

PVAFN: Point-Voxel Attention Fusion Network with Multi-Pooling Enhancing for 3D Object Detection

Yidi Li^{1,2} Jiahao Wen¹ Bin Ren^{3,4} Wenhao Li² Zhenhuan Xu¹ *Hao Guo¹ Hong Liu² Nicu Sebe⁴

¹College of Computer Science and Technology, Taiyuan University of Technology

²Key Laboratory of Machine Perception, Peking University

³University of Pisa ⁴University of Trento

Abstract—The integration of point and voxel representations is becoming more common in LiDAR-based 3D object detection. However, this combination often struggles with capturing semantic information effectively. Moreover, relying solely on point features within regions of interest can lead to information loss and limitations in local feature representation. To tackle these challenges, we propose a novel two-stage 3D object detector, called Point-Voxel Attention Fusion Network (PVAFN). PVAFN leverages an attention mechanism to improve multi-modal feature fusion during the feature extraction phase. In the refinement stage, it utilizes a multi-pooling strategy to integrate both multi-scale and region-specific information effectively. The point-voxel attention mechanism adaptively combines point cloud and voxel-based Bird’s-Eye-View (BEV) features, resulting in richer object representations that help to reduce false detections. Additionally, a multi-pooling enhancement module is introduced to boost the model’s perception capabilities. This module employs cluster pooling and pyramid pooling techniques to efficiently capture key geometric details and fine-grained shape structures, thereby enhancing the integration of local and global features. Extensive experiments on the KITTI and Waymo datasets demonstrate that the proposed PVAFN achieves competitive performance. The code and models will be available.

Index Terms—3D object detection, point-voxel, attention fusion, multi-pooling.

I. INTRODUCTION

Recently, LiDAR point cloud-based 3D object detection has become a focal point for addressing essential perception tasks in autonomous driving. LiDAR sensors are valued for their ability to provide precise distance measurements in diverse conditions, making them widely applicable in 3D tasks such as odometry, mapping [1, 2, 3], object tracking [4, 5, 6], and detection [7, 8, 9]. However, the raw point cloud data is inherently unstructured, sparse, and irregular, which complicates feature extraction and limits the effectiveness of traditional image-based methods. Consequently, it is imperative to develop and refine techniques specifically tailored for the analysis of point cloud data.

Based on the data type, current LiDAR-based 3D object detection methods can be broadly divided into two categories, i.e., the voxel-based and point-based methods. Voxel-based methods convert irregular point clouds into regular voxels and

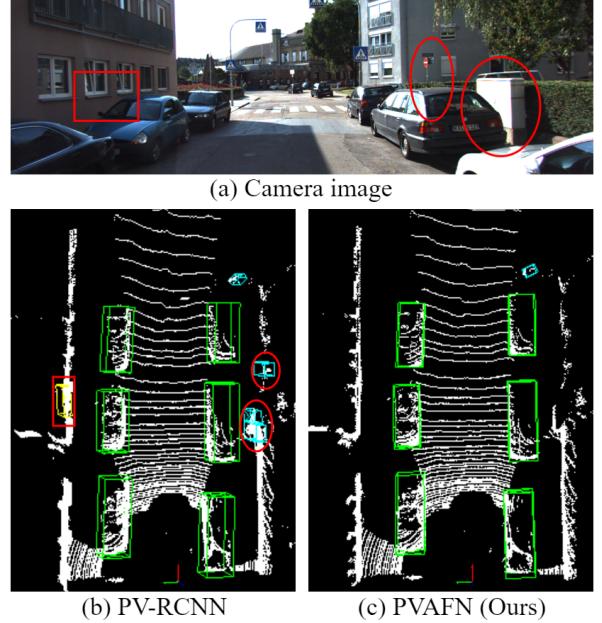


Fig. 1: Comparison between (b) PV-RCNN [8] and (c) our PVAFN. Green, blue, and yellow boxes denote predicted cars, pedestrians, and cyclists, while red boxes are false detections. PVAFN effectively minimizes false detections, such as utility boxes, road signs, and walls.

then learn high-dimensional features through 3D convolution [10, 7, 11, 12, 13]. Usually, it is hard for voxel-based methods to find a balance between accuracy and efficiency, i.e., smaller voxels bring higher accuracy at the cost of higher computational cost, while using more significant voxels may ignore potential local details within the voxels. In contrast, point-based methods directly take point clouds as input and extract keypoint features with PointNet [14] or its variants [15, 16, 17]. Although this approach minimizes information loss compared to voxelization, efficiently capturing the local geometric structure and fine-grained details of the point cloud remains challenging due to the data’s inherent sparsity and irregularity.

An alternative approach is to combine point-based and voxel-based representations, which can strike a balance between capturing richer geometric information and reducing

*Corresponding author: Hao Guo, guohao@tyut.edu.cn

computational cost. This has been validated by [8], though it still faces limitations in effectively representing points. Specifically, the naive concatenation operation may yield suboptimal results in inferring hierarchical spatial relationships and semantic contexts within the point cloud. Moreover, inadequate point sampling within a Region of Interest (RoI) during the refinement stage often leads to inaccuracies in determining object size and category. This issue is illustrated in Fig. 1, where Fig. 1 (a) shows the input camera image, Fig. 1 (b) displays the detection results from PV-RCNN[8], and Fig. 1 (c) presents the more accurate detections achieved by the proposed Point-Voxel Attention Fusion Network (PVAFN).

To address the challenge of insufficient semantic feature extraction when combining points and voxels, this paper introduces a novel two-stage 3D object detector with a point-voxel attention fusion module. This module involves three key steps: first, it integrates voxel features with BEV features to create multi-dimensional fusion features, differing from JPV-Net [18] by recognizing both voxels and BEV as regular point cloud representations in distinct dimensions. Next, the combined keypoint and hybrid features are processed through a self-attention layer [19] to enhance contextual information. Finally, the point-voxel attention fusion module adaptively merges point-wise features with voxel-BEV fusion features to improve semantic feature extraction.

Moreover, to address challenges such as point cloud sparsity, information loss, and limitations in local feature extraction, we propose a multi-pooling enhancement module for the refinement stage, which includes a RoI clustering pooling head and a RoI pyramid pooling head. The RoI clustering pooling head uses a density-based method to pinpoint key feature positions and aggregate features around density center points, thereby enhancing the capture of geometric information and eliminating background noise. The RoI pyramid pooling head incrementally expands the RoI and generates uniform grid points at each layer to form a grid pyramid [20], thereby improving global structure understanding through contextual information. The combined use of these heads significantly enhances detection accuracy.

Our main contributions are summarized as follows:

- We introduce a novel Point-Voxel Attention Fusion Network (PVAFN) for 3D object detection. PVAFN enhances feature representation by adaptively integrating point features with voxel-BEV fusion features through a module that combines self-attention and point-voxel attention, enriching contextual information.
- We propose a multi-pooling enhancement module that combines the RoI clustering pooling head and the RoI pyramid pooling head to efficiently capture key geometric details and fine-grained shapes, thereby enhancing local and global perceptions.
- Extensive experiments on the KITTI and Waymo 3D object detection datasets validate the effectiveness of PVAFN, demonstrating competitive performance in detecting cars, pedestrians, and cyclists.

II. RELATED WORK

A. Single Representation-Based 3D Object Detection.

Single representation-based 3D object detection includes voxel-based [21, 9] and point-based [16, 22] methods, each with unique benefits and challenges. Voxel-based methods like VoxelNet [10] use voxel grids and 3D CNNs for feature extraction, but face high computational costs, partially reduced by SECOND [7] with sparse CNNs. Advances also include BEV maps and pillar-based representations that leverage 2D CNNs for real-time detection [23, 24]. Despite improvements, voxel-based methods still contend with quantization errors. In contrast, point-based methods like PointNet [14] process point clouds directly without grid projection, using permutation-invariant layers to extract features. PointRCNN [25] and subsequent works [26] refine PointNet with two-stage frameworks and hybrid sampling. These methods offer enhanced flexibility and geometric details, often surpassing voxel-based techniques in detection performance. Both voxel-based and point-based approaches have advanced the field, each with unique benefits and limitations.

B. Point-Voxel Representation-Based 3D Object Detection.

Recently, methods have emerged that integrate the benefits of both point-based and voxel-based representations [8, 18]. For instance, STD [26] uses PointNet++ [15] for initial extraction and point pooling to refine features into voxel representations. Similarly, SA-SSD [27] introduces an auxiliary network that supervises voxel context to improve focus on intra-object structures. PV-RCNN [8] uses sparse convolution to generate 3D proposals and a voxel set abstraction module to refine features through keypoint extraction. The proposed PVAFN improves integration by preserving detailed point features and combining them with voxel-BEV hybrid features via the attention mechanism, achieving superior performance in merging point and voxel information than others.

C. Vision Transformer (ViTs) for Point Cloud Analysis.

Transformers [19, 28], originally for NLP, now excel in computer vision and 3D point clouds [29, 30, 31, 32]. Pointformer [17] uses local-global multi-head self-attention (MSA) for 3D point clouds, while Group-Free [33] applies MSA to all points, removing the need for handcrafted grouping. 3DETR [34] provides an end-to-end ViTs-based model with minimal 3D bias. VoTr [35] uses voxel-based ViTs with local and dilated attention to expand the receptive field. SST [36] applies MSA to non-empty voxels in shifted 3D windows. DGCNN [37] integrates BEV features with deformable attention, and VISTA [38] fuses global multi-view features via MSA. Despite advances, these methods often lose fine-grained details due to voxelization. PVAFN addresses this by adaptively combining point-wise features with voxel-BEV fusion, capturing richer contextual information.

III. THE PROPOSED METHOD

In this paper, we propose PVAFN, a novel two-stage 3D object detection network, detailed in Fig. 2. In stage

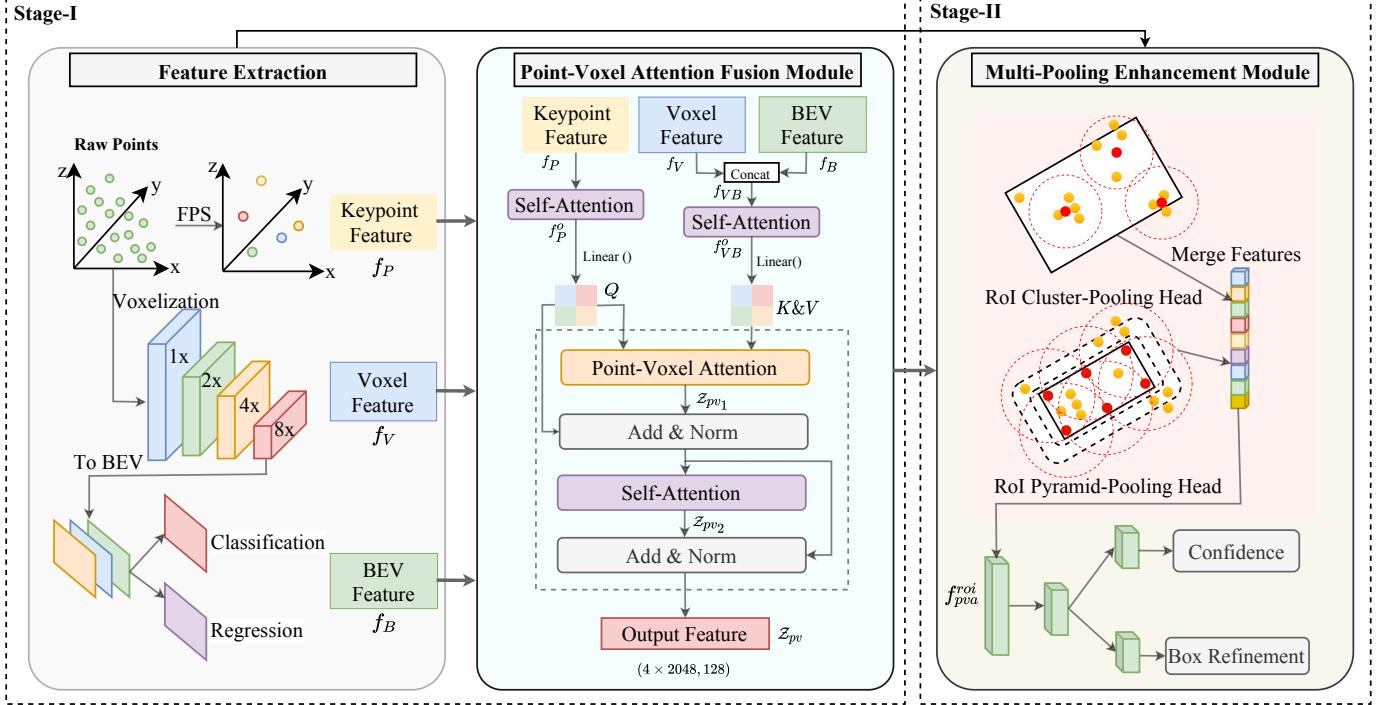


Fig. 2: Overall architecture of the proposed PVAFN. First, the raw point cloud undergoes keypoint sampling and voxelization. The resulting keypoint, voxel, and BEV features are fused using the point-voxel attention fusion module, which employs self-attention, point-voxel attention, and residual connections. The multi-pooling enhancement module then extracts geometric and fine-grained features for proposal generation and refinement.

I, downsampling and voxelization methods, similar to PVRCNN [8], are used to obtain keypoint features f_P , voxel features f_V , and BEV features f_B with the feature extraction module (left part of Stage-I in Fig. 2). These features are then processed by the proposed point-voxel attention fusion module to enhance contextual representation. In stage II, the multi-pooling enhancement module, comprising the ROI clustering pooling head for key geometric information and the ROI pyramid pooling head for fine-grained shape feature extraction, refines these features for classification and regression. Details of these components are provided in the following subsections.

A. Point-Voxel Attention Fusion Module

Motivation. Existing methods have demonstrated that ViTs can improve 3D object detection accuracy [17, 35, 34, 36]. For example, VoTr [35] innovatively applies MSA to both empty and non-empty voxel locations using sparse and submanifold voxel modules, establishing long-range relationships between voxels through an efficient attention operation. Despite these advances, voxel-only methods with ViTs still face information loss issues. Based on this observation, we conclude that a promising solution should make use of different kinds of information as much as possible. To this end, we propose the Point-Voxel Attention Fusion Module, which is designed to well integrate three kinds of 3D features with attention operation.

1) Main Pipeline.: As shown in Fig. 2, given the keypoint features f_P , voxel features f_V , and BEV features f_B from

the feature extraction module of Stage-I, f_P first go through a standard self-attention operation for keypoint feature aggregation and outputs f_P^o . Meanwhile, f_V and f_B are combined via concatenation operation for achieving better geometric advantages based on the inborn attributes from both the voxel and BEV features, forming f_{VB} . Similarly, another self-attention operation is utilized for f_{VB} and outputs f_{VB}^o . Then f_P^o and f_{VB}^o are linearly project to Q and $K \& V$ (See Fig. 3). We formulate this process as $Q = \mathbf{W}_1 f_P^o$, $K = \mathbf{W}_2 f_{VB}^o$, and $V = \mathbf{W}_{MLP} f_{VB}^o$, respectively. \mathbf{W}_1 and \mathbf{W}_2 are the learnable weight of 1D CNN while \mathbf{W}_{MLP} indicates the learnable weight of the MLP operation. Finally, Q , K , and V pass through the proposed point-voxel attention for better feature fusion.

2) Preliminaries.: Usually, the standard attention that commonly adopted for 3D point cloud [17] is formalized as follows:

$$\mathcal{Z} = \sum_{i \in \Omega(r)} \text{Softmax}(Q_i K_i^\top) \odot V_i, \quad (1)$$

where $\text{Softmax}(\cdot)$ denotes the Softmax function. \mathcal{Z} is the fused feature. Then, [39] proposed the point attention operators which works as an extension of the standard attention operator-based method, focusing more on the relationship between the position and features of points, and is defined as:

$$\mathcal{Z}_{pts} = \sum_{i \in \Omega(r)} \text{Softmax}(Q_i + K_i) \odot (V_i + Q_i). \quad (2)$$

3) Point-Voxel Attention.: The point attention operator-based method overly emphasizes the relationship between

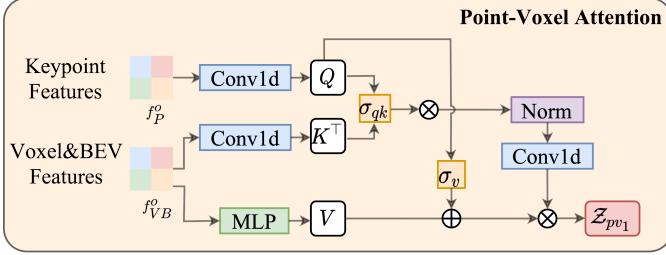


Fig. 3: Illustration of Point-Voxel Attention. It introduces a learnable gating function σ_* , composed of attention and graph operators, which can dynamically select attention components to achieve different optimization effects.

the position and features of points. However, experiments have shown that it is unsuitable for the relationship between keypoints and voxel features. Inspired by the above two approaches, we propose point-voxel attention, which fully considers the structural similarity and contextual relationships corresponding to keypoints features and voxel-BEV fusion features, with the following formulation:

$$\mathcal{Z}_{pv1} = \sum_{i \in \Omega(r)} \text{Softmax}(\sigma_{qk} Q_i K_i^\top) \odot (V_i + \sigma_v Q_i), \quad (3)$$

where σ_{qk} and σ_v are learnable gating functions implemented by linear projections of the output of the embedding by the sigmoid activation function. Through the gating functions σ_{qk} and σ_v , the point-voxel attention can selectively learn key features from the point-pixel features and balance the proportion of features used for fusion, respectively. Note that we omit the norm and 1D CNN in Eq. 3. \mathcal{Z}_{pv1} now contains the well-fused feature with attributes from both points and voxel. To further well-aggregate both information, another self-attention is applied to \mathcal{Z}_{pv1} and outputs \mathcal{Z}_{pv2} , and then a skip-connection and a normalization is applied to \mathcal{Z}_{pv2} , forming our final fused feature \mathcal{Z}_{pv} .

B. Multi-Pooling Enhancement Module

During the refinement process, we observe that some RoIs contain sparse points and extremely incomplete object shapes. As shown in Fig. 4, keypoints within the RoI are sparsely concentrated in a specific location, and there are background points, such as ground information within the RoI. Fig. 4(a) shows that using only the grid pooling head is insufficient to infer object classes, while Fig. 4(b) indicates that the cluster pooling head can effectively address this issue. To accurately infer geometric information and object categories, the precise location of key features and sufficient neighboring point data are essential. Therefore, we propose the multi-pooling enhancement module for RoI feature extraction, which includes a RoI clustering pooling head and a RoI pyramid pooling head.

1) *RoI Clustering Pooling Head*.: In previous refinement network studies, most feature extraction was conducted using RoI grid structures based on spherical radius search [8]. Although RoI grid pooling can obtain global RoI feature information, it is not easy to focus on extracting key foreground point features for some sparse and unevenly distributed points. To

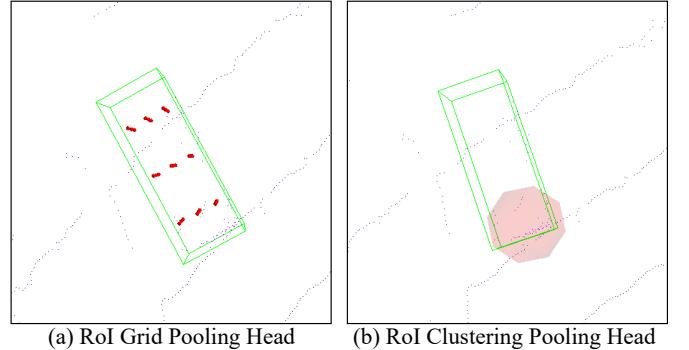


Fig. 4: Comparison of cluster pooling head and grid pooling head. For sparse and uneven RoIs, (a) (i.e., the grid pooling method) fuses all points within the RoI, including background information such as the ground. In contrast, (b) (i.e., the clustering pooling method) focuses only on the key geometric information of the target within the bounding box.

address this, we propose a RoI Clustering Pooling Head based on DBSCAN (Density-Based Spatial Clustering of Applications with Noise), which compensates for the shortcomings of RoI grid pooling, quickly locates key features, captures key geometric information, and removes background noise. For each 3D candidate box $b_i = (x_i, y_i, z_i, h_i, w_i, l_i, \theta_i)$, the RoI Clustering Pooling Head slightly enlarges its dimensions to create a new 3D box $b'_i = (x_i, y_i, z_i, h_i + \varphi, w_i + \varphi, l_i + \varphi, \theta_i)$, where φ is a constant used to expand the bounding box size. Interest points within the bounding box are clustered using DBSCAN, with clustering based on a specified search radius and minimum cluster size. For feature extraction, the process begins by calculating the average position of all points within each cluster. Subsequently, the features of all points in the cluster are aggregated around this average point.

2) *RoI Pyramid Pooling Head*.: The RoI pyramid pooling head extracts RoI features by constructing a pyramid grid structure. First, the original RoI region size is gradually expanded, and then uniform grid points are generated within the RoI at each level to construct the grid pyramid [20]. The RoI pyramid pooling head contains grid points inside and outside the RoI. The grid points inside the RoI can capture fine-grained shape structures for precise box refinement, and the grid points outside the RoI can capture extensive global contextual information for recognizing incomplete objects. The coordinates of the grid points are determined by the expanded RoI and grid sizes, expressed as:

$$p_{grid} = \left(\frac{\rho_l l_i}{n_l}, \frac{\rho_w w_i}{n_w}, \frac{\rho_h h_i}{n_h} \right) \cdot (0.5 + (i, j, k)) + (x_i, y_i, z_i), \quad (4)$$

where l_i , w_i , and h_i represent the length, width, and height of the candidate box, (x_i, y_i, z_i) is the coordinate of the candidate box, (n_l, n_w, n_h) represents the grid size, and ρ_l , ρ_w , and ρ_h represent the RoI size control rates. The grid points of each RoI level are grouped as sphere centers, and features are extracted through point-voxel attention. Let p_i, f_i be the coordinates of the i -th interest point near p_{grid} and its

corresponding feature vector. The process is as follows:

$$f_{pvAtt}^{roi} = \sum_{i \in \Omega(r)} \text{Softmax}(\sigma_{qk} Q_i^{pos} K_i^\top) \odot (V_i + \sigma_v Q_i^{pos}), \quad (5)$$

where Q_i^{pos} is the query embedding from p_{grid} to p_i , $K^i = \text{Linear}(f_i)$, V^i is the value embedding obtained from f_i , and σ_{qk} and σ_v are learnable gating functions. The point-voxel attention can effectively aggregate surrounding point features and preserve the internal structure. Finally, the RoI features from different levels are stacked and fused to obtain the grid point features of the RoI pyramid pooling head, which are then fused with the features from the RoI clustering pooling head to obtain enhanced features for refined localization and regression.

C. Optimization Objectives

The PVAFN framework is trained end-to-end, and the overall loss function consists of the first-stage RPN loss and the second-stage box refinement loss. For 3D proposal generation, we follow SECOND [7] and design a region proposal loss \mathcal{L}_{rpn} , which uses focal loss with default hyperparameters for classification, smooth-L1 loss for box regression, and cross-entropy loss for orientation, as follows:

$$\mathcal{L}_{rpn} = \mathcal{L}_{cls}^P + \beta \mathcal{L}_{reg}^P, \quad (6)$$

$$\mathcal{L}_{cls}^{rpn} = -\frac{1}{N} \sum_{i=1}^N \alpha(1-p_i)^\gamma \log \mu_i, \quad (7)$$

$$\mathcal{L}_{reg}^{rpn} = -\frac{1}{N} \sum_{r \in x,y,z,l,h,w,\theta} \mathcal{L}_s(\Delta\hat{r}, \Delta r), \quad (8)$$

where \mathcal{L}_{cls}^P and \mathcal{L}_{reg}^P represent the classification and regression losses, α and γ are prediction hyperparameters, μ_i is the probability of predicting the foreground point, $\Delta\hat{r}$ represents the predicted residual of the candidate box, and \mathcal{L}_s represents smooth-L1 loss.

Similarly, the second-stage box refinement loss \mathcal{L}_{rcnn} includes classification and regression loss functions. The regression loss function is defined the same as in the RPN, while the classification loss function is defined as:

$$\mathcal{L}_{cls}^{rcnn} = -\frac{1}{N} \sum_{i=1}^C (y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)), \quad (9)$$

where \mathcal{L}_{cls}^{rcnn} is the classification loss, y_i is the target classification label, \hat{y}_i is the target prediction probability, and N is the number of targets. Notably, in typical 3D object detection algorithms, when setting classification labels based on the IoU value, the classification label is set to 1 when the IoU σ_{iou} is greater than the foreground threshold $\sigma_{fg} = 0.75$, set to 0 when it is less than the background threshold $\sigma_{bg} = 0.25$, and set to -1 when σ_{iou} is greater than the background threshold but less than the foreground threshold.

However, this approach needs to pay more attention to key information within the target box. To fully learn all features,

we calculate the average classification loss for all points and set the target classification label as:

$$y_i = \frac{\sigma_{iou} - \sigma_{bg}}{\sigma_{fg} - \sigma_{bg}}, \quad y_i \in (0, 1), \quad (10)$$

where σ_{iou} is the IoU value, σ_{bg} is the background threshold, and σ_{fg} is the foreground threshold.

IV. EXPERIMENTS AND DISCUSSIONS

A. Implementation Details

1) *Datasets*.: We evaluate the proposed model on the KITTI [40] and Waymo [41] datasets. The KITTI dataset is a widely used autonomous driving benchmark containing 7,481 training samples and 7,518 test samples with three categories, i.e., cars, pedestrians, and cyclists. The training samples are split into a training set (3,712 samples) and a validation set (3,769 samples). Average Precision (AP) is used to evaluate all three difficulty levels (i.e., easy, moderate, and hard). Our model is trained on the training set and evaluated on the validation set. The Waymo dataset consists of 798 scenes for training and 202 scenes for validation. The evaluation protocol consists of AP and Average Precision weighted by Heading (APH). Moreover, it includes two difficulty levels: Level 1 indicates objects containing more than 5 points, while Level 2 indicates objects containing at least 1 point.

2) *Network Architecture*.: For 3D scenes in the KITTI dataset, the detection range is set to [0, 70.4] along the X-axis, [-40, 40] along the Y-axis, and [-3, 1] along the Z-axis, containing approximately 20,000 LiDAR points. A voxel size of (0.05m, 0.05m, 0.1m) is used as the voxel input to voxelize each scene, and 2,048 points are sampled from the original point cloud as the point input. For the Waymo dataset, the detection range is set to (-75.2, 75.2), (-75.2, 75.2), and (-2, 4), with a voxel size of (0.1m, 0.1m, 0.15m). As shown in Fig. 2, the voxel CNN consists of four 3D encoding levels and 2D convolutions for BEV maps, similar to the SECOND network. The feature dimension for the points is 32, the feature dimensions for the four voxel levels are (32, 64, 128, 128), and the feature dimension for the BEV map is 256. Then, the point feature dimensions are aligned with the voxel-BEV fused features through an MLP and passed to the point-voxel attention module, with the output of the 128-dimensional feature to the refinement network after multiple layers of attention. In the refinement network, the expansion range of the RoI in the RoI clustering pooling head is set to 0.4, with a minimum cluster size and radius set to 2 and 0.2, respectively. For the RoI pyramid pooling head, the official hyperparameters proposed by Pyramid-RCNN [20] are followed.

3) *Training and Inference Schemes*.: The PVAFN framework is trained end-to-end using the ADAM optimizer, with an initial learning rate and weight decay set to 0.01 and a batch size of 1, running on 4 Tesla V100 GPUs. The learning rate is decayed using a cosine annealing strategy over 80 training epochs. During training, the IoU thresholds for positive and negative anchors are set to 0.7 and 0.25, respectively. For a fair comparison, other configurations are kept the same as the baseline network [8].

Types	Methods	Car 3D AP _{R40}			Pedestrian 3D AP _{R40}			Cyclist 3D AP _{R40}		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
1-Stage	SECOND (Yan et al. 2018)	87.12	79.3	75.91	50.66	47.82	40.54	80.31	64.98	61.01
	PointPillars [24]	84.01	76.11	72.19	59.45	50.81	44.98	85.10	65.65	60.32
	CIA-SSD [42]	90.57	81.33	78.85	-	-	-	-	-	-
	SVGA-Net [43]	88.93	81.87	79.13	56.05	50.44	43.93	86.16	69.08	62.96
	IA-SSD [44]	90.47	81.72	78.20	55.90	50.03	44.00	90.23	73.25	68.41
2-Stage	PointRCNN (Shi et al. 2019)	88.96	77.05	74.50	56.11	48.41	42.33	83.54	65.82	60.33
	Part-A ² [21]	90.23	80.45	77.65	58.77	51.89	46.25	85.40	68.82	64.55
	Pyramid-PV [20]	90.00	83.48	80.09	-	-	-	-	-	-
	VoTr-TSD [35]	91.30	83.42	81.44	-	-	-	-	-	-
	PG-RCNN [45]	90.97	83.53	80.83	56.79	50.04	44.69	90.47	74.12	69.82
	PV-RCNN [8]	91.86	82.85	80.31	59.97	52.37	46.59	86.89	70.65	66.36
	PVAFN (Ours)	92.81	83.92	81.92	61.98	54.01	48.94	88.50	73.54	68.93

TABLE I: Comparison results with State-of-The-Art methods on the KITTI validation set. Best performance values are shown in bold, and second-best performance values are underlined.

Methods	Car 3D AP _{R40}		
	Easy	Mod.	Hard
SECOND (Yan et al. 2018)	82.02	72.68	66.27
CIA-SSD [42]	<u>89.59</u>	80.28	72.87
SVGA-Net [43]	<u>87.33</u>	80.47	75.91
IA-SSD [44]	88.87	80.32	75.10
PointRCNN (Shi et al. 2019)	86.96	75.64	70.70
Part-A ² [21]	87.81	78.49	73.51
GD-MAE [46]	88.14	79.03	73.55
P-PV-RCNN++ (Chen et al. 2024)	87.65	81.28	76.79
PV-RCNN [8]	90.25	81.43	76.82
PVAFN (Ours)	88.15	81.53	76.90

TABLE II: Performance comparison on KITTI test set.

B. Experimental Results

In this section, we train the model on the training set and adjust hyperparameters based on the evaluation results on the validation set. We also submit the detection results for the Car class to the official KITTI detection server for evaluation. The server calculates performance on the test set using 40 recall positions. Additionally, we evaluate the model using the Waymo dataset and compare its performance with State-of-The-Art (SoTA) 3D object detection methods. The compared methods are highly representative or employ similar techniques to those used in this paper.

As shown in Tab. I, in the 3D detection task, PVAFN significantly outperforms other state-of-the-art object detection models. Specifically, PVAFN improves by 0.95%, 1.07%, and 1.61% over PV-RCNN in the Car category in the three difficulty levels. For the Pedestrian category, PVAFN improves by 2.01%, 1.64%, and 2.35% over PV-RCNN in the three difficulty levels. For the Cyclist category, PVAFN improves by 1.61%, 0.89%, and 0.57% over PV-RCNN in the three difficulty levels. Additionally, Fig. 5 shows the qualitative results of PVAFN on the KITTI validation set.

As shown in Tab. II, we also compare PVAFN’s performance on the KITTI test set in the Car category with the latest 3D object detection methods. Compared to PV-RCNN, PVAFN performs better in moderate and hard difficulty levels. In addition, Tab. III shows the comparison results with other advanced methods on the Waymo validation set for the vehicle category, where PVAFN leads by 3.01%–3.47% at both levels due to its excellent ability to combine contextual information and capture key information.

Methods	LEVEL_1 3D AP/APH		LEVEL_2 3D AP/APH	
	1	2	1	2
SECOND (Yan et al. 2018)	72.37/71.58		63.82/63.23	
PointPillars [24]	71.56/70.87		63.15/62.59	
CIA-SSD [42]	70.61/69.58		61.73/60.82	
SWFormer [47]	77.80/77.30		69.20/68.80	
IA-SSD [44]	70.51/69.63		61.55/60.82	
CT3D [48]		76.30/-	69.04/-	
Part-A ² [21]		77.05/76.51	68.47/67.97	
Pyramid-PV [20]		76.30/75.68	67.23/66.68	
BtcDet (Xu et al. 2022)		<u>78.58/78.06</u>	70.10/69.61	
PDV (Hu et al. 2022)		76.85/76.33	69.30/68.81	
PV-RCNN [8]		77.51/76.89	68.98/68.41	
PVAFN (Ours)		80.93/80.36	72.01/71.61	

TABLE III: Comparison on Waymo vehicle validation set.

C. Ablation Experiments

To validate the effectiveness of the proposed method, we conducted ablation studies on the KITTI validation set focusing on the Point-Voxel Attention Fusion Module and the Multi-Pooling Enhancement Module.

1) *Effects of Point-Voxel Attention Fusion Module.:* We compared our module against the Voxel Set Abstraction (VSA) module [8] and other attention mechanisms to demonstrate the rationality of the design choices made for the Point-Voxel Attention Fusion Module components. Tab. IV shows the comparison results for 3D detection with 3 categories at the moderate difficulty level First, we examined the performance gains brought by the MSA [19]. Next, we compared methods based on the Standard Attention Operator (SAO) [17], Point Attention Operator (PAO) [39], and Graph-based Attention Operator (GO) [49]. Finally, we evaluated the effectiveness of using the Point-Voxel Attention alone and the combination of Self-Attention and Point-Voxel Attention. The results show that using the self-attention mechanism improves the AP (Average Precision) across all categories, highlighting the advantage of attention mechanisms in combining contextual information. Methods based on the Point Attention Operator and Graph-based Attention Operator are unsuitable for point-voxel level feature fusion. Our proposed Point-Voxel Attention outperforms other attention mechanisms, and its combination with Self-Attention significantly enhances detection performance across all categories.

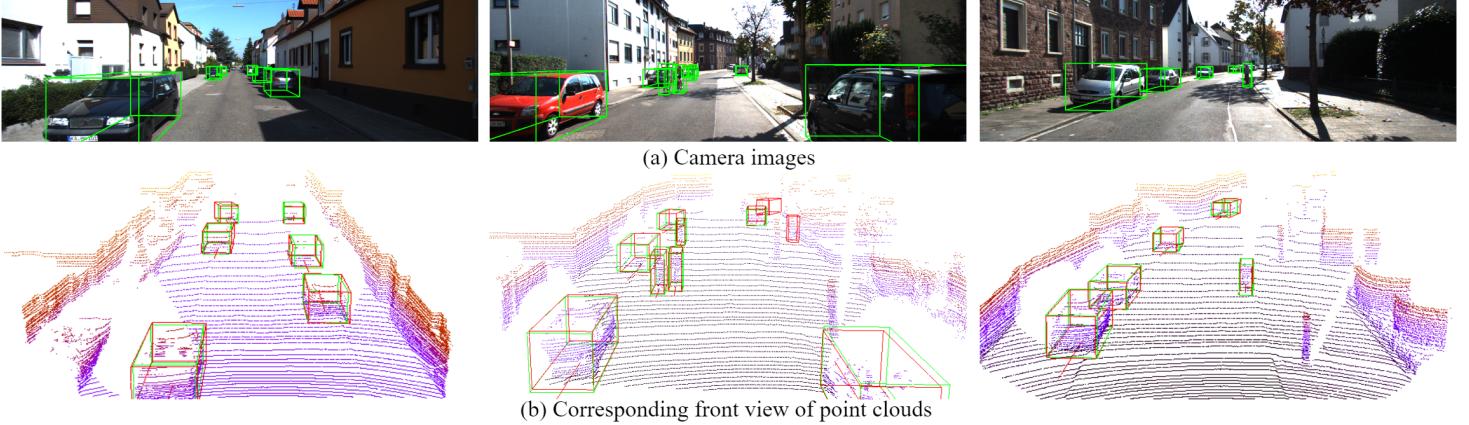


Fig. 5: Visualization results on the KITTI validation set. (a) shows the camera image, and (b) shows the corresponding front view of the point cloud. The ground truth bounding box is green, and the predicted bounding box is red.

Point & Voxel Fusion	3D AP _{R40} (Mod.)		
	Car	Ped.	Cyc.
VSA(PV-RCNN)	82.85	52.37	70.65
SA	83.14	52.42	70.81
SAO	83.02	52.50	70.31
PAO	82.00	52.40	68.44
GO	82.18	52.31	70.14
PVA (Ours)	83.54	53.01	70.89
SA+PVA (Ours)	83.74	53.88	71.32

TABLE IV: Comparison of the proposed point-voxel attention fusion module with other methods.

GPH	PPH	PPH ^{PVA}	CPH	3D AP _{R40} (Mod.)		
				Car	Ped.	Cyc.
✓	✓			82.85	52.37	70.65
		✓		83.48	52.55	70.91
			✓	83.51	52.92	71.31
	✓		✓	83.60	52.84	71.03
		✓	✓	83.53	53.85	71.11
		✓	✓	83.55	53.91	71.25

TABLE V: Performance comparison of different implementations of the proposed multi-pooling enhancement module.

2) *Effects of Multi-Pooling Enhancement Module.*: The proposed multi-pooling enhancement module comprises a Clustering Pooling Head (CPH) and a Pyramid Pooling Head (PPH). We compared the performance of the ROI Grid Pooling Head (GPH) [8], clustering pooling method, pyramid pooling method, and their combinations. In Tab. V, PPH refers to the pyramid pooling head [20], and PPH^{PVA} refers to the pyramid pooling head improved by our attention mechanism. Experiments demonstrated that the ROI clustering pooling head outperforms the grid pooling head. The improved ROI pyramid pooling head slightly outperforms the original PPH. Notably, the multi-pooling enhancement module, which combines the clustering pooling head and pyramid pooling head, achieves superior performance, improving the average precision by 0.6% – 1.54% across three categories compared to the grid pooling method.

V. CONCLUSION

In this paper, we proposed a novel two-stage 3D object detector based on the Point-Voxel Attention Fusion Network (PVAFN), which addresses the challenge of 3D object de-

tection by fusing point and voxel representations through contextual information. PVAFN has two main components: first, the proposed point-voxel attention mechanism adaptively fuses the features of points and voxel-BEV representations, capturing rich contextual information to mitigate the limitations of sparse point clouds. Second, in the refinement network stage, the proposed multi-pooling enhancement module not only acquires rich and high-granularity information through a pyramid structure but also focuses on foreground point feature extraction through the clustering pooling method, enabling rapid localization of key geometric features. PVAFN fully leverages the advantages of point and voxel representations, achieving competitive detection performance on the KITTI and Waymo datasets.

REFERENCES

- [1] J. Zhang, S. Singh *et al.*, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and systems*, vol. 2, 2014, pp. 1–9.
- [2] T. Shan and B. Englöt, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4758–4765.
- [3] H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-loam: Fast lidar odometry and mapping,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 4390–4396.
- [4] S. Wang, Y. Sun, C. Liu, and M. Liu, “Pointtracknet: An end-to-end network for 3-d object detection and tracking from point clouds,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 3206–3212, 2020.
- [5] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, “3d multi-object tracking in point clouds based on prediction confidence-guided data association,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 5668–5677, 2021.
- [6] H. Wu, Q. Li, C. Wen, X. Li, X. Fan, and C. Wang, “Tracklet proposal network for multi-object tracking on point clouds,” in *IJCAI*, 2021, pp. 1165–1171.
- [7] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, p. 3337, 2018.

- [8] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, “Pv-rnn: Point-voxel feature set abstraction for 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.
- [9] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, “Voxel r-cnn: Towards high performance voxel-based 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 1201–1209.
- [10] Y. Zhou and O. 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*, 2018, pp. 4490–4499.
- [11] J. Sun, Y.-M. Ji, F. Wu, C. Zhang, and Y. Sun, “Semantic-aware 3d-voxel centernet for point cloud object detection,” *Computers & Electrical Engineering*, vol. 98, p. 107677, 2022.
- [12] B. Ren, M. Liu, R. Ding, and H. Liu, “A survey on 3d skeleton-based action recognition using learning method,” *Cyborg and Bionic Systems*, vol. 5, p. 0100, 2024.
- [13] H. Liu, B. Ren, M. Liu, and R. Ding, “Grouped temporal enhancement module for human action recognition,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1801–1805.
- [14] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.
- [15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [16] W. Shi and R. Rajkumar, “Point-gnn: Graph neural network for 3d object detection in a point cloud,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.
- [17] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, “3d object detection with pointformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7463–7472.
- [18] N. Song, T. Jiang, and J. Yao, “Jpv-net: Joint point-voxel representations for accurate 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 2271–2279.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [20] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, “Pyramid r-cnn: Towards better performance and adaptability for 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2723–2732.
- [21] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, “From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, pp. 2647–2664, 2020.
- [22] Y. Zhang, D. Huang, and Y. Wang, “Pc-rgnn: Point cloud completion and graph neural network for 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 3430–3437.
- [23] B. Yang, W. Luo, and R. Urtasun, “Pixor: Real-time 3d object detection from point clouds,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.
- [24] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.
- [25] S. Shi, X. Wang, and H. Li, “Pointrcnn: 3d object proposal generation and detection from point cloud,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.
- [26] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, “Std: Sparse-to-dense 3d object detector for point cloud,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1951–1960.
- [27] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, “Structure aware single-stage 3d object detection from point cloud,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 873–11 882.
- [28] B. Ren, Y. Liu, Y. Song, W. Bi, R. Cucchiara, N. Sebe, and W. Wang, “Masked jigsaw puzzle: A versatile position embedding for vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 382–20 391.
- [29] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.
- [30] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, “Point-bert: Pre-training 3d point cloud transformers with masked point modeling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 313–19 322.
- [31] Q. Ma, D. P. Paudel, E. Konukoglu, and L. Van Gool, “Implicit-zoo: A large-scale dataset of neural implicit functions for 2d images and 3d scenes,” *arXiv preprint arXiv:2406.17438*, 2024.
- [32] B. Ren, G. Mei, D. P. Paudel, W. Wang, Y. Li, M. Liu, R. Cucchiara, L. Van Gool, and N. Sebe, “Bringing masked autoencoders explicit contrastive properties for point cloud self-supervised learning,” *arXiv preprint arXiv:2407.05862*, 2024.
- [33] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, “Group-free 3d object detection via transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2949–2958.
- [34] T. Guan, J. Wang, S. Lan, R. Chandra, Z. Wu, L. Davis, and D. Manocha, “M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transform-

- ers,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 772–782.
- [35] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, “Voxel transformer for 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3164–3173.
- [36] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang, “Embracing single stride 3d object detector with sparse transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8458–8468.
- [37] Y. Wang and J. M. Solomon, “Object dgcn: 3d object detection using dynamic graphs,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 20745–20758, 2021.
- [38] S. Deng, Z. Liang, L. Sun, and K. Jia, “Vista: Boosting 3d object detection via dual cross-view spatial attention,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8448–8457.
- [39] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, “Point transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16259–16268.
- [40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, pp. 1231–1237, 2013.
- [41] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine et al., “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.
- [42] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, “Cia-ssd: Confident iou-aware single-stage object detector from point cloud,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 3555–3562.
- [43] Q. He, Z. Wang, H. Zeng, Y. Zeng, and Y. Liu, “Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 870–878.
- [44] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, “Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18953–18962.
- [45] I. Koo, I. Lee, S.-H. Kim, H.-S. Kim, W.-J. Jeon, and C. Kim, “Pg-rcnn: Semantic surface point generation for 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18142–18151.
- [46] H. Yang, T. He, J. Liu, H. Chen, B. Wu, B. Lin, X. He, and W. Ouyang, “Gd-mae: Generative decoder for mae pre-training on lidar point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9403–9414.
- [47] P. Sun, M. Tan, W. Wang, C. Liu, F. Xia, Z. Leng, and D. Anguelov, “Swformer: Sparse window transformer for 3d object detection in point clouds,” in *European Conference on Computer Vision*. Springer, 2022, pp. 426–442.
- [48] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, “Improving 3d object detection with channel-wise transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2743–2752.
- [49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Transactions on Graphics*, vol. 38, pp. 1–12, 2019.