DTSSD: Dual-Channel Transformer-Based Network for Point-Based 3D Object Detection

Zhijie Zheng, Zhicong Huang, Jingwen Zhao, Haifeng Hu, *Member, IEEE*, and Dihu Chen

***Abstract*—In the field of 3D object detection, previous methods mainly utilize one channel feature encoding network to extract point-wise features. Despite the effectiveness, we find that only leveraging one channel encoding network is not sufficient and impedes the detection performance. To this end, we propose a dual-channel transformer-based feature encoding network, which integrates both set abstraction layer and transformer block as backbone. It enables the model to exploit fine-grained as well as long-range contextual information of objects, thus providing complementary relationship of two methods. In addition, a centroid estimation module is introduced to obtain powerful representation of the whole object. Finally, considering the significance of point density, which is crucial for detection performance, we propose a central density-aware enhancement module to equip center features with distinct density features. Experimental results on KITTI dataset show the effectiveness of our proposed method.**

***Index Terms*—3D object detection, attention, center estimation, point density, transformer.**

I. Introduction

With the rapid development of deep learning, object detection has aroused much research interest recently. Accordingly, it can be classified into 2D object detection [1], [2], [3] and 3D object detection. Currently, 3D object detection [4], [5], [6], [7] has been increasingly popular with researchers as it plays a vitally significant role in several scenarios like automatic driving and so forth. Nonetheless, it is still quite challenging to perform accurate 3D object detection owing to the non-uniformity and orderless nature of point clouds [8]. To tackle this, existing methods can be divided into two categories, that is, point-based methods and voxel-based methods.

Point-based methods have recently dominated in the research field of 3D object detection, which are effective and straightforward by using set abstraction layers to aggregate point-wise features. [9] proposes a centroid-aware sampling strategy to maintain more important foreground points. [10] introduces local and global transformer which take place of PointNet++ [11] in order to learn features more effectively. These methods are innovative and provide a new perspective for 3D object detection. However, there still exist two limitations of point-based methods: (1) they lack efficient point feature learning strategy. That is, they only consider one channel feature encoding network, i.e., PointNet++ or transformer as backbone to learn point features. (2) they ignore distinct point density information of non-uniform distributed point cloud, which contributes to the precise detection performance.

To tackle the above challenges, we propose a dual-channel transformer-based feature encoding network to enhance the extracted point-wise features. Besides, a centroid estimation module is adopted following [9] to predict instance centers of objects based on point-wise features so that we can obtain more robust representation of the whole object. Furthermore, taking inspiration from [12], we propose a central density-aware enhancement module to equip object centers with additional density information, which helps the representative centers better perceive the surrounding environment, thus boosting the final detection performance.

Overall, our contributions of this paper can be summarized as follows:

1) We propose a dual-channel transformer-based single-stage feature encoding network (DTSSD), which contains both set abstraction layer and transformer block to solve the insufficient feature learning problem, then a centroid estimation module is performed to predict accurate instance centers of possible objects, providing the robust object representation.
2) We propose a central density-aware enhancement module termed CDEM, with the purpose of supplementing distinct centers with additional density features of neighboring points.
3) The experimental results on challenging KITTI dataset demonstrate that our approach achieves competitive performance for 3D object detection task.

II. Proposed Method

In this letter, we propose the dual-channel transformer-based network, a single-stage point-based 3D object detection pipeline. As shown in Fig. 1, our backbone comprises of stacked set abstraction layers and transformer blocks for efficient feature extraction. Next, a centroid estimation module facilitates the model to predict instance centers of possible objects. Furthermore, we propose the central density-aware enhancement module to focus on point density information around centers.

A. Dual-Channel Transformer-Based Backbone

Current point-based methods adopt set abstraction layer or transformer block as the backbone. Generally, the former

Manuscript received 28 March 2023; revised 30 May 2023; accepted 2 June 2023. Date of publication 6 June 2023; date of current version 11 July 2023. This work was supported by the Science and Technology Program of Guangdong Province under Grant 2022B0701180001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Arash Mohammadi. *(Corresponding author: Dihu Chen.)*

Zhijie Zheng, Zhicong Huang, and Jingwen Zhao are with the Sun Yat-sen University, Guangzhou 510275, China (e-mail: zhengzhj6@mail2.sysu.edu.cn; huangzhc3@mail2.sysu.edu.cn; zhaojw27@mail2.sysu.edu.cn).

Haifeng Hu and Dihu Chen are with the School of Electronic and Information Technology, Sun Yat-sen University, Guangzhou 510275, China (e-mail: huhaif@mail.sysu.edu.cn; stscdh@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/LSP.2023.3283468

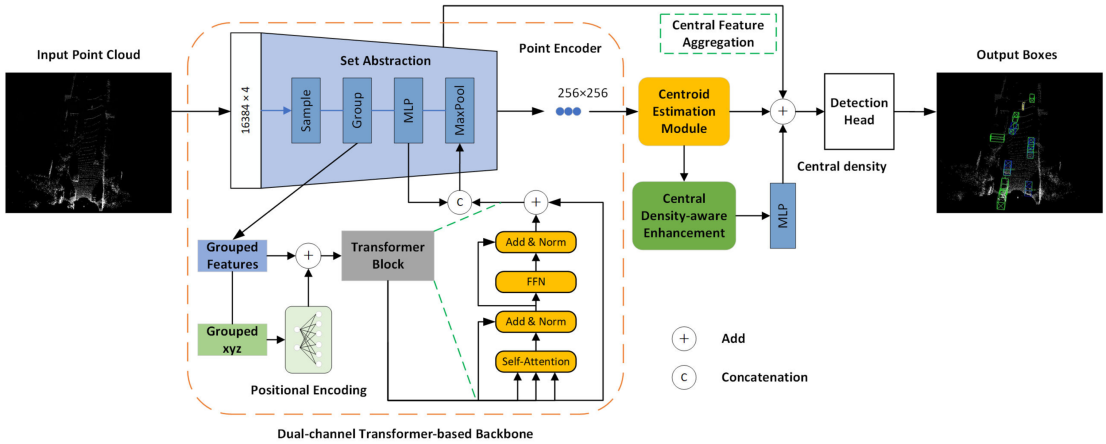

Fig. 1. The overall framework of our proposed DTSSD. The input point clouds are first put into dual-channel transformer-based backbone to learn multi-scale fine-grained as well as long-range contextual features. Then the centroid estimation module is used to generate predicted instance centers on the basis of downsampled foreground points. Finally, central density-aware enhancement module is used to obtain point density features, which are fused with center features for final proposal generation.

methods [4], [9] are able to capture fine-grained point-wise features, while the latter [10], [13] can mine long-range contextual information [14] of objects. Whereas, most previous works only consider one channel feature encoding network, which is not sufficient for feature extraction of sparse point clouds. Therefore, we design a dual-channel transformer-based backbone to combine the merits of the above two methods, as illustrated in the point encoder in Fig. 1.

Specifically, given the input point cloud which is denoted as $\{d_i = \{x_i, f_i\} | i = 1, \ldots, K\}$, where $x_i \in R^3$ are the xyz coordinates of points, $f_i \in R^C$ are the point intrinsic features like reflectance, and K is the number of point clouds. We first use furthest point sampling (FPS) to downsample the point clouds into a set of key points. Afterwards, the ball query is performed to search N neighboring points around the key points within a hand-crafted radius and get the corresponding indices. Then based on the obtained indices, we utilize grouping operation to obtain the coordinates and features of neighboring points. Next, a series of MLPs(Multi-Layer Perceptron) are adopted to aggregate the neighboring point features to generate fine-grained point-wise features for the key points as

$$\tilde{f}_i^{(c)} = Max\left\{G(\tilde{f}_j^{(n)})\right\}, j = 0, \ldots, N-1 \tag{1}$$

where $G(\cdot)$ denotes the multi-layer perceptron network, $Max(\cdot)$ is the max-pooling operation. $\tilde{f}_j^{(n)}$ represents the grouped feature of neighboring points.

Simultaneously, we use two layers of 2D convolution as positional encoding to obtain the location information of grouped points. It is subsequently added with grouped features, denoted as $f_i^{(g)}$, and put into transformer block for exploiting long-range contextual information. Specifically, the transformer block contains $multi\text{-}head\ self\text{-}attention$ layer and $feed\text{-}forward$ network (FFN) module. The obtained features can be formulated as follows:

$$Q_i = f_i^{(g)} W^Q, K_i = f_i^{(g)} W^K, V_i = f_i^{(g)} W^V \tag{2}$$

$$Multihead(Q, K, V) = CONCAT(head_1, \ldots, head_n)$$

$$where \quad head_i = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{3}$$

$$out = f_i^{(g)} + MultiHead(Q, K, V) \tag{4}$$

$$\tilde{f}_i^{(trans)} = \sigma\left(f_i^{(g)} + FFN(out)\right) \tag{5}$$

where W^Q, W^K, W^V are the projections for query, key and value, respectively. n denotes the number of attention heads and d_k is the scaling factor to smooth gradients. $\sigma(\cdot)$ is the layer normalization function.

Finally, we concatenate the latent features from the above two branches together, which are integrated by the additional aggregation layers to get rich representation of point clouds as follows:

$$\tilde{f}_i^{(fuse)} = \phi\left(CONCAT\left(\tilde{f}_i^{(c)}, \tilde{f}_i^{(trans)}\right)\right) \tag{6}$$

where $\phi(\cdot)$ denotes the aggregation layer, consisting of one-dimensional convolution layer with batchnorm and relu.

Based on the above analysis, our proposed backbone can effectively obtain representative features $\tilde{f}_i^{(fuse)}$, which includes fine-grained as well as long-range contextual information of objects from dual channels. It minimizes the problem of insufficient feature learning, which limits the performance of previous point-based detectors utilizing only one-channel backbone.

B. Centroid Estimation Module

Unlike former point-based methods [15] using FP layers to broadcast features back to original point clouds for proposal generation, we aim to utilize aggregated semantic clues to predict instance centers. By this means, we can not only obtain the robust representation of the whole object but also improve the model efficiency through saving much computational overhead. Specifically, candidate center points are generated by predicting an offset to their corresponding instance within ground-truth bounding box, as shown in Fig. 2. It is noted that the shifted centers can better represent the whole object, thus boosting the detection performance [16]. Besides, considering that several

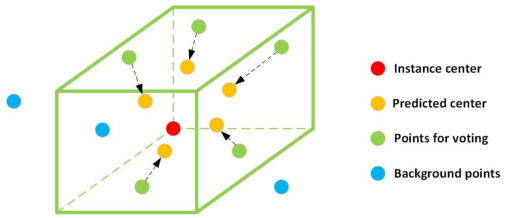

Fig. 2. The illustration of centroid estimation module. The predicted offsets to instance centers are obtained through feeding point semantic clues into two layers of shared MLPs.

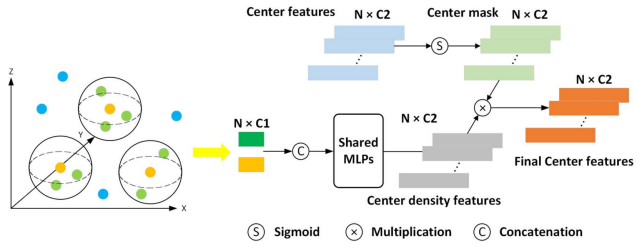

Fig. 3. The illustration of our proposed central density-aware enhancement module. The dot density features of each center are calculated and multiplied with center masks from center features to supplement centers with rich location information.

representative points outside the box contribute to the enhancement of final detection performance, we enlarge the ground-truth box to include more points. However, this method will bring about extra background points, which will negatively affect the accurate prediction of instance centers. Therefore, we use a discriminant function R_s to decide which point should be counted for voting based on its semantic features.

The centroid estimation module is trained by centroid-aware loss [9] that is utilized to supervise the predicted centers closer to ground truth centers. The loss formulation is as follows:

$$L_{cvo} = \frac{1}{|H_+|}\frac{1}{|P_+|}\sum_i\sum_j\left(|\triangle\tilde{s}_{ij} - \triangle s_{ij}| + |\tilde{s}_{ij} - \overline{s_i}|\right) \quad (7)$$

$$L_{cvo} = L_{cvo}\cdot R_s\left(p_{ij}\right), \quad \overline{s_i} = \frac{1}{|P_+|}\sum_j \hat{s}_{ij} \quad (8)$$

where $\triangle\tilde{s}_{ij}$ denotes the predicted offset to the instance centers, $\triangle s_{ij}$ denotes the offset from points p_{ij} to the centers of ground-truth box. $\tilde{s}_{ij}$ is the coordinates of predicted centers and $\overline{s_i}$ indicates the average distance of all predicted centers within i_{th} instance. $|P_+|$ and $|H_+|$ represent the number of points for voting and the amount of all points on object surface, respectively.

C. Central Density-Aware Enhancement Module

Since distinct point density features are crucial to the accurate object detection, which helps the model better perceive the surrounding geometry and location information of latent objects [12]. Therefore, we propose a central density-aware enhancement module by calculating dot density features of each predicted center to enhance the centers with point density features, as shown in Fig. 3.

Specifically, given the predicted instance centers, the estimated density of each center within the ball is computed through the index of neighboring points around the center.

First, let $\{idx_{ij}^{(n)}|i=0,\ldots,M-1; j=0,\ldots N-1\}$ denote the index of neighboring point of i-th instance center. M is the number of predicted instance centers. N represents how many neighboring points are around each center within ball. Then, the dot density feature is calculated as follow:

$$\tilde{f}_i^{(density)} = \delta\left(\sum_{i=0}^{M-1}\sum_{j=0}^{K-1} I\left(\left|idx_i^{(c)} - idx_{ij}^{(n)}\right|\right)\right) \quad (9)$$

where $idx_i^{(c)}$ is the i-th instance center index, $I(\cdot)$ is the function to decide the number of points around centers and δ denotes the sigmoid function. Thus, we can obtain the discriminative dot density feature of each center. Subsequently, the density features are put into shared MLPs to generate high dimension feature maps. While considering that some density information may bring noise interference, which leads to false detection, we use center features to generate center masks through sigmoid function with a predefined threshold. And the masks are multiplied with density feature maps, which are added with center features for the final bounding box prediction as follows:

$$mask_i = \begin{cases} 1 & \text{if } \delta\left(\tilde{f}_i^{(c)}\right) > 0.8 \\ 0 & \text{if } \delta\left(\tilde{f}_i^{(c)}\right) < 0.8 \end{cases} \quad (10)$$

$$F_i = mask_i \times MLP(\tilde{f}_i^{(density)}) + \tilde{f}_i^{(c)} \quad (11)$$

D. Loss Function

The whole loss of our framework comprises of foreground point sampling loss L_{fps}, center voting loss L_{cvo}, bounding box classification loss L_{cls} and generation loss L_{box}. The total loss L_{total} is as follows:

$$L_{total} = \lambda_{fps}L_{fps} + \lambda_{cvo}L_{cvo} + \lambda_{cls}L_{cls} + \lambda_{box}L_{box} \quad (12)$$

where λ_{fps}, λ_{cvo}, λ_{cls} and λ_{box} are the predefined weighted parameters.

To be specific, we use cross entropy loss and smooth-L1 loss for bounding box classification and generation, respectively. Besides, the bounding box generation loss L_{box} includes location regression loss L_{loc}, size regression loss L_{size}, angle regression loss L_{angle}, and corner loss L_{corner}. Therefore, the formulation of L_{box} is

$$L_{box} = L_{loc} + L_{size} + L_{angle} + L_{corner} \quad (13)$$

III. Experiments

A. Datasets and Implementation Details

In this section, we validate our model performance on the challenging KITTI Dataset [23], which contains 7481 training samples and 7518 testing samples. Following the common practice [24], we divide the training samples into train split (3712 samples) and val split (3769 samples). Then, the performance of our model is evaluated on the KITTI online test server. For the evaluation metric, we adopt standard average precision calculated with 40 recall positions to evaluate our model performance

TABLE I
QUANTITATIVE COMPARISONS OF VARIOUS METHODS ON THE KITTI TEST SET

Type	Method	3D Car (IoU=0.7)			3D Ped. (IoU=0.5)			3D Cyc. (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
1-stage	VoxelNet [8]	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
	SECOND [17]	84.65	75.96	68.71	45.31	35.52	33.14	75.83	60.82	53.67
	PointPillars [18]	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
	VoTr-SSD [13]	86.73	78.25	72.99	-	-	-	-	-	-
	DVFENet [19]	86.20	79.18	74.58	43.55	37.50	35.33	78.73	62.00	55.18
	Faraway-Frustum [20]	87.45	79.05	76.14	46.33	38.58	35.71	77.36	62.00	55.40
	IA-SSD(multi) [9]	88.34	80.13	75.04	46.51	39.03	35.60	78.35	61.94	55.70
2-stage	EPNet [21]	89.81	79.28	74.59	-	-	-	-	-	-
	Sem-Aug [22]	89.41	80.77	75.90	-	-	-	-	-	-
	Pointformer [10]	87.13	77.06	69.25	50.67	42.43	39.60	75.01	59.80	53.99
	PointRCNN [15]	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
1-stage	DTSSD	88.62	79.94	74.91	45.03	38.75	36.70	80.61	66.12	60.10

Fig. 4. Qualitative results performed on the KITTI training split. Ground-truth bounding boxes are marked in blue and predicted bounding boxes are marked in green, cyan, and yellow for three classes, respectively.

of three categories (Car, Pedestrian, Cyclist) on easy, moderate and hard difficulties.

Our model is trained end-to-end for 80 epochs on the KITTI Dataset [23] with a batch size of 16 on 4 NVIDIA GTX 1080Ti GPUs, which is optimized by Adam optimizer [25]. The initial learning rate is set to 0.01 and is updated by one-cycle strategy [26]. Besides, in order to avoid overfitting, we adopt 4 commonly used data augmentation methods in [27] (random flipping, random global scaling, random global rotations and ground truth data augmentation).

B. Comparison With State-of-the-Arts

Table I shows the model performance on the KITTI $test$ set. It can be seen that the proposed DTSSD outperforms other detectors on cyclist detection performance by a large margin. Specifically, our DTSSD surpasses IA-SSD(multi) [9] by (2.26%, 4.18%, 4.40%) regarding to cyclist detection. Moreover, our model achieves competitive car detection performance even with several two-stage detectors, which further shows the effectiveness of our proposed DTSSD. Besides, we conduct model complexity experiment to compare model parameters of different methods, as shown in Table III. It can be observed that our model parameters are much smaller than PDV [12] by almost 4 times, thus proving the low complexity of our model. Furthermore, we present some qualitative results of DTSSD on the KITTI training split, as shown in Fig. 4.

C. Ablation Studies

In the section, we conduct ablation experiments to show the effectiveness of different components in our model. Note

TABLE II
ABLATION STUDIES OF DTSSD WITH DIFFERENT COMPONENTS ON KITTI VAL SPLIT

	CD	DT	Car Mod (IoU=0.7)	Ped. Mod (IoU=0.5)	Cyc. Mod (IoU=0.5)
1	-	-	78.97	54.98	71.68
2	✓	-	78.93	55.17	71.47
3	-	✓	79.09	57.54	72.28
4	✓	✓	**79.39**	**58.64**	**72.77**

TABLE III
COMPARISONS ON THE MODEL COMPLEXITY OF VARIOUS METHODS

Methods	Model parameters
DTSSD(ours)	3.20M
PDV [12]	12.86M
IA-SSD [9]	2.70M
VoTr-SSD [13]	4.84M
PointRCNN [15]	4.04M
EPNet [21]	15.68M

that, all ablation experiments are trained on the training split and evaluated on the validation split based on 3D mAP with 11 recalls. As shown in Table II, CD and DT represent the central density-aware enhancement module and dual-channel transformer-based backbone, respectively. We adopt IA-SSD [9] as the baseline on the first row. When both two modules are appended to the baseline and we obtain the best detection performance, which outperforms the baseline by (0.42%, 3.66%, 1.09%) on car, pedestrian and cyclist category, respectively. This validates the effectiveness of dual-channel transformer-based backbone and central density-aware enhancement module.

IV. CONCLUSION

In this letter, we propose a novel dual-channel transformer-based network DTSSD, which can extract fine-grained as well as long-range contextual information of objects. The centroid estimation module guides the model to focus on instance centers to obtain robust representation of the whole object. We further devise the central density-aware enhancement module to supplement the centers with additional density features to boost the detection performance. Our future work includes utilizing RGB image information to minimize the false detection of small-scale objects to improve our model performance.

REFERENCES

[1] G. Lee, S. Hong, and D. Cho, "Self-supervised feature enhancement networks for small object detection in noisy images," *IEEE Signal Process. Lett.*, vol. 28, pp. 1026–1030, 2021.

[2] D. Zhou, Y. Tian, W. Chen, and G. Huang, "Self-supervised saliency estimation for pixel embedding in road detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 1325–1329, 2021.

[3] Y. Zhuge, G. Yang, P. Zhang, and H. Lu, "Boundary-guided feature aggregation network for salient object detection," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1800–1804, Dec. 2018.

[4] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11037–11045.

[5] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10526–10535.

[6] Z. Li, Y. Yao, Z. Quan, W. Yang, and J. Xie, "SIENet: Spatial information enhancement network for 3D object detection from point cloud," *Pattern Recognit.*, vol. 128, 2021, Art. no. 108684.

[7] C. Chen, Z. Chen, J. Zhang, and D. Tao, "SASA: Semantics-augmented set abstraction for point-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 221–229.

[8] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[9] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J.-H. Wan, and Y. Guo, "Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18931–18940.

[10] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with pointformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7459–7468.

[11] C. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[12] J. S. K. Hu, T. Kuai, and S. L. Waslander, "Point density-aware voxels for lidar 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8459–8468.

[13] J. Mao et al., "Voxel transformer for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3144–3153.

[14] S. Deng and Q. Dong, "GA-NET: Global attention network for point cloud semantic segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 1300–1304, 2021.

[15] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[16] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9276–9285.

[17] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, 2018, Art. no. 3337.

[18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12689–12697.

[19] Y. He et al., "DVFENet: Dual-branch voxel feature extraction network for 3D object detection," *Neurocomputing*, vol. 459, pp. 201–211, 2021.

[20] H. Zhang, D. Yang, E. Yurtsever, K. A. Redmill, and U. Ozguner, "Faraway-frustum: Dealing with lidar sparsity for 3D object detection using fusion," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 2646–2652.

[21] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 35–52.

[22] L. Zhao, M. Wang, and Y. Yue, "Sem-Aug: Improving camera-lidar feature fusion with semantic augmentation for 3D vehicle detection," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9358–9365, Oct. 2022.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[24] X. Wu et al., "Sparse fuse dense: Towards high quality 3D detection with depth completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5408–5417.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 7–9, 2015.

[26] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay," *CoRR*, vol. abs/1803.09820, 2018. [Online]. Available: http://arxiv.org/abs/1803.09820

[27] 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 *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 1201–1209.