Journal of Robotics and Control (JRC)
Volume 6, Issue 2, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i2.25602

Three-Dimensional Object Detection in Point Clouds with Multi-Stage Proposal Refinement Network

Jyothsna Datti [1*], Ramesh Chandra Gollapudi [2]
[1, 2] Department of Computer Science & Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Bachupally, Hyderabad, Telangana, India-500090
Email: [1] jyothsna.datti@gmail.com, [2] rameshchandra_g@vnrvjiet.in
*Corresponding Author

***Abstract*—Three-dimensional object detection in point clouds serves a vital role in autonomous driving and robotics. Point Clouds provide a vivid representation of 3D data that enables reliable object detection by acquiring the spatial distribution of points in a scene, facilitating the localization and identification of the objects within three-dimensional space. Precise localization of the objects remains challenging, particularly for moderately visible objects which attributes to inconsistent quality proposals. To tackle this, this paper presents a multi-stage proposal refinement network to generate the qualitative predictions. The research contribution is, first to improve the quality of proposals in partially visible objects, the model is integrated with 3D Resnet backbone through the refinement module at various stages. Second, to improve the quality of predictions, a confidence-weighted box voting mechanism is incorporated ensuring the precise bounding box detections. Experimentation analysis was carried out on the KITTI, NuScenes and the custom LIDAR datasets. Notably, the proposed method achieves an average precision of 82.45% for Car class, 44.94% for Pedestrian class and 66.12% for Cyclist class on the moderate category of KITTI dataset, but in the hard category with high occlusion need to be improved. On Nuscenes dataset, the model achieved mAP of 66.2%. In custom dataset, 2739 training frames, 342 frames for validation, and 343 frames for testing were taken which achieved an average precision of 82.40% for Car, 44.10% for pedestrian and 67.90% for Cyclist. The results indicate that multi-stage refinement network enhances to perform the object detection precisely, which is critical to localize and detect the target in autonomous driving and robotics.**

Keywords—Object Detection; LiDAR 3D Point Clouds; Progressive Refinement; Localization.

I. INTRODUCTION

Recent breakthroughs in 3D deep learning [1] have unveiled the new capabilities that enable machines to interpret complex spatial scenes from autonomous driving to medical imaging. While 2D deep learning only works with images [2]-[4] 3D deep learning benefits from the depth and spatial structure, enhancing its capability of capturing complex scenes through the analysis of objects in three-dimensional space. This kind of approach is essential for object detection tasks, to enable the object's position and geometry which significantly improve accuracy and safety. Deep learning systems are better equipped to handle real-world scenarios requiring precise spatial perception, such as navigating through the obstacles [5] and detection of the objects in real-time which is essential for autonomous robots and vehicles. A critical source of 3D data in this domain is 3D point clouds, a data format that represents the surfaces and edges of objects in three-dimensional space through collections of individual points. The utilization of the point clouds [6] is central to the 3D deep learning, depicting objects as distinct collection of points in three-dimensional space with their shapes and spatial relationships. The recent advancements in the autonomous vehicles and robotics gained much attention in [7]-[14], [15]-[17] and [18]-[20] particularly in lidar-based point clouds. These point clouds are acquired by the sensing technologies such as LIDAR, RADAR and RGB-Depth Cameras [21], delivering an effective and versatile approach to encoding three-dimensional information. Depth images [22] obtained from RGB-depth cameras are transformed into point cloud data to identify the positions of objects and walls within a 3D space or the real-world space. To effectively detect the objects, precise localization is important within the point clouds, which involves novel neural network architectures to handle the structures.

The evolution of 3D object detectors has significantly improved with the unique point cloud architectures for better accuracy and efficiency. From the PointNet series, the PointNet [23] and then its successor, PointNet++ [24], brought efficient methods for direct learning from raw point clouds. Building on these bases, Frustum-PointNet [25] extended the PointNet model into 3D object detection for autonomous vehicles. The latest advancements in 3D detection commonly employ either single-stage or two-stage frameworks to process data effectively. Single stage detectors streamline the object detection process without generating region proposals and directly connect to the detection head, achieving the balance between speed and accuracy. Some of the most prominent models include SSD [26] adapted for 3D detection and its variants, namely SE-SSD [27], SA-SSD [28], 3D-SSD [29] and PVT-SSD [30]. Other single stage detectors are [31]-[33]. Two-stage detectors are another variant which refines the initial proposals for higher accuracy mainly in complex environments, [34]-[39] are few efficient two-stage detectors. Object detection in 3D point clouds represents categories such as point-centric, voxel-centric, and point-voxel centric approaches. Point-centric approaches such as PointGNN [33], pointdrop [40] and improved-

PointRCNN [35] applied graph based, augmentation and segmentation-based approaches on the raw points directly. Point-Voxel approaches such as PV-rcnn [41] and pvrcnn++ [42] are effective in terms of detection capability but are highly computational. In contrast to these approaches, voxel-centric approaches balance the trade-off by organizing the unstructured point clouds in structured format. VoxelNet [31] being pioneer to voxel-centric approach; the successor methods include [18], [43]-[52] gained better performance with various strategies. Pillar-based approaches are also a type of voxel-centric approach that is light-weight. PointPillar [32] being pioneer to this kind, and the successors are [53]-[56].

Despite the progress achieved in the point clouds and object localization, one significant issue persists is the precise localization in the moderately visible objects. This is critical for real-time systems such as autonomous vehicles, robotic positioning and navigation, as it enables reliable spatial understanding and decision-making. In autonomous driving, occlusions caused by other vehicles or buildings can obscure critical information such as lane boundaries, traffic signals or potential hazards. Precise localization aids the vehicle to maintain its position and predict the moderately visible objects and navigate safely. Similarly, in robotics, occlusions from objects or machinery can disrupt the robot's view to determine its own position or the target. Precise localization in such scenarios allows robots to operate reliably and carry out the tasks such as navigating warehouses, object manipulation, working collaboratively with humans efficiently and safely. Precise localization of the objects in autonomous driving and robotics is affected by several factors. Firstly, variability in object size, shape and texture [57] complicates the consistent identification and positioning. Secondly, lighting conditions, shadows and background clutter [58][59] makes the objects difficult to detect due to the alteration of the visibility of objects. Thirdly, occlusion [59] [60], where objects are partially blocked by other objects or elements in the environment, hiding essential features necessary for precise localization. This research focuses on the precise localization in occlusion-related scenarios i.e., moderately visible objects. Precise localization is achieved by improving the quality of the 3D proposals, which is critical for 3D object detection systems because they serve initial candidate regions where the objects are likely to be located in the 3D space.

Occlusion in LiDAR data is a critical challenge for 3D object detection, where detection of moderately visible objects leads to improper proposals which is important for better detection. LGSLNet [60] addresses occlusion by leveraging the multi-view features and semantic-aware convolution to improve the 3D object detection, this still has a limitation to improve the proposal quality and reduce the inference time on lower batch sizes i.e., achieved inference of 127ms at batch size 2. AGONet [61] constructs the conceptual scenes with non-occluded objects enriching the feature representation. A re-weighting module dynamically emphasizes informative regions, refining feature adaptation to boost the detection accuracy but for highly sparse and occluded objects, the lack of proper proposals needs to improvise the detection accuracy of the model for precise localization of the objects. The fusion framework in [62] enhances object localization in occluded areas by dynamically integrating LiDAR's precise depth data with monocular vision's rich texture details. The Depth-Aware Transformer (DAT) interprets spatial hierarchies, allowing for more accurate depth estimation and object positioning even in challenging occluded environments, however, precise localization in sparse point clouds need to be improved. While these works have made progressive strides in occlusion-related challenge, there is a need to localize the objects precisely and improvement of the precision is critical for autonomous driving applications and robotics navigation. The work proposed in this paper addresses the limitations as mentioned.

A new vision for object recognition has recently emerged as a result of several domains of computer vision by Transformer [63]-[68] models, which are particularly good at learning context-based local representations. DETR [57] considered the detection as a set prediction task and performs parallel detection on images for 2D detection. DETR [67] also proposed a deformable cross-scale aggregation attention module. A 3D version of DETR is proposed by 3D-DETR [68]. Multi-stage methods were also less explored in the area of 3D point cloud detection and are proven to be very effective in 2D detection tasks. [69]-[72] applies cascade R-CNN where the inputs from previous layers are the input of the next layers for the further detection process. Another RPN network is CT3D [73] which is a channel-wise transformer with an encoder-decoder module. With Intersection over Union, [69] improves Intersection over Union by estimating the overlap between detected object and its associated ground truth object. Other work on muti-stage networks is [74]. However, multi-stage refinement is less explored in 3D object detection tasks especially for point clouds. With this motive, the current work addresses a refinement module with multiple stages for effective, precise localization and good quality predictions under partly occluded scenarios with moderately visible objects addresses the problems mentioned as above. The research contribution is three-fold:

- A multi-stage feature refinement module is integrated to the 3D Resnet backbone to improve the quality of the proposals.
- A confidence-weighted fusion for box voting is incorporated for the final detections ensuring precise bounding box detections.
- Comparison with cutting-edge methods and performance improvement to assess the proposed method.

II. METHOD

This research presents the model pertaining to 3D Object detection is as follows- 3D Resnet is the main backbone used to process the voxelized point cloud data. Firstly, the raw point clouds are distributed into volumetric cells called as voxels, and computing the local features within each voxel. These voxelized features are further processed by a 3D Resnet architecture incorporating residual connections for capturing spatial hierarchies while preserving the important details at multiple scales.

The 3D Resnet backbone outputs a three-dimensional feature map which is forwarded to Region Proposal Network (RPN) and generates initial bounding box proposals. At this stage, multi-stage refinement module (MSRM) uses the features from each proposal and refines the proposals. This iteratively enhances the bounding box predictions and confidence scores by refining the features and aggregating the attention-based information across stages. In the refinement network, attention module captures rich object characteristics across the proposal stages, which help boost distant and complex object detection. Finally, the comprehensive training objective with the combined RPN and multi-stage refinement losses optimizes the model for robust 3D object detection. Fig. 1 depicts the comprehensive architecture of the proposed model and Fig. 2 shows the flow of the multi-stage refinement network, where the model is operated under three stages of refinement.

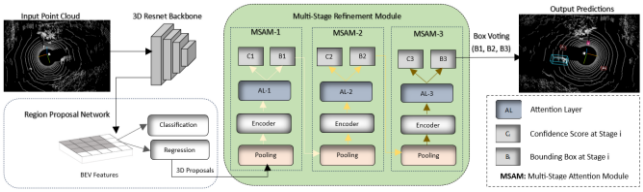

Fig. 1. Overall Architecture of the proposed method

A. 3D Resnet Backbone

1) Preprocessing and Voxelization:

The raw point cloud $P = \{p_i = 1, 2, 3, \ldots., N\}$ contains points $P_i = (x_i, y_i, z_i, f_i)$ where x_i, y_i, z_i are spatial coordinates, and f_i represents feature intensity. These points are divided into voxels $V = \{v_j \mid j = 1, 2, \ldots, M\}$. For each voxel v_j, features are aggregated by averaging the spatial and feature information as represented in equation (1).

$$v_j = \frac{1}{|P_j|} \sum_{p \in P_j} p \tag{1}$$

Where p_j denotes the points within the voxel v_j.

The voxelized data V is processed through a 3D ResNet backbone that uses sparse convolution and sub manifold convolution [75] to derive the spatial features efficiently thereby handling the sparsity in 3D point clouds. The input to the sparse 3D Resnet backbone is 3D grid of size which is computed as in the equation (2).

$$grid\_size_i = \frac{max_i - min_i}{voxel\_size_i}, for\ i \in \{x, y, z\} \tag{2}$$

where x, y, z are the three-dimensional coordinates, $\min_i$ and $\max_i$ are the boundaries along the axes, $voxel\_size_i$ is the voxel size over the axes.

The backbone generates a hierarchical feature map by progressively down sampling the input to 1x, 2x, 4x and 8x over the four convolutions used in 3D Resnet backbone. The squeeze and excitation [76] is applied after each convolution to progressively obtain the critical features. The detailed explanation of the 3D Resnet with Squeeze and excitation mentioned in [77].

The feature maps obtained after 3D backbone are depicted in Fig. 3. The higher activation points in the feature maps of the LiDAR point cloud represent the learned features with more probability for being an object.

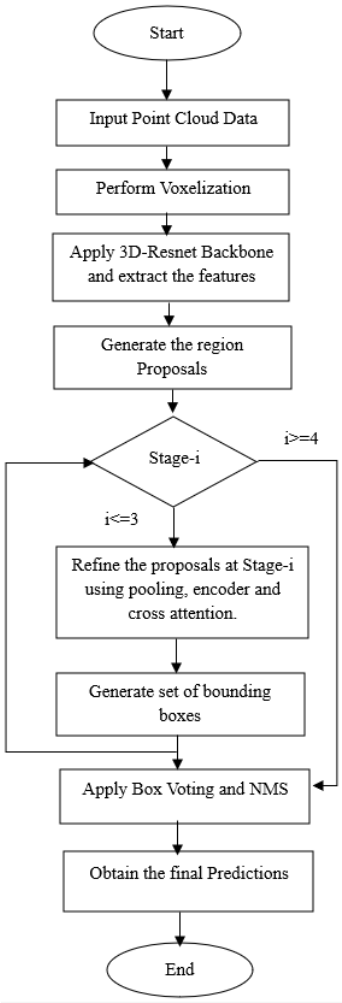

Fig. 2. 3D object detection using multi-stage refinement

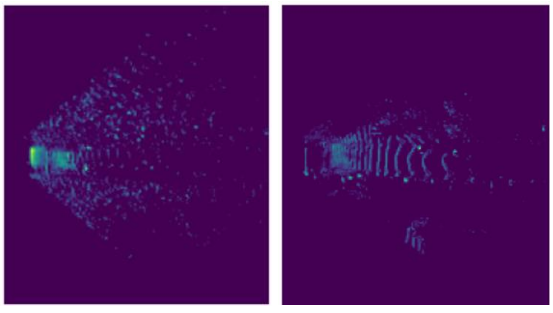

Fig. 3. Feature maps obtained at 96th and 112th iterations

B. Region Proposal Network

The approach RPN adopts is directly upon the feature maps produced by the backbone 3D ResNet on processing the input 3D point cloud data. These feature maps are compressed further along the z-axis to form a representation called Bird-Eye-View (BEV). Then, a 2D backbone that consists of a standard convolution with non-linearity is applied according to the BEV representation in order to encode the features further. Two sibling 1x1 convolutions are added on top of the 2D backbone to obtain the bounding box

offsets and categories along with 3D region proposals, which utilizes the IoU-based matching for ground-truth bounding boxes to the anchors, compute's part scores similar to [24]. The RPN generates a set of 3D proposals with initial bounding box parameters and confidence scores, which are then refined in subsequent stages. The loss computed at this stage is the loss generated at the region proposal networks defined as in [71].

The output of the RPN contains both the proposed regions, also known as Region of Interest (RoIs), and their corresponding scores that assess the possibility that a region with an object existence. The network specifically outputs the bounding boxes in coordinates such as center positions, dimensions as well as objectness scores that represent the confidence with which a proposal contains an object. These proposals are further refined by the refinement module, which involves fine-grained classification and regression.

C. Multi-Stage Refinement Module (MSRM)

The MSRM refines the initial bounding boxes generated by the RPN. For each proposal, the MSRM performs iterative refinement, updating the box coordinates and confidence scores. The MSRM is structured with multiple stages, each incorporating the multi-stage Attention Module (MSAM) to enhance the features. In this stage, a single detection head is restructured into T progressive stages, forming a multi-stage detection framework, where each stage consists of a localization adjustment branch and the confidence prediction branch. Each stage of MSAM consists of pooling layer, embedding layer and attention layer which is in Fig. 1. Let R be the initial set of proposals denoted as $R^0 = \{R_i^0\}_{i=1}^M$, where M is the total number of proposals and these proposals are iteratively refined across stages, with the proposals at t^{th} stage is $R^t = \{R_i^t\}_{i=1}^M$. When F_{3D} is the 3D features obtained from the backbone network, pooling extracts features f_m^t related to the region as in equation (3).

$$f_m^t = Pooling(F_{3D}, R_m^{t-1}) \tag{3}$$

The pooled features are encoded by the encoding layer, which contains series of operations that processes the feature maps during the encoding stage. The sequence of operations is depicted in the Fig. 4, which is a sequential operation as in equation 4. This consists of a linear layer, batchnorm1d and ReLU with dropout layer and the linear layer, batchnorm1d and ReLU.

$$F' = Sequential(f_m^t) \tag{4}$$

Each encoded feature F' is refined under t stages. The current stage feature is concatenated with the previous stage feature and so on. Finally, at the t^{th} refinement stage the current feature is computed as in equation (5).

$$F_t = [F_0', F_1', \ldots . F_t'] \tag{5}$$

For each stage, MSAM computes attention scores using queries, keys, and values derived from proposal features. The attention scores from the attention layer [78] are obtained using equation (6) where $Q_t = F_t' W_t^q$, $K_t = F_t' W_t^k$ and $V_t = F_t' W_t^v$, C' is the feature dimension and i refers to the i^{th} head in multi-head attention to improve the representational capability in learning the features.

$$F'_{(t,i)} = softmax\left(\frac{Q_{(t,i)}(K_{(t,i)})^T}{\sqrt{C'}}\right) V_{(t,i)} \tag{6}$$

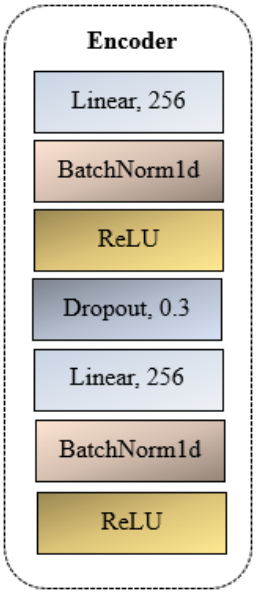

Fig. 4. Encoding layer

For H heads in multi-head attention, the feature vector is formulated by concatenating attention outputs across stages, providing a composite feature vector for bounding box refinement and score prediction as in the equation (7). The self-attention and cross attention stages are depicted in Fig. 5, where in the initial stage i.e., stage-1 features obtained from current stage only were taken. In stage-2, features from stage-1 have been concatenated to the stage-2. Similarly, for Stage-3, features from Stage-1 and stage-2 are concatenated to Stage-3 through which refinement is done at multiple stages.

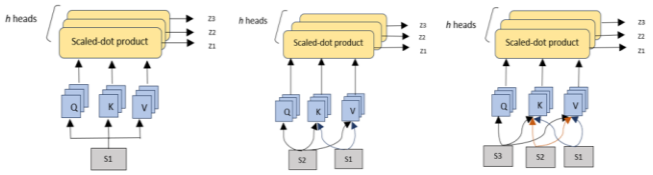

Fig. 5. Self-Attention and Cross-Attention at Stage-1(S1), Stage-2(S2) and Stage-3(S3)

$$F_t = Concat(F'_{(t,0)}, F'_{(t,1)} \ldots F'_{(t,H)}) \tag{7}$$

The thresholds at each stage for the objects Car, Pedestrian and Cyclist for the refinement are $u_1 = [0.5, 0.45, 0.45]$, $u_2 = [0.55, 0.5, 0.5]$ and $u_3 = [0.6, 0.55, 0.55]$. The thresholds are adjusted progressively in each stage such that stage-1 focuses to learn broader set of data whereas in the stage-2 model focuses on learning the hard samples with higher threshold compared to stage-1. Similarly, stage-3 foreground samples are refined with stricter threshold as mentioned in u_1, u_2 and u_3 respectively. The thresholds are set in this manner since the 3D object detection evaluation is done at 0.7 for Cars and 0.5 for

Pedestrians and Cyclist classes. Multi- Layer Perceptron is processed layer wise using linear transformation and GELU activation function which finally represents the bounding box values and confidence scores.

The pooling is performed by RoI grid pooling which extracts features from 3D grid of size G×G×G. The number of RoIs being R, T is the total number of grid points ($R.G^3$), *K* be the average number of neighbors for each grid point, C be the dimension of the grouped features where C is voxel feature dimension and 3 is the dimension of the relative coordinates, C' is the dimension of the output after the encoder layer the computational complexity is measured as O(T.K.(C+3).C'). These grouped voxels occupy more memory when encoder layer is applied. Hence, to mitigate this, a decomposition step is applied where voxel features and the relative coordinates are processed in separate streams. Firstly, voxel features are processed independently (O (N.C.C')). Secondly, the relative coordinates are processed fewer computations for the relative coordinates O(M·K·3·C′)). Finally, the RoI grid pooling is computed as O (N.C.C'+M.K.3.C') which reduces the number of computations for each type of feature, leading to better computational efficiency. For S stages and the cross attention, the computational complexity is O (R. C^2. S). The total computational complexity of the multi stage refinement module is the O (N.C.C'+M.K.3.C'+ R. C^2. S).

D. Box Voting

3D object detection is a challenge with the need to ensure the quality of predictions precisely. To further improve the accuracy of predictions, box voting strategy has been implemented among the stages. NMS-based methods usually suppress the detections of low confidence and there by retains the high confidence bounding boxes. Also, this has been implemented with the motivation of strong and weak predictions at each stage are ensembled to obtain correct detections. This is mitigated by adopting the confidence-weighted box which computes the average of the detection confidences and aligns the boxes with the respective detection confidences as in equation (8). Box regression is performed similar to voxel-rcnn [37] and pv-rcnn [41]. To compute the confidence score, bilinear interpolation was adopted to extract the corresponding features and these features are averaged to produce the scalar confidence value for each of the anchors, making the predictions consistent.

$$V = \frac{\sum_{i=1}^{m} C_i . B_i}{\sum_{i=1}^{m} C_i} \quad (8)$$

Where C_i is the merged confidence and B_i is the bounding box parameters of i^{th} proposal and m is the total number of proposals.

After bounding boxes merging and box voting, there will be some high quality and redundant boxes. To remove this, we perform Non-Max-Suppression (NMS) on these boxes to predict the final bounding boxes. The procedural flow for bounding boxes merging, voting and NMS was depicted in Fig. 6. With this approach, several predictions obtained from multiple scales as well as high/low confidence are combined to form more accurate and reliable final outcomes.

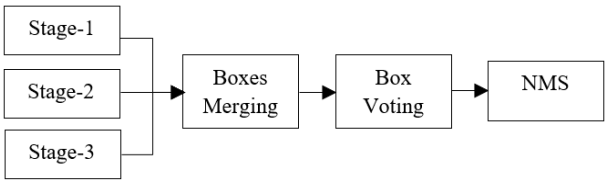

Fig. 6. Wok Flow of Box Merging, Voting and NMS

As discussed in Table X, the inference time increases progressively with each stage of the model. While the detection performance at the fourth stage is nearly similar to that of the third stage, this marginal difference does not justify the additional computational cost. Since the performance of the third-stage and fourth-stage models is nearly identical, we prioritize computational efficiency and adopted the three-stage model. This configuration achieves a performance with a running speed of 66.40ms per frame on a single NVIDIA 4080 GPU, striking an optimal balance between the precision and computational complexity.

III. RESULTS AND DISCUSSIONS

A. Datasets and Implementation Details

KITTI Dataset: The experimentation was performed on the KITTI [79] 3D object detection dataset. The dataset with 7481 frames was taken, which was split into 3712 frames of data for training and 3769 frames of data for testing. In addition to this, custom split [80] on the training set of 3712 frames in a 5:1 ratio of training and validation samples is done respectively. The point clouds are clipped in the range of [0m,70.4m] in the X-axis, [-40m, +40m] in the Y-axis and [-3m, +1m] in the Z-axis. The voxel size is set as (0.05m, 0.05m, 0.05m) with RoI per Image 160 at each stage. To compute average precision, the IoU thresholds are set at 0.7 for cars, 0.5 for pedestrians, and 0.5 for cyclists. The proposed model was trained on a single RTX 4080, NVIDIA machine with 16GB graphics memory and 32GB RAM with 80 epochs, batch size of 4 and the Adam one cycle optimizer with the learning rate of 0.01 and weight decay of 0.01. The detailed hyperparameters explanation is mentioned in Appendix.

NuScenes Dataset: The experimentation was performed on the NuScenes [81] 3D object detection dataset. The dataset with 28130 samples was taken in which 22504 samples are split into training set, 5626 samples for validation and 6019 samples for testing. For NuScenes dataset, point clouds are clipped in the range of [-54m, +54m] in the X-axis and Y-axis, [-5m, +3m] in the Z-axis. The voxel size is (0.075m, 0.075m, 0.2m) with RoI per Image 160. To compute average precision, the IoU thresholds are set at 0.7 for cars, 0.5 for pedestrians, and 0.5 for cyclists. The proposed model was trained on a single RTX 4080, NVIDIA machine with 16GB graphics memory and 32GB RAM with 40 epochs, batch size 2 and utilized Adam one cycle optimizer with the learning rate of 0.01 and weight decay of 0.01.

Custom Dataset: The dataset was collected using the 32-channel ouster LIDAR, which is acquired in the outdoor space near the host institution. The LIDAR has an operating mode of 1024 towards horizontal and 32 towards vertical mode at 10 frames per second, with a horizontal field of view as 360^0 and vertical field of view as 45^0. The dataset is point-

cloud dataset which comprises of 2739 training frames, 342 frames for validation, and 343 frames for testing. The dataset is in .npy format with x, y, z and reflectance values in each frame. For the custom dataset, point clouds are clipped in the range of [-54m, +54m] in the X-axis and Y-axis, [-5m, +3m] in the Z-axis. The voxel size is (0.05m, 0.05m, 0.05m). To compute average precision, the IoU thresholds are set at 0.7 for cars, 0.5 for pedestrians, and 0.5 for cyclists. The experimentation was done on the pseudo labels obtained by the KITTI dataset. The training of the proposed model is done on a single NVIDIA GeForce RTX 4080 consisting of 16GB GPU memory and 32GB of RAM with 80 epochs, employing a batch size of 2 and a learning rate of 0.01 with the Adam One Cycle optimizer with weight decay of 0.01.

B. Evaluation of the proposed Method

Evaluation on the KITTI dataset: The proposed model is assessed on the three objects Car, Pedestrian and Cyclist which in turn has three categories for each object – Easy, Moderate and Hard. Experimentation was done on the proposed two stage network on 3DResnet, 3DResnet integrated to SE module at input stage only and 3DResnet integrated to SE module at all layers with multi-stage refinement under various stages. Fig. 7. depicts the mean average precision results for the objects "Car", "Pedestrian" and "Cyclist" on the validation set. Here, the 3D Resnet backbone is experimented on the CT3D based RPN and also the proposed multi-stage attention module with and without box voting.

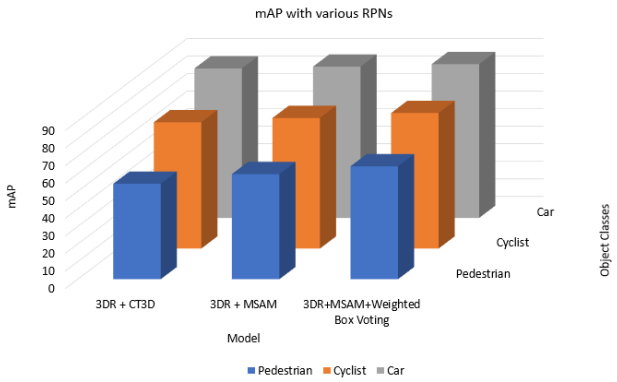

Fig. 7. Comparison of mean Average Precision on 3D Resnet with various RPNs on the three classes

As discussed in Table I, Table II and Table III, the proposed method is compared to the state-of-the-art models on the KITTI test split which progresses the average precision over the existing methods especially in moderate and hard categories of 3D detection. Compared to one of the recent works VoxelNeXt, the proposed model improvises the Car moderate and hard categories by 2.1% and 0.62% respectively, Pedestrian moderate and hard categories by 2.22% and 1.32% respectively, Cyclist moderate and hard categories by 0.83% and 2.12% respectively. Compared to another recent work pvt-ssd [30] especially for Car detection, the results were also improvised under moderate and hard categories by 0.16% and 0.67% respectively. The underlined value represents the second highest detection value in the moderate category where Car has an improvement in average precision of 0.68%, Cycle has an improvement of 0.08% and Cyclist has an improvement of 0.17%. Table IV shows the performance comparison with the average precision metric on the validation set for the Car class, which yielded better results over the state-of-the-art. Also, the recall is more compared to the state-of-the-art approaches as shown in Fig. 8, yielding to 78.1%. The main findings of the present study indicate the precise object localization for the moderated visible objects with the average precision of 82.45% for Car, 44.94% for Pedestrian and 66.12% for Cyclist classes.

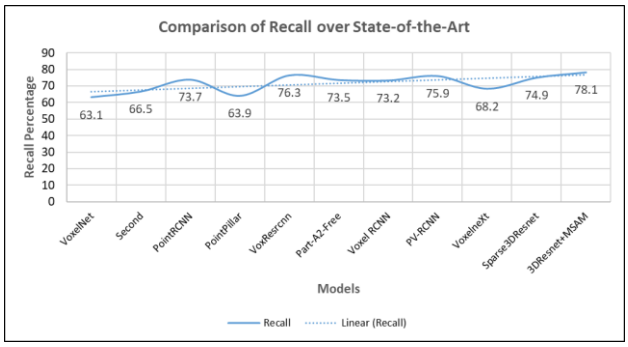

Fig. 8. Comparison of Recall over state-of-the-Art

Fig. 9 depicts the prediction of the objects with classes "Car", "Pedestrian" and "Cyclist" classes on the proposed method. The car object is indicated by green; the pedestrian object is indicated by blue and the cyclist object is indicated by yellow. The model has been tested and visualized at a 0.7 threshold and is able to localize and detect the objects precisely. Fig. 10 shows the comparison of the most recent model VoxelNext in terms of visualization for three samples of the KITTI dataset. The visualization is depicted in three parts – the first one is the original image of the point cloud; the second one is the visualization of the frame using the VoxelneXt model and the third one is the predicted visualization of the proposes multi-stage attention module. The highlighted red symbol indicates the false negative of the VoxelneXt model which suffers from missing detections for long range detections of 0-70m for KITTI dataset. This also depicts the proposed model's ability in terms of precise localization of the distant objects. Also, due to the implementation of transformer-based approach the model is able to localize and detect the objects precisely.

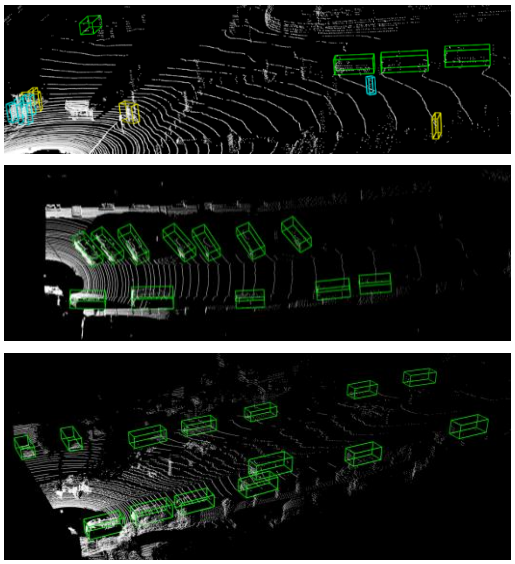

Fig. 9. Qualitative Predictions on the KITTI dataset

TABLE I. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC ON THE CAR CLASS - KITTI TEST SPLIT

Model Name	Car% (R40)		
	E	***M***	***H***
VoxelNet [31]	81.97	65.46	62.85
Second [39]	83.13	73.66	66.2
PointRCNN [34]	85.94	75.76	68.32
PointPillar [32]	79.05	74.99	68.3
PointGNN [33]	88.33	79.47	72.29
Part-A2[36]	85.94	77.86	72
PV-RCNN [41]	90.25	81.43	76.82
Voxel RCNN [37]	90.76	81.69	77.42
SparseDet [48]	90.79	81.17	78.11
CT3D [67]	87.83	81.77	77.16
VoxelNeXt [44]	89.1	80.35	76.9
Proposed Model (ours)	89.81	82.45	77.52

TABLE II. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC ON THE PEDESTRIAN CLASS - KITTI TEST SPLIT

Model Name	Pedestrian% (R40)		
	E	***M***	***H***
VoxelNet [31]	57.86	53.42	48.87
Second [39]	51.07	42.56	37.29
PointRCNN [34]	49.43	41.78	38.63
PointPillar [32]	52.08	43.53	41.49
PointGNN [33]	51.92	43.77	40.14
Part-A2[36]	54.49	44.5	42.36
Voxel RCNN [37]	52.57	44.86	39.09
PV-RCNN [41]	52.17	43.29	40.29
SparseDet [48]	52.92	44.61	41.8
VoxelNeXt* [44]	52.1	42.72	39.08
Proposed Model (ours)	51.89	44.94	40.4

TABLE III. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC ON THE CYCLIST CLASS - KITTI TEST SPLIT

Model Name	Cyclist% (R40)		
	E	***M***	***H***
VoxelNet [31]	67.17	47.65	45.11
Second [39]	70.51	53.85	46.9
PointRCNN [34]	73.93	59.6	53.59
PointPillar [32]	75.78	59.07	52.92
PointGNN [33]	78.6	63.48	57.08
Part-A2[36]	78.58	62.73	57.74
Voxel RCNN [37]	77.54	64	53.15
PV-RCNN [41]	78.6	63.71	57.65
SparseDet [48]	81.93	65.95	60.41
VoxelNeXt [44]	81.33	65.31	57.43
Proposed Model (ours)	81.62	66.12	59.55

TABLE IV. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC ON THE CAR CLASS - KITTI VAL. SPLIT

Model Name	Car % (R40)		
	E	***M***	***H***
VoxelNet [31]	81.97	65.46	62.85
Second [39]	87.43	76.48	69.1
PointRCNN [34]	88.88	78.63	77.38
PointGNN [33]	87.89	78.34	77.38
Part-A2 [36]	89.47	79.47	78.54
PV-RCNN [41]	92.57	84.83	82.69
Voxel RCNN [37]	92.38	85.29	82.86
SparseDet [48]	93.81	84.78	84.33
CT3D [67]	92.85	85.82	83.46
VoxelNeXt [44]	92.51	84.35	82.71
Proposed Model (ours)	93.01	**85.97**	**83.63**

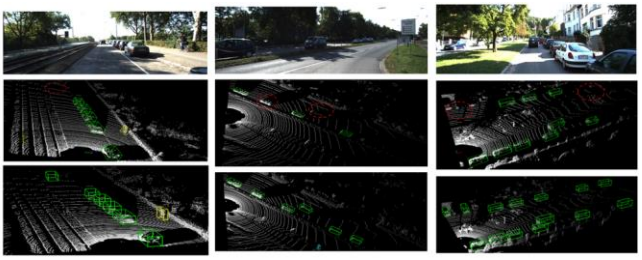

Fig. 10. Comparison of Visualization between VoxelNeXt and the proposed MSAM model for missing object detection. First row represents the Image corresponding to the point cloud, second row represents detections on VoxelNeXt, third row represents the detections on the proposed model

Evaluation on the NuScenes dataset: The proposed model is evaluated on the objects Car, Truck(Trk), Bus, Trailer(Trl), Construction Vehicle(C.V), Pedestrian(Ped), Bicycle(Byc), Motor Cycle(Mot), Traffic Cone(T.C) and Barrier(Bar) on the NuScenes dataset. To facilitate training and testing of the proposed model on NuScenes dataset, the detection head is replaced with center head similar to VoxelneXt [44] and Center Point [49] methods.

Table V depicts the average precision results on the NuScenes test split. Compared to VoxelNeXt, the proposed model has an improvement of 3% on the Car class, 6.1% improvement in Trailer class and 1.3% improvement on Motor Cycle class. As mentioned in Table V, the proposed method achieved better performance on the benchmarks for the mean average precision(mAP) metric. The underlined value represents the second highest detection model corresponding to the particular class. Bus, Truck, Pedestrian, Bicycle, Traffic Cone and Barrier are the second highest detectors. Table VI depicts the performance of average precision on the NuScenes Validation split, which represents superior over other methods.

Evaluation on the Custom dataset: The custom dataset was acquired as mentioned in the section 4.1.3 and annotations for 500 frames were done using the Computer Vision Annotation Tool (CVAT), which is a lidar-based 3D object detection annotation tool for point clouds. The other frames labels were the pseudo labels obtained from the KITTI dataset. Initially, the labelled data was fine-tuned using the pre-trained model of the proposed work with the aid of OpenPCDet [86]. On the unlabeled data, pseudo labels were obtained from the predictions. At this stage, retain the labels with high confidence score and discard the remaining labels. The criteria considered here is 0.7 for Car, 0.5 for pedestrian and 0.5 for cyclist objects. Finally, the predictions obtained from the labelled data and the predictions obtained from the pseudo labels are combined and the model is trained on the combined dataset and evaluated for the average precision metric. This is iterated until the desired performance is met. In this case, two iterations were done since the desired predictions are obtained with comparable performance to KITTI dataset at the second iteration itself.

TABLE V. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC– NUSCENES TEST SPLIT

Method	Car	T.C	Ped	Mot	C.V	Trl	Bar	Byc	Bus	Trk	mAP
Point Pillar [32]	68.4	30.8	59.7	27.4	4.1	23.4	38.9	1.1	28.2	23	30.5
3D-SSD [29]	81.2	31.1	70.2	36	12.6	30.5	47.9	8.6	61.4	47.2	42.7
CBGS [82]	81.1	70.9	80.1	51.5	10.5	42.9	65.7	22.3	54.9	48.5	52.8
CenterPoint [49]	84.6	76.7	83.4	53.7	17.5	53.2	70.9	28.7	60.2	51	58
Focals-Conv [83]	86.7	81.4	**87.5**	64.5	23.8	59.5	74.1	36.3	**67.7**	56.3	63.8
PillarNet [56]	87.4	**82.1**	87.2	67.4	**30.4**	61.8	**76**	40.3	60.9	56.7	65
VoxelNeXt [44]	84.6	79	85.8	73.2	28.7	55.8	74.6	**45.7**	64.7	53	64.5
Ours	**87.6**	81.7	87.4	**74.5**	28.3	**61.9**	75.3	43.6	65.3	56.4	66.2

TABLE VI. 3D OBJECT DETECTION PERFORMANCE COMPARISON USING AVERAGE PRECISION METRIC – NUSCENES VAL SPLIT

Method	Car	Mot	C.V	Bar	T.C	Ped	Bus	Byc	Trl	Trk
SECOND [39]	81.8	42.5	15	59.2	57.4	77.7	66.9	17.5	37.3	51.7
CenterPoint [49]	85	58.8	15.5	67.1	70	85.3	69.5	40.9	35.7	58.2
WYSIWYG [84]	80	18.5	7.5	34.5	27.9	66.9	54.1	0	28.5	35.8
Transfusion [85]	86.9	72.9	25.2	70.3	77.2	87.5	73.1	57.3	43.4	60.8
AGONet [61]	81.5	32.5	13.3	51.2	48.1	72.2	62.2	5.9	34	50.1
PillarNeXt [54]	84.8	68.4	21.8	68.2	74.2	86.1	68.3	56.5	37.1	58
VoxelNeXt [44]	85.6	59.7	17.9	68.1	70.8	85.4	71.6	43.4	38.6	58.4
Ours	88.2	74.8	28.9	77.1	84	89.2	66.2	44.6	63.2	57.3

Refer to Table VII for the iteration-wise results over validation set and Table VIII on the test set. Refer to Fig. 11 for the iteration-wise average precision on the validation split. Table IX shows the detailed performance comparison of precision, recall and f1-score metrics over iterations on the test split for the classes Car, Pedestrian and Cyclist. Fig. 12 clearly shows the visualization of the predictions after final refinement and detection of the distant objects. Various markers are depicted to represent the distance – purple cone represents the range up to 20m, orange represents the range up to 40m, blue cone to 60m and magenta cone to 80m. With the implementation of transformer-based multi-stage approach, the model is able to detect the distant objects also. The major challenge faced while working on the custom dataset is pseudo label processing, where the labels generated initially were noisy. To mitigate this issue, pseudo labels are refined iteratively with confidence-based thresholding to obtain the qualitative labels.

TABLE VII. ITERATION-WISE MAP ON THE VALIDATION SPLIT FOR THE 3D RESNET WITH MSRM MODEL ON THE CUSTOM DATASET USING PSEUDO LABEL REFINEMENT

Iteration No.	Car %	Ped. %	Cyclist %
Iteration-1	82.37	45.06	67.98
Iteration-2	84.51	47.02	69.58

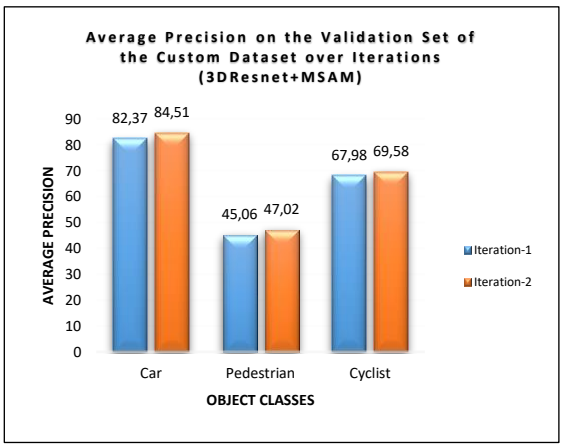

Fig. 11. Average Precision on the Validation Set for each iteration

TABLE VIII. ITERATION-WISE AVERAGE PRECISION RESULTS ON THE TEST SET OF THE CUSTOM DATASET

Model	Car %	Ped. %	Cyclist %
Iteration-1	81.3	43.5	66.8
Iteration-2	82.4	44.1	67.9

TABLE IX. 3D OBJECT DETECTION PERFORMANCE COMPARISON OF PRECISION, RECALL AND F1-SCORE METRICS ON THE CUSTOM DATASET

Iteration	Class	Precision	Recall	F1-Score
Iteration-1	Car	81.30	83.40	82.34
	Pedestrian	43.50	66.51	52.60
	Cyclist	66.80	79.30	72.52
Iteration-2	Car	82.40	84.70	83.53
	Pedestrian	44.10	69.40	53.93
	Cyclist	67.90	81.20	73.96

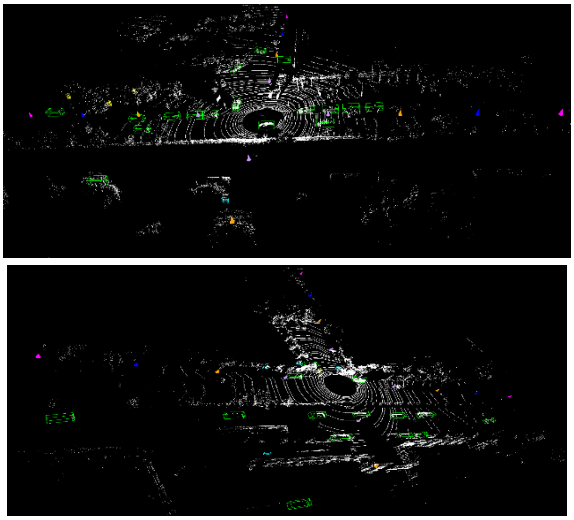

Fig. 12. Visualization of predictions over ranges

C. Ablation Study

Refinement Stages: Initially, the model has been tested with multiple stages as shown in Table X, which depicts the average precision results at various stages of the multi-stage refinement network. The improvement of proposals has been observed stage-wise for the Easy, Moderate and Hard categories for the three stages and in the fourth stage, similar

performance is observed. For the efficiency of computational complexity, three stages are adopted with the inference of 78.93ms on a single 4080 GPU.

TABLE X. STAGE-WISE AVERAGE PRECISION RESULTS ON THE KITTI DATASET FOR CAR AND CYCLIST CLASSES

Stage	Car			Cyc.			Inf. Time (ms)
	E	M	H	E	M	H	
1	88.84	80.79	75.12	80.32	64.81	57.92	60.40
2	89.32	81.48	76.46	81.29	65.14	58.78	66.35
3	89.81	82.45	77.52	81.62	66.12	59.55	78.93
4	89.97	82.41	77.43	81.78	66.09	59.48	101.56

Post-Processing Methods: To verify the effectiveness of the box voting strategy, the proposed method has been tested on the 3D Resnet directly with NMS, 3D Resnet integrated with multi stage refinement module with NMS and 3D Resnet integrated with multi-stage refinement module with box voting followed by NMS. The results have shown that the model yields better with box voting with NMS rather than NMS alone. Since the predicted boxes at each stage has strong detections as well, box voting has contributed to the improvement of the average precision for the mentioned classes in Table XI. The key findings of this study indicate that with the confidence-weighted box voting, there is an improvement in the average precision of the classes in all the categories.

TABLE XI. ABLATION EXPERIMENTS ON THE BOX VOTING STRATEGY IN MSRM WITH 3D RESNET BACKBONE

MS	Car			Ped.			Cyc.		
	E	M	H	E	M	H	E	M	H
3DR	88	79.2	76.2	47.3	39.9	37.7	77.9	62.2	55.8
+MN	89.1	82.3	77.2	51.2	44.8	40	81.5	65.9	58.2
+MB	89.8	82.4	77.5	51.8	44.9	40.4	81.6	66.1	59.5

Note: (3DR): 3D Resnet (+MN): 3DR+**M**ultiStage Refinement Module + **N**MS (+MB): 3DR+**M**ultiStage Refinement Module+**B**ox Voting with NMS

D. Inference Analysis

The analysis of the inference time of all the models discussed in Table XII are obtained from [60]. The proposed method has been inferred with the batch sizes of 2 and 4 yielding to the inference time of 78.93 and 66.40 respectively. In comparison with LGSLNet [60], which is LiDAR-only method, the proposed method achieved better inference time at both the batch sizes 2 and 4. Additionally, the model not only reduces the inference time but also improved the detection accuracy in terms of average precision for moderately visible objects.

TABLE XII. COMPARISON ON THE KITTI DATASET. L-LIDAR-ONLY L+C: LIDAR+CAMERA, AP-AVERAGE PRECISION OF CAR (MODERATE CATEGORY)

Method	Modality	Batch-Size	AP-Car (%)	Inference Time (ms)
Voxel-RCNN	L	8	81.69	26.1
PDV	L	8	81.86	54.2
Part-A2	L	8	77.86	35.8
PV-RCNN	L	8	81.43	55.4
Focals-Conv	L+C	8	82.28	158.9
LGSLNet	L	8	82.16	81
		4		96
		2		127
MSAM-proposed	L	4	82.45	**66.4**
		2		**78.93**

E. Limitation

Small Object Detection: Fig. 13 depicts the two cases describing the failure cases of the model where the first part represents the detection capability of the model in normal case and the second part represents the missing detection in occluded scenario. This poses a limitation in detecting the small objects due to the low-resolution of the point cloud for small objects, capturing the fine-grained details should be improvised to boost the detection accuracy in pedestrians.

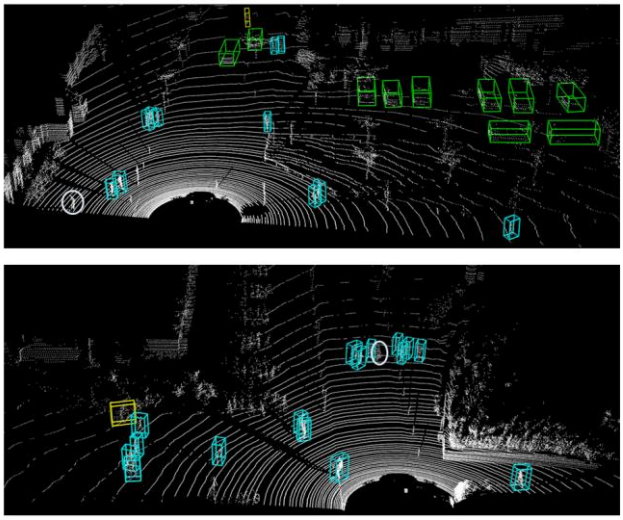

Fig. 13. Pedestrian detection results on the KITTI dataset

The main findings of this study indicate good quality proposals obtained through the refinement of proposals stage-wise while achieving good performance of 82.45% for Car class and 66.12% for Cyclist class in moderately visible objects. The improved precision and recall can be attributed to two key innovations. First is, integration of 3D Resnet with multi-stage refinement network improves the proposal quality. Second is, with box voting strategy, the performance of the predictions was improved w.r.t the average precision. Also, the inference time of the model is reduced compared to the methods mentioned in Table VII.

IV. CONCLUSION

In this work, good quality proposals are obtained by using multi-stage refinement integrated with a 3D ResNet backbone in 3D object detection. The proposals are refined stage-wise using the pooling and attention mechanism for the precise localization as well as for the detection of partly visible and distant objects. 3D Resnet-based backbone was incorporated in the voxel-driven approach which contributes rich, hierarchical feature maps that capture both local and global contextual information about the scene, which in turn aids the RPN in generating high-quality proposals. With the aggregation of multi-stage refinement mechanism and box voting, the model improvised the average precision of 82.45% for Car, 44.94% for pedestrian and 66.12% for Cyclist for the moderately visible objects and results demonstrated that this study provides valuable insights into detecting the objects in 3D point cloud data.

The proposed method improvised the precision and recall of the objects; especially in Car and Cyclist detection, suitable for autonomous driving applications. Also, the

method's ability to precisely localize the objects and accurately detect can be applied in robotic navigation and positioning including safety assessment. To improve the detection capability of small objects, further research can be extended to multi-modal inputs to improve the accuracy of small objects i.e., Pedestrian to capture the fine-grained details of the small objects.

ACKNOWLEDGMENT

The researchers would like to acknowledge DRDO ER&IPR grant no. ERIP/ER/202108001/M/01/1786 for the procurement of LIDAR and the host institute for providing the required computational resources for working on this project and also for consistently encouraging to do the research work.

APPENDIX

In Appendix section, we discuss the hyperparameter tuning with experimental results.

The hyperparameters of the model are chosen empirically based on the recall rate and Inference time. The model has been tested with voxel size [0.05,0.05,0.05] with various RoI per image. RoI per Image is one of the hyperparameters causing trade-off between average precision and inference time. Table XIII shows the average precision of the model at various RoIs at the learning rate of 0.01.

TABLE XIII. AVERAGE PRECISION RESULTS WITH ROIS

RoI	Car			Pedestrian			Cyclist		
	E	M	H	E	M	H	E	M	H
120	84.6	78.4	76.6	46.7	39.2	37.9	78.7	64.9	56.4
140	89.3	82.5	77.8	49.1	40.2	38.1	80.3	66.3	60
150	88.7	82.3	77.6	50.6	41.6	38.2	79.2	65.1	58.6
160	89.8	82.5	77.5	51.9	44.9	40.4	81.6	66.1	59.6

Table XIV shows the inference time, memory usage, mean average precision(mAP) and recall at various RoIs. It is observed that a smaller number of RoIs require a larger feature map per RoI to maintain sufficient information, causing an increase in memory usage. More RoIs contribute to efficient packed feature maps optimizing the memory usage. Also, the recall rate tends to decrease with smaller number of RoIs. With respect to mAP, the detection capability for RoI 140 for moderate and hard categories improved but there is a downfall in the detection of small objects pedestrian. With respect to inference time, the model has better inference time at RoI 140 but looking into mAP and recall, there is no significant improvement in inference time compared to the results obtained with RoI 160. However, there is a tradeoff between inference time and precision-recall with slight increase in inference time balancing mean average precision and recall at RoI 160 which is chosen as the best hyperparameter of the model.

TABLE XIV. PERFORMANCE COMPARISON: MAP, RECALL, INFERENCE TIME, AND MEMORY USAGE

RoI	mAP (%)	Recall (%)	Infer. Time (ms)	Memory (GB)
120	64.3	38.01	62.14	3.9
140	64.84	75	62.99	3.5
150	64.66	75	63.09	3.3
160	66.03	78.1	64.4	2.8

REFERENCES

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021, doi: 10.1109/tpami.2020.3005434.

[2] O. Bouazizi *et al.*, "Road object detection using SSD-MobileNet algorithm: Case study for real-time ADAS applications," *Journal of Robotics and Control (JRC)*, vol. 5, no. 2, pp. 551–560, 2024, doi: 10.18196/jrc.v5i2.21145.

[3] D. Jyothsna and G. R. Chandra, "Multi-Object Detection in 3D Point Cloud's Range Image Using Deep-Learning Technique," *14th International Conference on Cloud Computing, Data Science &Engineering (Confluence)*, pp. 401–406, 2024, doi: 10.1109/confluence60223.2024.10463402.

[4] P. Chotikunnan, T. Puttasakul, R. Chotikunnan, B. Panomruttanarug, M. Sangworasil, and A. Srisiriwat, "Evaluation of Single and Dual image Object Detection through Image Segmentation Using ResNet18 in Robotic Vision Applications," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 263–277, 2023, doi: 10.18196/jrc.v4i3.17932.

[5] J. Mao, S. Shi, X. Wang, and H. Li, "3D Object Detection for Autonomous Driving: A Comprehensive Survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 1909–1963, 2023, doi: 10.1007/s11263-023-01790-1.

[6] W. Zimmer *et al*., "A survey of robust 3D object detection methods in point clouds," *arXiv preprint arXiv:2204.00106*, 2022.

[7] Z. Song *et al*., "VoxelNextFusion: A Simple, Unified, and Effective Voxel Fusion Framework for Multimodal 3-D Object Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023, doi: 10.1109/tgrs.2023.3331893.

[8] L. Wang *et al*., "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3781–3798, 2023, doi: 10.1109/tiv.2023.3264658.

[9] Z. Song *et al*., "Robustness-Aware 3D Object Detection in Autonomous Driving: A Review and Outlook," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 15407-15436, 2024, doi: 10.1109/TITS.2024.3439557.

[10] S. Xu, F. Li, Z. Song, J. Fang, S. Wang and Z. -X. Yang, "Multi-Sem Fusion: Multimodal Semantic Fusion for 3-D Object Detection," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-14, 2024, doi: 10.1109/TGRS.2024.3387732.

[11] L. Wang *et al*., "Multi-Modal and Multi-Scale Fusion 3D Object Detection of 4D Radar and LiDAR for Autonomous Driving," in *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 5628-5641, 2023, doi: 10.1109/TVT.2022.3230265.

[12] L. Wang *et al*., "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy," in *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3781-3798, 2023, doi: 10.1109/TIV.2023.3264658.

[13] H. Xie, W. Zheng, Y. Chen, and H. Shin, "Camera and LiDAR-based point painted voxel region-based convolutional neural network for robust 3D object detection," *Journal of Electronic Imaging*, vol. 31, no. 5, 2022, doi: 10.1117/1.jei.31.5.053025.

[14] L. Wang *et al*., "Fuzzy-NMS: Improving 3D Object Detection With Fuzzy Classification in NMS," in *IEEE Transactions on Intelligent Vehicles*, 2024, doi: 10.1109/TIV.2024.3409684.

[15] X. Zhang *et al*., "RI-Fusion: 3D Object Detection Using Enhanced Point Features With Range-Image Fusion for Autonomous Driving," in *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-13, 2023, doi: 10.1109/TIM.2022.3224525.

[16] X. Zou, H. He, and C. Li, "Three-dimensional point cloud registration algorithm based on the correlation coefficient," *Journal of Electronic Imaging*, vol. 32, no. 1, 2023, doi: 10.1117/1.jei.32.1.013010.

[17] L. Fan, F. Wang, N. Wang and Z. Zhang, "FSD V2: Improving Fully Sparse 3D Object Detection with Virtual Voxels," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024, doi: 10.1109/TPAMI.2024.3502456.

[18] L. Wang *et al*., "SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving," *Knowledge-Based Systems*, vol. 259, p. 110080, 2023, doi: 10.1016/j.knosys.2022.110080.

[19] Z. Song, H. Wei, L. Bai, L. Yang, and C. Jia, "GraphAlign: Enhancing Accurate Feature Alignment by Graph matching for Multi-Modal 3D Object Detection," *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3335–3346, 2023, doi: 10.1109/iccv51070.2023.00311.

[20] Z. Song, C. Jia, L. Yang, H. Wei, and L. Liu, "GraphAlign++: An Accurate Feature Alignment by Graph Matching for Multi-Modal 3D Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 4, pp. 2619–2632, 2024, doi: 10.1109/tcsvt.2023.3306361.

[21] Y. Liu, S. Meng, H. Wang, and J. Liu, "Deep learning based object detection from multi-modal sensors: an overview," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 19841–19870, 2023, doi: 10.1007/s11042-023-16275-z.

[22] Le, Van-Hung. "Visual Slam and Visual Odometry Based on RGB-D Images Using Deep Learning: A Survey," *Journal of Robotics and Control (JRC)* vol. 5, no. 4, 2024, doi:10.18196/jrc.v5i4.22061.

[23] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017, doi: 10.1109/cvpr.2017.16.

[24] 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.

[25] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018, doi: 10.1109/cvpr.2018.00102.

[26] W. Liu *et al*., "SSD: Single Shot MultiBox Detector," *Computer Vision – ECCV*, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[27] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud," 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14489–14498, 2021, doi: 10.1109/cvpr46437.2021.01426.

[28] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure Aware Single-Stage 3D Object Detection From Point Cloud," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11870–11879, 2020, doi: 10.1109/cvpr42600.2020.01189.

[29] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11040-11048, 2020, doi: 10.1109/cvpr42600.2020.01105.

[30] H. Yang et al., "PVT-SSD: Single-Stage 3D Object Detector with Point-Voxel Transformer," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13476–13487, 2023, doi: 10.1109/cvpr52729.2023.01295.

[31] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018, doi: 10.1109/cvpr.2018.00472.

[32] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection From Point Clouds," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12697-12705, 2019, doi: 10.1109/cvpr.2019.01298.

[33] W. Shi and R. Rajkumar, "Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1708–1716, 2020, doi: 10.1109/cvpr42600.2020.00178.

[34] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3D Object Proposal Generation and Detection From Point Cloud," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–779, 2019, doi: 10.1109/cvpr.2019.00086.

[35] K. Fukitani et al., "3D object detection using improved PointRCNN," *Cognitive Robotics*, vol. 2, pp. 242–254, 2022, doi: 10.1016/j.cogr.2022.12.001.

[36] 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," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 8, pp. 2647-2664, 1 Aug. 2021, doi: 10.1109/TPAMI.2020.2977026.

[37] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1201–1209, 2021, doi: 10.1609/aaai.v35i2.16207.

[38] 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," *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2703–2712, 2021, doi: 10.1109/iccv48922.2021.00272.

[39] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018, doi: 10.3390/s18103337.

[40] W. Ma, J. Chen, Q. Du, and W. Jia, "PointDrop: Improving Object Detection from Sparse Point Clouds via Adversarial Data Augmentation," *25th International Conference on Pattern Recognition (ICPR)*, pp. 10004–10009, 2021, doi: 10.1109/icpr48806.2021.9412691.

[41] S. Shi et al., "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10526–10535, 2020, doi: 10.1109/cvpr42600.2020.01054.

[42] S. Shi et al., "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," *International Journal of Computer Vision*, vol. 131, no. 2, pp. 531–551, 2022, doi: 10.1007/s11263-022-01710-9.

[43] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-Scale Voxel Feature Aggregation for 3D Object Detection from LIDAR Point Clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020, doi: 10.3390/s20030704.

[44] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21674–21683, 2023, doi: 10.1109/cvpr52729.2023.02076.

[45] T. Chen and C. Han, "Three-dimensional object detection with spatial-semantic features of point clouds," *Journal of Electronic Imaging*, vol. 32, no. 5, 2023, doi: 10.1117/1.jei.32.5.053039.

[46] J. Han, Z. Wan, Z. Liu, J. Feng, and B. Zhou, "SparseDet: Towards End-to-End 3D Object Detection," *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 781–792, 2022, doi: 10.5220/0010918000003124.

[47] D. Zhang, Z. Zheng, H. Niu, X. Wang, and X. Liu, "Fully Sparse Transformer 3-D Detector for LiDAR Point Cloud," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023, doi: 10.1109/tgrs.2023.3328929.

[48] L. Liu *et al*., "SparseDet: A Simple and Effective Framework for Fully Sparse LiDAR-Based 3-D Object Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–14, 2024, doi: 10.1109/tgrs.2024.3468394.

[49] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D Object Detection and Tracking," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11784-11793, 2021, doi: 10.1109/cvpr46437.2021.01161.

[50] Z. Song, H. Wei, C. Jia, Y. Xia, X. Li, and C. Zhang, "VP-Net: Voxels as Points for 3-D Object Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023, doi: 10.1109/tgrs.2023.3271020.

[51] L. Fan, F. Wang, N. Wang, and Z.-X. Zhang, "Fully sparse 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 351–363, 2022.

[52] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying Voxel based Representation with Transformer for 3D Object Detection," *arXiv preprint arXiv:2206.00630*, 2022.

[53] L. Guo, K. Lu, L. Huang, Y. Zhao, and Z. Liu, "Pillar-based multilayer pseudo-image 3D object detection," *Journal of Electronic Imaging*, vol. 33, no. 1, 2024, doi: 10.1117/1.jei.33.1.013024.

[54] J. Li, C. Luo, and X. Yang, "PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17567–17576, 2023, doi: 10.1109/cvpr52729.2023.01685.

[55] Li, Xusheng *et al*., "PillarNeXt: Improving the 3D detector by introducing Voxel2Pillar feature encoding and extracting multi-scale features." *arXiv preprint arXiv:2405.09828*, 2024.

[56] G. Shi, R. Li, and C. Ma, "PillarNet: Real-Time and High-Performance Pillar-Based 3D Object Detection," *Computer Vision – ECCV*, pp. 35–52, 2022, doi: 10.1007/978-3-031-20080-9_3.

[57] Y. Yang, J. Wang, X. Guo, X. Yang, and W. Qin, "Methods for Improving Point Cloud Authenticity in LiDAR Simulation for Autonomous Driving: A Review," in *IEEE Access*, vol. 13, pp. 4562-4580, 2025, doi: 10.1109/ACCESS.2025.3525805

[58] R. Ramana, V. Vasudevan, and B. S. Murugan, "Spectral Pyramid Pooling and Fused Keypoint Generation in ResNet-50 for Robust 3D Object Detection," *IETE Journal of Research*, pp. 1–13, 2025, doi: 10.1080/03772063.2025.2453897.

[59] F. Liu, Z. Lu, and X. Lin, "Vision-based environmental perception for autonomous driving," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 239, no. 1, pp. 39-69, 2025, doi: 10.1177/09544070231203059.

[60] R. Qiao, H. Ji, Z. Zhu, and W. Zhang, "Local-to-Global Semantic Learning for Multi-View 3D Object Detection From Point Cloud," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 9371-9385, Oct. 2024, doi: 10.1109/TCSVT.2024.3396870.

[61] L. Du *et al*., "AGO-Net: Association-Guided 3D Point Cloud Object Detection Network," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8097-8109, 1 Nov. 2022, doi: 10.1109/TPAMI.2021.3104172.

[62] H. Liu, T. Su, and J. Guo, "Autonomous driving enhanced: a fusion framework integrating LiDAR point clouds with monovision depth-aware transformers for robust object detection," *Engineering Research Express*, p. 015414, 20 Jan. 2025.

[63] N. Carion *et al*., "End-to-end object detection with transformers," *European conference on computer vision*, pp. 213-229, 2020.

[64] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[65] H. Zhao *et al*., "Point transformer." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

[66] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021, doi: 10.1007/s41095-021-0229-5.

[67] X. Zhu et al., "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[68] Y. Wang *et al*., "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries." *Conference on Robot Learning, PMLR*, pp. 180–191, 2022.

[69] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018, doi: 10.1109/cvpr.2018.00644.

[70] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High Quality Object Detection and Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021, doi: 10.1109/tpami.2019.2956516.

[71] A. Liu, L. Yuan, and J. Chen, "CSA-RCNN: Cascaded Self-Attention Networks for High-Quality 3-D Object Detection From LiDAR Point Clouds," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024, doi: 10.1109/tim.2024.3476690.

[72] Q. Cai *et al*., "3d cascade rcnn: High quality object detection in point clouds," *arXiv preprint arXiv:2211.08248*, 2022.

[73] H. Shenga *et al*., "Improving 3D Object Detection with Channel-wise Transformer," *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2723–2732, 2021, doi: 10.1109/iccv48922.2021.00274.

[74] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. VanGool, "Towards a weakly supervised framework for 3D point cloud object detection and annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp.1–16, 2021.

[75] B. Graham and L. van der Maaten, "Submanifold Sparse Convolutional Networks," *arXiv preprint arXiv:1706.01307*, 2017.

[76] S. Qiu, Y. Wu, S. Anwar, and C. Li, "Investigating Attention Mechanism in 3D Point Cloud Object Detection," *International Conference on 3D Vision (3DV)*, pp. 403–412, 2021, doi: 10.1109/3dv53792.2021.00050.

[77] J. Zhang, J. Wang, D. Xu, and Y. Li, "HcNet: a point cloud object detection network based on height and channel attention," *Remote Sensing*, vol. 13, no. 24, p. 5071, 2021.

[78] C. He, R. Li, S. Li, and L. Zhang, "Voxel Set Transformer: A Set-to-Set Approach to 3D Object Detection from Point Clouds," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8407–8417, 2022, doi: 10.1109/cvpr52688.2022.00823.

[79] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012, doi: 10.1109/cvpr.2012.6248074.

[80] M. Liu, J. Ma, Q. Zheng, Y. Liu, and G. Shi, "3D object detection based on attention and multi-scale feature fusion," *Sensors*, vol. 22, no. 10, p. 3935, May 2022, doi: 10.3390/s22103935.

[81] H. Caesar *et al*., "nuScenes: a multimodal dataset for autonomous driving," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621-11631, Jun. 2020, doi: 10.1109/cvpr42600.2020.01164.

[82] B. Zhu *et al*., "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[83] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3D object detection," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5418–5427, Jun. 2022, doi: 10.1109/cvpr52688.2022.00535.

[84] P. Hu, J. Ziglar, D. Held, and D. Ramanan, "What You See is What You Get: Exploiting Visibility for 3D Object Detection," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10998–11006, Jun. 2020, doi: 10.1109/cvpr42600.2020.01101.

[85] X. Bai *et al*., "TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1080–1089, Jun. 2022, doi: 10.1109/cvpr52688.2022.00116.

[86] O. D. Team. *Openpcdet: An open-source toolbox for 3d object detection from point clouds*. https://github.com/open-mmlab/OpenPCDet, 2020.