 30]. However, each form of projection has its strengths and weaknesses. For example, features representation in RV are denser and align with the LiDAR scan patterns. Thus, small objects, *e.g.*, pedestrians, traffic cones, motorcycles, and bicycles, are more visible and easier to be detected and classified. However, determining the sizes and boundaries of crowded or distant objects is difficult in RV due to occlusions and size variations. On the other hand, BEV avoids the issues presented in RV, but its sparse and coarse representations make it challenging to detect smaller objects.

Similar to [3, 5], we attempt to leverage the strength of each view to improve the performance of 3D object detection. Here, we introduce the concept of feature weighting to combine multi-view features, by adaptively weighting each feature value in RV and BEV maps based on its perceived importance in boosting the detection performance. The multi-view backbone augmentation is composed of two steps. First, RV feature maps from the segmentation model backbone are projected to the BEV plane and further downsampled to match the resolution of the extracted BEV feature maps in the detection backbone. Then, the RV- and BEV-based feature maps are fused using a proposed space-channel attention module, which weights each feature map based on its usefulness for the detection task. These two steps are elaborated below.

4.1.1 Cascade RV Feature Fusion Module

The RV feature maps generated in the panoptic segmentation backbone can help augment the detection backbone, which operates in the BEV plane, in detecting smaller objects that are otherwise not properly represented.

A cascade feature fusion module, as shown in Figure 3, processes multi-scale RV feature maps and prepares them for the detection backbone augmentation. The coarser RV feature maps, r_1 and r_2, are obtained from intermediate layers of the CPSeg encoder, which contain contextual RV information that can benefit multiple tasks. On the other hand, high-resolution feature maps, r_3, encode additional geometric information of point cloud in the RV plane to emphasize the locations and presence of objects. More specifically, the learned geometric features extracted from surface normal vectors originated in CPSeg are concatenated with the 3D Cartesian coordinates associated with each point.

In the proposed cascade feature fusion module, starting from the coarsest-scale, feature maps r_1 are first processed by a Convolutional Bottleneck Attention Module (CBAM) [25]. This module is responsible for adaptive feature refinement along space- and channel-dimensions. Then, the resulted feature maps are up-sampled by a factor of 2 using a 3×3 Transposed Convolution layer and passed to the boundary refinement layer to reduce the upsampling artifacts. These up-sampled features are then concatenated to higher-resolution feature maps, r_2, and the same previously mentioned operations are applied on the concatenated feature maps. After the features are concatenated at the highest-resolution scale with r_3, they are passed through a CBAM and a 1×1 convolution layer, and subsequently projected to the BEV plane. Finally, a sequence of down-sampling blocks reduces the resolution of the features to the specifications of the detection backbone. Each $Space2Depth$ operation reduces the spatial resolution of the feature maps by half and doubles the channel number, and the accompanied 1×1 convolution layer compresses the feature maps along the depth dimension.

4.1.2 Attention-based RV-BEV Feature Weighting Module

After the cascade RV feature fusion module, feature maps extracted from two different views are concatenated. An attention-based weighting module is crucial so that the model learns to adaptively focus on feature values from a particular view given the scenario. To this end, a modified CBAM [25] is proposed here to provide useful weighting for RV and BEV features.

The diagram of this module is shown in Figure 4. The attention-based RV-BEV weighting module takes in the concatenated feature maps from both views, and distributes them to channel- and space-attention streams. The channel-

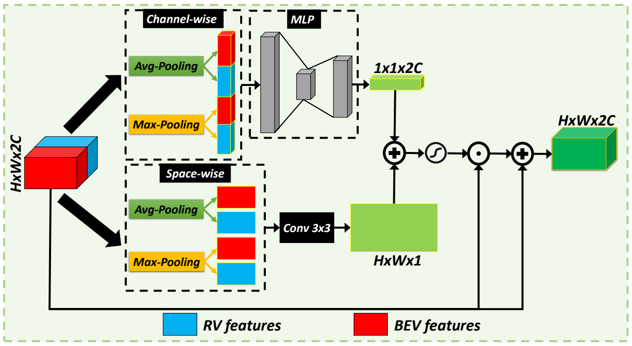

Figure 4. Attention-based RV-BEV feature weighting module. The model learns to highlight the best feature values among the RV and BEV representations along space and channel dimensions. Best viewed in color.

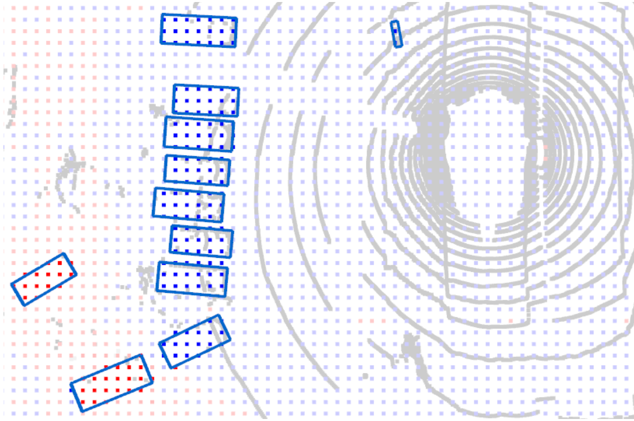

Figure 5. Visualization of the attention map generated by the attention-based RV-BEV feature weighting module. Regions in red and blue indicate locations where BEV and RV features are deemed more significant with higher weights, respectively. Boxes in blue are ground truth object bounding boxes. Colors in background are shaded for better visualization. Best viewed in color.

attention stream applies max-pooling and average-pooling across x- and y-axis of the BEV and RV feature maps, resulting in 4 sets of feature vectors, which are then fed to a multilayer perceptron (MLP) to learn an attention value for each channel in RV and BEV feature maps. The space-attention stream applies the same pooling operations depth-wise to obtain 4 feature maps, which are then fed to a 3×3 convolution layer with a dilation rate of 3 to learn spatial attention values. The resulted feature map from the space-attention stream and the feature vector from the channel-attention stream are both broadcasted to the same shape, summed, and passed through a $sigmoid$ activation function to generate an attention map. By applying this attention map to the input feature maps, the feature values from either views that help the detection task are highlighted. This enables the detection backbone to dynamically pay more attention to features from a particular view given the presented scene. For instance, as shown in Figure 5, the attention-based RV-BEV weighting module assigns higher weights for the RV features representing nearby and smaller objects, while for the occluded and distant objects it favors BEV features more.

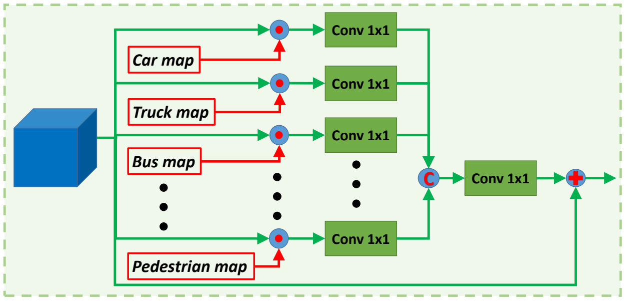

Figure 6. Class-wise foreground attention module. This module is responsible to embed the semantic information regarding the object of interest in the feature maps.

4.2. Class-wise Foreground Attention Module

In the proposed multi-task framework, the panoptic segmentation estimates the semantic labels for all points in the RV plane, covering both foreground and background objects. However, for 3D object detection, we are only interested in detecting foreground objects. Thus, leveraging the foreground semantic information can ease the detection task by placing more focus on class-specific foreground regions.

For this purpose, a class-wise foreground attention module is designed, as shown in Figure 6. The module takes in the estimated probability maps for each foreground object category from CPSeg. These maps are projected to the BEV plane and down-sampled via max-pooling to match the resolution of the combined RV-BEV feature maps. For each foreground object category, a class-wise attention branch is created, which performs element-wise multiplication between the probability map of the specific class and the input feature maps, and then compresses the channel depth of the resulted output through a 1×1 convolution layer. After the attention branch, feature maps are gathered and passed through another 1×1 convolution layer to keep the channel size the same as the input feature map. Finally, input feature maps are added to these semantically rich feature maps via a skip connection. Overall, this module embeds the foreground semantic information in the feature maps, which can help both the classification and localization tasks.

4.3. Center Density Heatmap Module

The aim of this module is to provide instance-level information for the detection head. The estimated 3D boxes center offsets for each point from CPSeg are used to create a heatmap of potential locations of 3D boxes centers to guide the detection head. More specifically, an estimated foreground mask from CPSeg is used to filter out background points, as the center offsets from those points are not mean-

Method	mAP	NDS	Car	Truck	Bus	Trailer	CV	Ped	Motor	Bic	TC	Barrier
CenterPoint [30]	56.4	64.8	84.7	54.8	67.2	35.3	17.1	82.9	57.4	35.9	63.3	65.1
Ours	**60.3**	**67.1**	**85.1**	**57.1**	**68.3**	**43.6**	**20.5**	**84.7**	**62.5**	**43.6**	**71.5**	**66.0**
Improvement	*+3.9*	*+2.3*	*+0.4*	*+2.3*	*+1.1*	*+8.3*	*+3.4*	*+1.8*	*+5.7*	*+7.7*	*+8.2*	*+0.9*

Table 1. Comparison of the proposed method with the two-stage CenterPoint method on the nuScenes validation set. In the columns, CV, Ped, Motor, Bic, and TC are abbreviations for Construction Vehicle, Pedestrian, Motorcycle, Bicycle, and Traffic Cone, respectively.

ingful. The remaining foreground points are shifted according to the offset predictions and projected to the BEV plane. Then, the center density heatmap is generated as,

$$H(x, y) = Tanh(log(C(x, y) + 1)) \tag{1}$$

where $C(x, y)$ is the number of projected points, $H(x, y)$ is the resulted center density heatmap, and x and y represent the horizontal and vertical coordinates on the BEV plane, respectively. The $Tanh$ activation function is applied after a log operation to constrain the heatmap values in $[0, 1]$.

This gray-scale center density heatmap is applied to the feature maps extracted from the 2D detection backbone before the detection head. An element-wise multiplication is performed between feature maps and the estimated center density heatmap and the result is added to the feature maps. As shown in Figure 1, the center density heatmap provides an effective guide to direct the detection head towards each possible box center region.

5. Experiments

We introduce the dataset used (Sec. 5.1) and the implementation details of our framework and the methods (Sec. 5.2). Moreover, we compare the proposed method with other state-of-the-art methods (Sec. 5.3) and conduct ablation studies to show the effectiveness of each new component in our framework (Sec. 5.4).

5.1. Dataset

The nuScenes [2] is a popular large-scale driving-scene dataset. It provides both ground-truth 3D boxes and panoptic labels. Moreover, as the proposed method relies on both ground-truth panoptic labels and the 3D boxes, we use the nuScenes dataset for both training and evaluation. The nuScenes dataset contains 1000 sequences of driving scenes. Overall, 40K frames are annotated for the 3D object detection task with 10 object categories, from which 28K, 6K, and 6K frames are for training, validation, and test sets, respectively. The mean Average Precision (mAP) is one of the metrics used for 3D object detection, which is calculated on the BEV plane based on different center distance thresholds *i.e.* 0.5m, 1.0m, 2.0m, and 4.0m. Another important metric is the nuScenes Detection Score (NDS), which is a weighted sum of mAP and 4 other metrics that determine the quality of the detections in terms of box translation, scale, orientations, velocity, and other attributes [2].

To demonstrate the robustness of the proposed model, we also perform experiments on the Waymo Open Dataset [21] after creating our own panoptic segmentation labels given the 3D ground-truth boxes. Detailed information regarding the experiment performed on the Waymo dataset and results are included in the supplementary materials.

5.2. Implementation Details

We used the Pytorch deep learning library [12] and based our implementation on the OpenPCDet [23], an open-source project for LiDAR-based 3D object detection. In order to demonstrate that the proposed framework can improve any BEV-based 3D object detection method, we use SECOND [26] and PointPillars [9] 3D object detectors in our framework as alternatives to the CenterPoint. For each experiment using our multi-task framework, the panoptic segmentation model CPSeg and one of the 3D object detection models, such as CenterPoint, are trained jointly from scratch in an end-to-end manner.

To supervise the detection model, the focal loss is used for regressing the center heatmap and for all the other regression targets, the Smooth L1 loss is exploited. The focal loss is computed over all locations on the output heatmap, while the regression loss is only calculated over positive locations. We followed CPSeg for all the panoptic segmentation loss terms. All the models were trained for 120 epochs on 8 Tesla V100 GPUs with Adam optimizer and a weight decay of 10^{-2}. Finally, the learning rate was set to 10^{-3} and the One Cycle policy was used for learning rate scheduling.

5.3. Results

First, we present the detection results of our method and the CenterPoint on the nuScenes validation set, as shown in Table 1. It can be seen that the proposed multi-task framework, which is based on the single-stage CenterPoint detection model, outperforms the original two-stage CenterPoint considerably in NDS and mAP. More specifically, the proposed method significantly improves the AP for all the object categories, specially the smaller ones, such as motorcycles, bicycles, and traffic cones. This performance boost is made possible by exploiting the panoptic segmentation, particularly the RV features in which the smaller objects are better preserved and represented. Qualitative results on two sample LiDAR frames of the validation set are shown in Figure 7, while additional quantitative and qualitative com-

Method	mAP	NDS	Car	Truck	Bus	Trailer	CV	Ped	Motor	Bic	TC	Barrier
WYSIWYG [7]	35.0	41.9	79.1	30.4	46.6	40.1	7.1	65	18.2	0.1	28.8	34.7
PointPillars [9]	30.5	45.3	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
PointPainting [24]	46.4	58.1	77.9	35.8	36.2	37.3	15.8	73.3	41.5	24.1	62.4	60.2
PMPNet [29]	45.4	53.1	79.7	33.6	47.1	43.1	18.1	76.5	40.7	7.9	58.8	48.8
SSN [33]	46.4	58.1	80.7	37.5	39.9	43.9	14.6	72.3	43.7	20.1	54.2	56.3
CBGS [32]	52.8	63.3	81.1	48.5	54.9	42.9	10.5	80.1	51.5	22.3	70.9	65.7
CVCNet [3]	55.3	64.4	82.7	46.1	46.6	49.4	22.6	79.8	59.1	31.4	65.6	69.6
CenterPoint [30]	58.0	65.5	**84.6**	**51.0**	60.2	53.2	17.5	83.4	53.7	28.7	76.7	70.9
HotSpotNet [4]	59.3	66.0	83.1	50.9	56.4	53.3	23.0	81.3	63.5	36.6	73.0	**71.6**
Ours	**60.9**	**67.3**	**84.6**	50.0	**63.2**	**55.3**	**23.4**	**83.7**	**65.1**	**38.9**	**76.8**	68.2
Improvement	*+1.6*	*+1.3*	*0.0*	*-1.0*	*+3.0*	*+2.0*	*+0.4*	*+0.3*	*+1.6*	*+2.3*	*+0.1*	*-3.4*

Table 2. Comparison of the state-of-the-art methods on the nuScenes test set. In the columns, CV, Ped, Motor, Bic and TC represent Construction Vehicle, Pedestrian, Motorcycle, Bicycle, and Traffic Cone, respectively.

parisons are included in the supplementary materials.

The comparison results with the state-of-the-art methods on the nuScenes test set are shown in Table 2. It can be seen that the proposed method surpasses others by a considerable margin, increasing both the NDS and mAP by 1.3 and 1.6 points over HotSpotNet, respectively. This improvement is even higher compared to the CenterPoint model, with 1.8 and 2.9 points increase in NDS and mAP, respectively.

The advantage of the proposed method over CenterPoint can be further shown in Figure 8. By using the guidance from RV panoptic features, our method improves on the corner cases, such as overlapping pedestrians or sparsely represented traffic cones. As shown in 9, point representation in BEV is less detailed, which is less problematic for large objects with more LiDAR points, but critical for thin and shallow objects. RV panoptic features are able to capture more details for bicycles and motorcycles and, as seen in Figure 8, provide strong guidance for both classification and box regression. In terms of larger classes such as bus, the larger receptive field of the network based on incorporation of the RV helps detector to perceive them better. In addition, the center density map guides the detector to locate the box centers of large objects, which is otherwise difficult as the centers are located far away from the LiDAR points on the corresponding surfaces.

5.4. Ablation Studies

Effects of each proposed component The key components proposed in our multi-task framework include the multi-view backbone augmentation, class-wise foreground attention, and center density heatmap modules. The effect of each of these components on the performance of the proposed method evaluated using the nuScenes validation set is shown in Table 3. Based on these results, the multi-view backbone has the most impact on performance improvement by providing panoptic-level RV feature maps to augment the detection backbone. Moreover, the use of class-wise foreground attention and center density heatmap

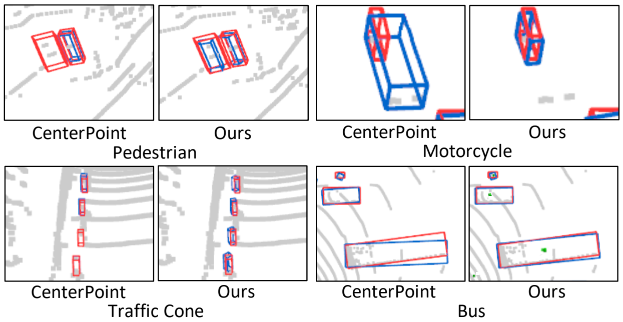

Figure 8. Class-wise qualitative results on the nuScenes validation set, containing the predicted bounding boxes (in blue), ground truth bounding boxes (in red), and center density heatmap for the bus class (in green). Best viewed in color.

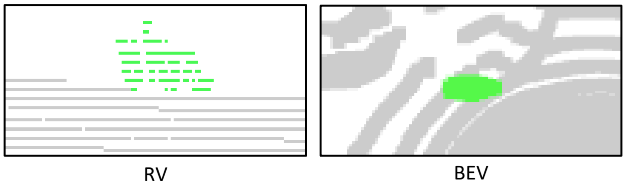

Figure 9. Comparison of point cloud representation in BEV and RV planes for a motorcycle. Best viewed in color.

Method	MBA	CFA	CDH	NDS
Baseline				63.8
	✓			66.5
	✓	✓		66.9
	✓	✓	✓	**67.1**

Table 3. Effects of different proposed component on the performance improvement evaluated on the nuScenes validation set. In the columns MBA, CFA, and CDH are abbreviations for multi-View backbone augmentation, Class-wise foreground attention, center density heatmap modules, respectively. The baseline is the single-stage CenterPoint method with VoxelNet (VN) backbone.

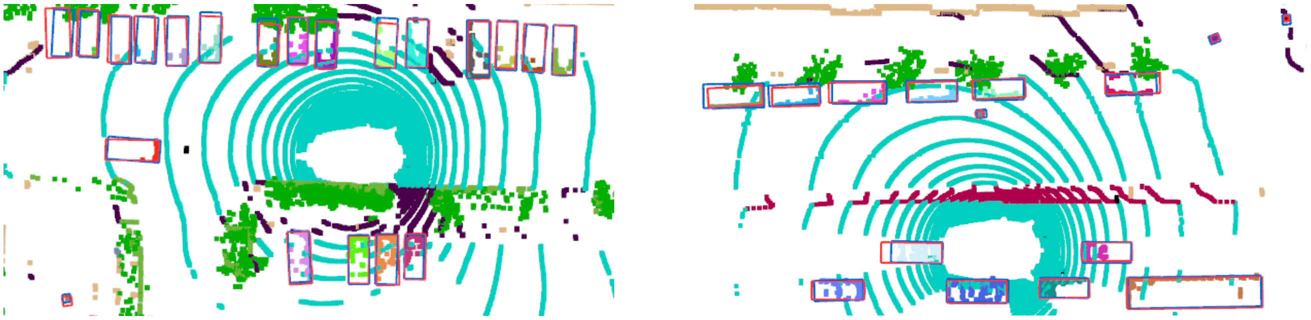

Figure 7. Examples of qualitative results, containing the predicted bounding boxes (in blue), ground truth bounding boxes (in red), and panoptic segmentation results. Best viewed in color.

Method	mAP	NDS
PointPillars [9]	43.0	56.8
Multi-Task+PointPillars	**50.5**	**60.5**
Improvement	*+7.5*	*+3.7*
SECOND [26]	51.7	62.6
Multi-Task+SECOND	**56.2**	**64.8**
Improvement	*+4.5*	*+2.2*

Table 4. Performance of PointPillars and SECOND under the proposed multi-task framework on the nuScenes validation set.

modules also contribute to the performance gains in detection scores considerably. This suggests that the injection of panoptic segmentation information provides helpful guidance for CenterPoint instead of creating confusion.

Compatibility with other BEV-based 3D object detection models To demonstrate that our proposed multi-task framework can potentially improve the performance of any BEV-based detection method, we ran another set of experiments, with two different 3D object detection methods, PointPillars and SECOND. While the detection backbone and the detection head are swapped, the rest of the framework and experiment setup are unchanged. As shown in Table 4, when integrated as part of the multi-task framework, the performance of these two detectors are improved significantly. This demonstrates that the effectiveness of our framework is universal across different BEV-based 3D detection methods. More specifically, using a feature weighting mechanism to combine multi-task, multi-view features provides an unintrusive way to enrich any BEV-based detection backbone. Furthermore, feature maps that embed potential locations of object boundaries and centers are well received by any detection head.

Effects of using a pre-trained CPSeg model Another alternative for guiding the 3D object detection model is to pre-train the panoptic segmentation model prior to training the 3D object detection model (Single-task learning). We pre-trained the CPSeg model using panoptic targets, and subsequently trained the CenterPoint model while keeping the weights of CPSeg frozen. From the results in Table 5, it can be seen that multi-task learning has a better performance. This shows that when jointly trained, CPSeg learns to pick up RV features that not only benefits the panoptic segmentation task, but also guides the detection backbone.

Method	mAP	NDS
Single-task learning	59.9	66.8
Multi-task learning	**60.3**	**67.1**

Table 5. Performance comparison between the proposed multi-task learning and the single-task learning (pre-trained CPSeg with frozen weights) on the nuScenes validation set.

5.5. Limitation and Future Work

Despite observing a boost in performance, integrating the object detection method as part of our multi-task framework has a shortcoming. The proposed framework is composed of two separate backbones, which increases the overall framework complexity. Despite some modifications to simplify both backbones, our proposed method runs at 6 FPS on the nuScenes dataset. We plan to design a shared backbone for both 3D panoptic segmentation and object detection for a reduced complexity and faster run-time.

6. Conclusions

We propose a framework for guiding the LiDAR-based 3D object detection method using panoptic segmentation. In this framework, the RV features of the panoptic segmentation model backbone are used to augment the BEV features of the detection model. Furthermore, the semantic information estimated by the segmentation model highlights the location of each class of foreground objects in the detection backbone. Also, the instance-level information guides the detection head to attend to possible centers of each object bounding box in the BEV plane. Experimental results on the nuScenes dataset, demonstrate the effectiveness of the proposed framework for increasing the detection accuracy of multiple BEV-based 3D detection methods.

References

[1] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv preprint arXiv:2005.09927*, 2020. 2

[2] 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 *CVPR*, pages 11621–11631, 2020. 2, 6

[3] Qi Chen, Lin Sun, Ernest Cheung, and Alan L Yuille. Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization. *NeurIPS*, 2020. 2, 4, 7

[4] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *ECCV*, pages 68–84. Springer, 2020. 2, 4, 7

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, pages 1907–1915, 2017. 4

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *CVPR*, pages 6569–6578, 2019. 2, 3

[7] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *CVPR*, pages 11001–11009, 2020. 7

[8] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, pages 1–8, 2018. 1

[9] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 1, 2, 4, 6, 7, 8, 11, 12

[10] Enxu Li, Ryan Razani, Yixuan Xu, and Liu Bingbing. Cpseg: Cluster-free panoptic segmentation of 3d lidar point cloud, 2021. 2, 3

[11] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *IROS*, pages 8505–8512, 2020. 4

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32:8026–8037, 2019. 6

[13] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, pages 918–927, 2018. 1, 2

[14] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 2

[16] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020. 2

[17] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 1, 2

[18] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE TPAMI*, 2020. 1

[19] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV Workshops*, pages 0–0, 2018. 2

[20] Kshitij Sirohi, Rohit Mohan, Daniel Bscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation, 2021. 4

[21] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 6, 11

[22] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*, pages 5725–5734, 2021. 1, 2

[23] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6

[24] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, pages 4604–4612, 2020. 7

[25] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 4

[26] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 4, 6, 8, 11, 12

[27] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, pages 7652–7660, 2018. 1, 2

[28] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, pages 1951–1960, 2019. 1, 2

[29] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *CVPR*, pages 11495–11504, 2020. 7

[30] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 2, 3, 4, 6, 7, 12

[31] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 1, 2

[32] 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. 7

[33] Xinge Zhu, Yuexin Ma, Tai Wang, Yan Xu, Jianping Shi, and Dahua Lin. Ssn: Shape signature networks for multi-class object detection from point clouds. In *ECCV*, pages 581–597, 2020. 7