CA-Res3D: A Residual Framework with Channel Attention for 3D Object Detection

D. Jyothsna, *Member, IAENG,* G. Ramesh Chandra

***Abstract*—Accurate 3D object detection in point clouds remains critical for applications such as autonomous driving and robotics, where effective feature extraction is essential for robust performance. Traditional voxel-based networks often treat all feature channels equally, limiting the ability to focus on the most informative cues and leading to suboptimal detection accuracy in complex environments. This study presents a framework that embeds channel attention (CA) modules within a 3D ResNet backbone to improve feature representation and improve detection performance. The first contribution adopts a feature recalibration mechanism that adaptively reweights voxel feature channels by capturing global spatial context, effectively emphasizing salient features while suppressing irrelevant information. The second contribution is a stage-wise CA placement strategy that inserts recalibration blocks at multiple levels of the network, enabling progressive refinement of features from low-level details to high-level semantic abstractions. Comprehensive experiments demonstrate that the designed approach improves the mean average precision by 0.53% and 1.6% on KITTI and NuScenes datasets, respectively, for 3D object detection. The results indicate that the proposed model localizes the objects, which is critical for autonomous vehicles and robots.**

***Index Terms*—Point Clouds, Voxels, 3DResnet, Channel Attention, Object Detection, Feature Recalibration.**

I. Introduction

WITH the increasing deployment of autonomous vehicles and robotics, 3D object detection has become a critical task to understand the surrounding environment. This rapid advancement of autonomous systems and robotics has intensified the demand for accurate 3D object detection from point-cloud data. Deep learning frameworks are well-suited for real-world applications that demand accurate spatial understanding, such as the real-time interpretation of objects on images [1] and LiDAR-driven point clouds maneuvering around obstacles [2], capabilities that are critical for autonomous systems like robots and self-driving vehicles. Unlike 2D images, 3D Point Clouds obtained via LiDAR sensors, depth cameras [3] provide rich spatial perceptual data of the surrounding vicinity, enabling precise localization of unordered objects, and non-uniform, making direct application of traditional convolutional networks suboptimal. To address these challenges, structured representations, such as voxel grids and pillar columns, have been widely adopted to convert irregular point clouds into regular data formats suitable for convolutional neural networks. Despite significant progress, the ability to capture fine-grained details, particularly in cluttered or occluded environments, remains limited. In many real-world scenarios, accurate object localization is hindered by insufficient feature representation. This motivates the need for more intelligent mechanisms that can adaptively enhance informative features while suppressing irrelevant ones.

Existing point cloud approaches focus on point-centric, voxel-centric, and point-voxel-centric approaches. While point-centric approaches such as point-rcnn [4] and improved point-rcnn [5], implementing point-based and region-based convolutions, point GNN [6], a graph-based convolution approach, capture refined and fine-grained details and handle the unordered set of points efficiently, operating directly on the raw point clouds. However, these methods face challenges when large-scale point clouds are considered due to computational complexity. The other category includes voxel-centric approaches, where VoxelNet [7] and Second [8] have improved on this by first transferring the point clouds into discrete voxels that allow 3D convolutional operations on a structured grid. VoxelNet [7] was among the pioneers who adopted a voxel-based approach to handling sparse data using an end-to-end Voxelization strategy, which combines feature extraction along with bounding box prediction, followed by SECOND [8], which enhanced voxel processing with a sparse convolution network. PartA2[9], VoxelNeXt [10], Voxel-NeXt-Fusion [11], Trans-LGS [12], Focals-Conv [13], CenterPoint [14], CBGS [15], WYISYG [16], UVTR [17], Transfusion [18], Voxel-rcnn [19], and AGONet [20] are other recent voxel-based approaches. Pillar-based approaches are a variant of the voxel-centric approach, which are lightweight and the methods include PointPillars [21], SAE-Pillars [22], Pillar-Net [23], Pillar-NeXt [24], Pillar-Focus [25], PE-Pillar [26], and SP-Pillars [27]. These have surfaced as a cornerstone for efficient 3D object detection by converting sparse and unstructured point clouds into structured 2D representations. The other category is hybrid point-voxel approaches: these combine the best of each method - using voxelization for efficiency while helping to preserve fine details via direct processing of point clouds. PV-RCNN [28], PVRCNN++ [29], Pass PVRCNN++ [30] are point-voxel methods that are effective in detecting objects but are highly computational. Out of these approaches, voxel-driven approaches balance this trade-off by organizing the unstructured point clouds into structured format.

Despite these advantages, processing raw point clouds poses difficulties owing to non-uniform spatial sampling, irregular data distribution, and varying object scales. To process the irregularly distributed data, voxelization-based 3D convolutional neural networks (CNNs) [31]-[33] have emerged as a go-to architecture for extracting meaningful features from these sparse inputs, facilitating effective detection across diverse scenes. Pillar-based models, a variant of the voxel-centric approach, simplify 3D processing further

Manuscript received June 9, 2025; revised September 8, 2025.

D. Jyothsna is a PhD candidate (2201105002) of the Department of Computer Science and Engineering, JNT University Hyderabad, Telangana, India (e-mail: jyothsna_d@vnrvjiet.in).

G. Ramesh Chandra is a Professor of the Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India (corresponding author phone: +91 9885497583; e-mail: rameshchandra_g@vnrvjiet.in).

by collapsing the height dimension and treating vertical columns as 2D inputs, which allows the use of lightweight 2D convolutional backbones. However, this height compression is problematic for detecting tall or overlapping objects. Moreover, pillar encoding typically involves pooling point features within each pillar without distinguishing between informative and redundant features. In voxel-centric approaches, even though it simplifies processing, it suffers from uniform feature treatment across voxels without differentiating between the importance of various cues embedded in the voxel features. This indiscriminate processing often leads to noisy or less informative channels, negatively impacting the overall feature representation. The assumption that all voxel features contribute equally to the detection task limits the model's capacity to adapt to scene complexity. Moreover, with the varying object appearances, it requires adaptive mechanisms that can highlight the most relevant features while suppressing irrelevant or redundant information. Without such adaptability, feature representations may lack the discriminative power [7],[21] needed for accurate detection, particularly when dealing with partial observations or challenging backgrounds.

To address the limitation of discriminative power caused by uniform feature encoding in voxel-driven representations, recent methods have introduced Transformer attention mechanisms [12]. These enhancements aim to focus on the relevant features only across the spatial dimensions. Inspired by PC classification & segmentation [34] [35], the proposed method aims to adaptively emphasize relevant features and suppress less informative ones across spatial and channel dimensions. To this end, this study adopts a feature recalibration technique that dynamically adjusts the contribution of each feature channel based on its global importance. This helps the network focus on object-specific cues while ignoring background clutter. Additionally, feature recalibration ensures that meaningful spatial regions are amplified even if they are underrepresented in the raw input. By integrating such mechanisms into the backbone of voxel-based networks, it becomes possible to improve the network's sensitivity to critical object features, leading to enhanced detection performance under challenging conditions.

Therefore, this paper proposes an approach that integrates feature recalibration based on Channel Attention (CA) blocks directly within a 3D ResNet backbone. The proposed method adaptively adjusts the importance of voxel feature channels by capturing global spatial context, thereby amplifying salient and informative cues while suppressing irrelevant ones. Additionally, this work introduces a stage-wise CA placement strategy, inserting recalibration modules at multiple depths in the network to progressively refine features from low-level details to high-level semantics. This hierarchical recalibration enhances feature discriminability throughout the network. The main contributions are summarized as follows.

- This paper adopts a feature recalibration mechanism for voxelized point cloud data for selective feature channels by emphasizing relevant features.
- This work proposes a stage-wise CA placement strategy that enables progressive and hierarchical feature refinement, effectively leveraging the network's deep architecture to improve feature discrimination.
- To assess the proposed framework, this work demonstrates through extensive experiments, improvised over cutting-edge methods for 3D object detection in point clouds.

The following sections organize the remaining content of this work: Section II elaborates on the research methodology, including the techniques and procedures applied. Section III discusses the experimental findings, analyzing the implications and significance. Finally, Section IV summarizes the key conclusions and suggests avenues for further investigation.

II. Methodology

This section presents a two-phase voxel-driven 3D object detection system, including a 3D ResNet-based backbone, a region proposal network, a RoI head incorporating RoI pooling capabilities, and a detection head for the boundary box refinement. The proposed method distributes the points into voxel units and extracts the features over a 3D ResNet backbone. The 3D sparse voxels are transformed to BEV, and multiple 2D convolutional operations generate 3D region proposals, which are then processed by the Region of Interest (RoI) pooling layer to isolate relevant features. The extracted features were then forwarded to the detection head to accurately refine the predicted bounding boxes. Fig. 1 depicts the overall architecture of the proposed approach.

Problem Statement: Given a point cloud input $P = \{p_i \mid p_i \in \mathbb{R}^3,\ i = 1, 2, \ldots, N\}$, the goal of 3D object detection is to estimate a set of objects' bounding boxes $B = \{b_j \mid b_j \in \mathbb{R}^7,\ j = 1, 2, \ldots, M\}$ where each bounding box b_j is parameterized by location, size, and orientation in the 3D space. Typically, point clouds are voxelized into a discrete grid representation *X*, where the detection model f extracts feature *F* from the voxelized input and predicts the bounding boxes B'. The objective is to learn a feature extractor that maps the input voxels to highly discriminative features.

$$F = f(X) \tag{1}$$

$$B' = D(F) \tag{2}$$

where D(.) represents the detection head. A major challenge lies in adaptively enhancing relevant feature channels to improve detection accuracy, especially in complex environments. To address this issue, a channel attention with feature recalibration function is defined as eq3, which adaptively modulates the feature map by learning channel-wise factors.

$$R : \mathbb{R}^C \rightarrow \mathbb{R}^{C'} \tag{3}$$

A. Voxelization Process

The voxelization step transforms an unstructured 3D point cloud $P = \{\mathbf{p}_i = (x_i, y_i, z_i)\}_{i=1}^{N}$ into a discretized volumetric grid that enables efficient convolutional operations. The entire 3D region of interest is divided into equally spaced voxels with dimensions (v_x, v_y, v_z), resulting in a 3D voxel grid $G \in \mathbb{R}^{D \times H \times W}$. Each point p_i is mapped to its corresponding voxel via eq4.

$$(d_i, h_i, w_i) = \left(\left\lfloor \frac{z_i - z_{\min}}{v_z} \right\rfloor, \left\lfloor \frac{y_i - y_{\min}}{v_y} \right\rfloor, \left\lfloor \frac{x_i - x_{\min}}{v_x} \right\rfloor \right) \tag{4}$$

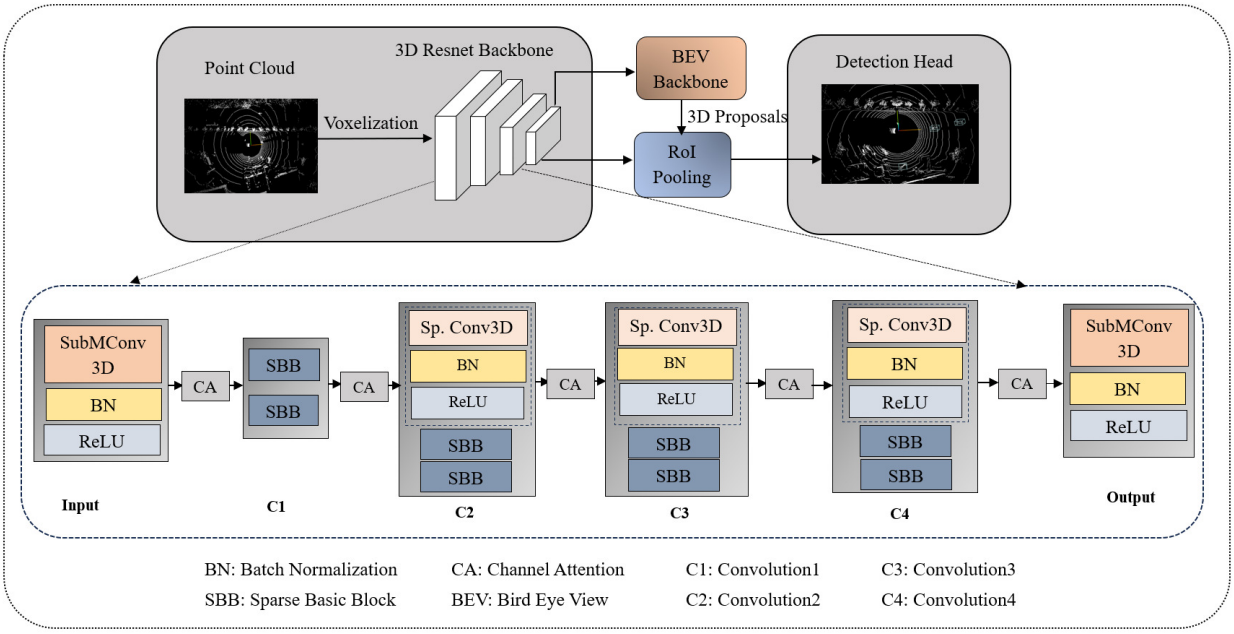

Fig. 1. Outline of the voxel-centric 3D object detection architecture. The proposed architecture begins by converting the raw input point cloud into structured voxel representations, enabling efficient 3D convolution. These voxel features are then processed through a 3D ResNet backbone with intermediate Channel Attention modules to enhance feature discrimination. The resulting high-level features are projected into Bird's Eye View (BEV) using a dedicated BEV backbone for spatial reasoning. Finally, 3D object proposals are generated and refined through a detection head for accurate object localization and detection.

This mapping allows grouping of spatially close points into common voxel bins. Within each voxel, features from the associated reflectance points are aggregated by mean feature encoding by learned transformations. The voxelized output, represented as $V \in \mathbb{R}^{C\times D\times H\times W}$ serves as a compact, spatially aligned input to 3D convolutional networks for downstream tasks like region proposal and object detection.

B. 3D Resnet Backbone

The proposed architecture utilizes a 3D ResNet backbone by incorporating a sparse 3D convolutional neural network (CNN) framework to efficiently generate feature representation from voxelized 3D data. The input to the 3D Resnet-based backbone is the voxelized grid dimensions obtained from equation 1. The input block initializes the feature extraction by transforming the input voxel features into a higher-dimensional space. Using SubMConv3d, it computes only non-empty voxels without changing the spatial resolution, which ensures that essential structural information is conserved. The conv1 block refines features through two SparseBasicBlock modules, each containing submanifold convolutions and residual connections. This block maintains spatial resolution and addresses the vanishing gradient problem. This step is vital for preserving fine-grained features at the original scale and is critical in detecting small objects. The conv2 stage introduces downsampling through SparseConv3d, which has a stride value of 2 and reduces the spatial dimension while increasing the feature dimension to 32. In this stage, the two sparse basic blocks refine the downsampled features. This block primarily handles the mid-level patterns through transitions from local to more abstract representations. In the conv3 block, further downsampling reduces the spatial resolution to one-fourth of the original and expands feature channels to 64. Sparse convolutions and basic blocks improve the high-level feature extraction, which tries to overcome the difficulty of maintaining semantic context at coarser resolutions. The final convolutional block, conv4, performs additional downsampling, which results in 128-dimensional features at one-eighth of the original resolution. The internal architecture of the backbone is provided in Table I, where SConv3D is the submanifold convolution and Sp.Conv3D is the sparse convolution. Conv_1, Conv_2, Conv_3, Conv_4 are the convolution layers in which SConv3D and Sp.Conv3D are implemented. The residual block, the Sparse Basic Block (SBB), contains two convolution layers with the Submanifold convolution operations performed alongside batch normalization and ReLU. This helps prevent vanishing gradients during training and accelerates convergence, which is shown in Fig. 2.

TABLE I
INTERNAL ARCHITECTURE DETAILS OF THE 3D RESNET BACKBONE
I-INPUT, O-OUTPUT, K-KERNEL, S-STRIDE, P-PADDING,
SCONV3D - SUBMANIFOLD3DCONV, SP. CONV3D -
SPARSECONVOLUTION3D

Conv.	Layers	I	O	K	S	P
Input	SConv3D	4	16	3,3,3	1,1,1	1,1,1
Conv_1 (C1)	SConv3D	16	16	3,3,3	1,1,1	1,1,1
Conv_2 (C2)	Sp. Conv3D	16	32	3,3,3	2,2,2	1,1,1
	SConv3D	32	32	3,3,3	1,1,1	1,1,1
Conv_3 (C3)	Sp. Conv3D	32	64	3,3,3	2,2,2	1,1,1
	SConv3D	64	64	3,3,3	1,1,1	1,1,1
Conv_4 (C4)	Sp. Conv3D	64	128	3,3,3	2,2,2	0,1,1
	SConv3D	128	128	3,3,3	1,1,1	1,1,1
Output	Sp. Conv3D	128	128	3,1,1	2,2,2	0,0,0

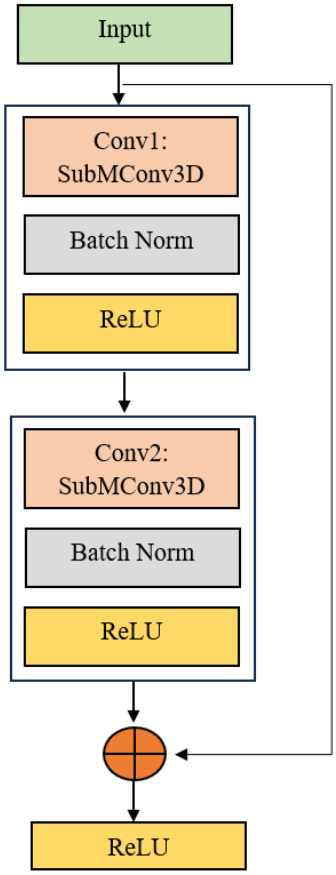

Fig. 2. Residual Block (Sparse Basic Block)

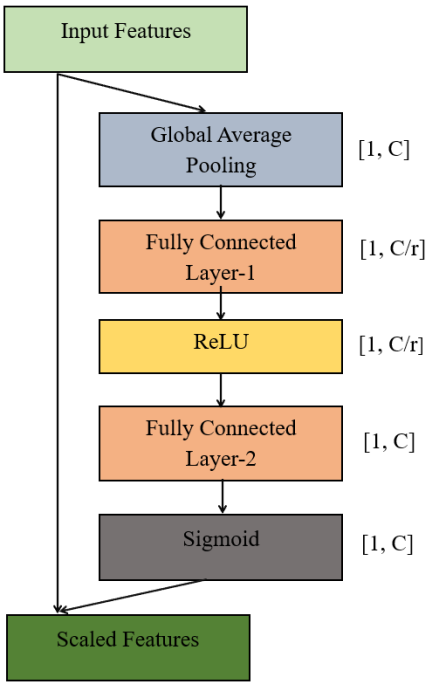

Fig. 3. Channel Attention

1) Channel Attention: Inspired by the Squeeze-and-Excitation mechanism [36], the proposed work designed channel attention through a feature recalibration module that explicitly models interdependencies between channels using global spatial context to improve the quality of voxel features by adaptively recalibrating the importance of each feature channel. The recalibration is critical because the spatial distribution of features across voxels contains rich cues that can guide the model to selectively enhance relevant feature channels and filter out redundant and irrelevant ones. The recalibration process begins by squeezing the spatial dimensions of the feature map $X \in \mathbb{R}^{D\times H\times W\times C}$ into a compact descriptor using global average pooling as eq5 for each channel *c=1,… C*. Next, the channel-wise descriptors pass across two sequential linear layers employing ReLU activation, with the final output modulated by a sigmoid function. This facilitates the model to learn inter-channel dependency and produces an attention score for each channel as depicted in eq6 where W1 and W2 denote the weights of learned fully connected layers, and σ indicates the sigmoid function. Finally, the feature maps are scaled as follows after recalibration - multiplication along the channel-wise dimension of the encoded features with attention scores rescales it as in eq7.

$$Z_c = \frac{1}{D \times H \times W} \sum_{d=1}^{D} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{c,d,h,w} \tag{5}$$

$$S_c = \sigma\left(W_2 \cdot ReLU(W_1 \cdot Z_c)\right) \tag{6}$$

$$x'_c = S_c \cdot X_c \tag{7}$$

This operation enhances the discriminative power of informative channels and suppresses less useful or noisy channels that might result from incomplete or ambiguous input observations. Fig. 3 shows the feature recalibration procedure for each channel. By integrating this module within the backbone, the network learns to focus on the most relevant features for robust 3D object detection.

2) 3D Resnet-Channel Attention Pipeline: While single-stage recalibration improves feature quality, feature representations in deep networks are hierarchical: early layers capture local spatial patterns and edges, while deeper layers encode more abstract semantic concepts. Thus, applying channel attention at only one stage limits the network's ability to refine features across different abstraction levels. To leverage this hierarchical nature, this study proposes a stage-wise SE placement strategy wherein recalibration modules are integrated after each major stage of the 3D ResNet backbone. Formally, let the backbone consist of L stages, each producing feature maps $\{X^{(1)}, X^{(2)}, \ldots, X^{(L+1)}\}$, where:

$$\{X^{(1)}, X^{(2)}, \ldots, X^{(L+1)}\} \tag{8}$$

This design allows the network to adaptively refine feature maps progressively. Early stages can focus on enhancing fine-grained local details such as edges and surfaces, while deeper stages emphasize semantic consistency. The multi-stage recalibration enables more nuanced and context-aware feature refinement throughout the entire network. Implementation outcomes exhibit that it consistently improves feature discriminability and detection performance over single-stage

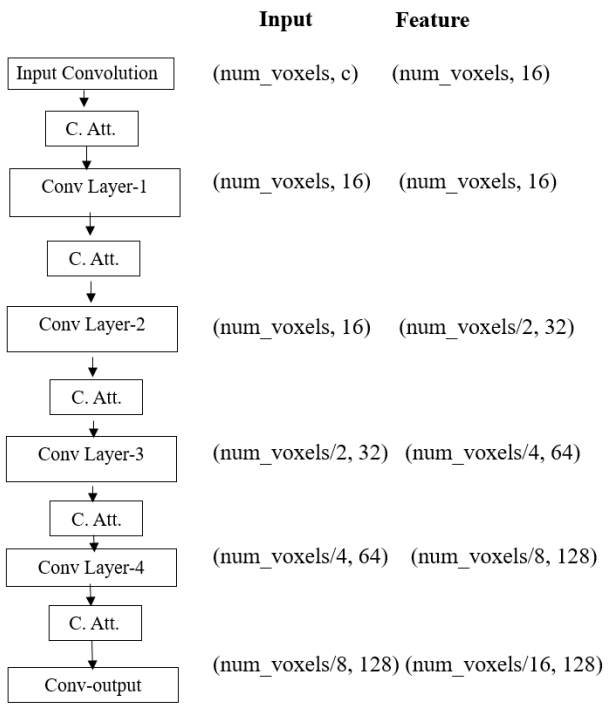

Fig. 4. Pipeline for Sparse 3D Resnet integrated with Channel Attention

recalibration or no recalibration. Fig. 4 depicts the pipeline for the integration of channel attention with ResNet blocks.

C. BEV & RPN

To simplify spatial reasoning in 3D point cloud data, the Bird's Eye View (BEV) projection is employed. The 3D voxel feature map $F_{BEV}(x, y) \in \mathbb{R}^{C \times D \times H \times W}$ is collapsed along the vertical (Z) axis to obtain a 2D BEV feature map $F_{BEV}(x, y)$ by summing features across all height slices.

$$F_{BEV}(x, y) = \sum_{z} F(x, y, z) \tag{9}$$

This results in a height-compressed representation, preserving spatial layout in the X-Y plane while aggregating semantic cues from all vertical layers, making it suitable for dense prediction tasks on ground planes. The Region Proposal Network (RPN) operates on this BEV feature map. It overlays a set of predefined anchors on the 2D BEV grid and evaluates each anchor for objectness and localization. For every anchor location, the RPN predicts p'- the objectness score (probability of containing an object), t'- the bounding box regression vector for refinement.

$$(p', t') = RPN(F_{BEV}) \tag{10}$$

D. Detection Head

The detection head refines the Region of Interest (RoI) features to improve the 3D box predictions. The initial transformation is a shared two-layer MLP transforming the RoI features into compact feature vectors. Then, these flattened representations are routed through dual processing streams- a regression branch for spatial localization and a classification branch for detection confidence. The box regression branch computes the residual adjustments from the region proposals

Fig. 5. Loss Curves

in 3D to the ground-truth bounding boxes, thus ensuring more precise localization. The entire network undergoes end-to-end training with a combined loss function, which serves as a classification and regression loss aimed at predicting the final bounding box parameters.

$$L = L_{cls}(v', v) + \lambda L_{reg}(b', b) \tag{11}$$

where *y'* and *b'* are predicted class scores and bounding boxes, *y* and *b* are ground-truth labels and boxes.

The training loss curve presented in Fig. 5 illustrates the convergence behavior of the proposed model for the classification, regression, and overall loss components. In the initial training stages, all three losses start at relatively high values, reflecting the model's unoptimized state. As the number of iterations increases, there is a steep decline in the loss values, indicating that the model is quickly learning meaningful representations. The regression loss consistently decreases at a faster rate compared to the classification loss, suggesting that the model is more efficient in minimizing localization errors. The total loss follows a smooth declining trend, combining the effects of both components, and eventually stabilizes, confirming that the model has reached a balanced state of learning.

III. Experimentation Analysis

A. Datasets and Implementation Details

KITTI Dataset: The experimental analysis was carried out using the KITTI [37] dataset with 7481 data instances. Here, the samples are further split into 3712 instances for training and 3769 instances for testing as a default split. Furthermore, custom split the training split of 3712 samples in a 5:1 ratio of training and validation sets. The training of the proposed model is done on a single NVIDIA GeForce RTX 4080 consisting of 16GB of GPU memory and 32GB of RAM over 100 epochs, employing a batch size of 2 and a learning rate of 0.01 with the ADAM optimizer. To evaluate the performance of the model, the Average Precision metric was computed. 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.

NuScenes Dataset: The experimentation was carried out using the NuScenes [38] dataset with 28130 samples, of which 22504 samples are split into the train set, 5626 samples are split into the validation split, and 6019 samples for testing. For the 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 the average precision metric, 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 Adam one-cycle optimizer with a learning rate of 0.01 and a weight decay of 0.01.

B. Evaluation of the Proposed Method

Results on the KITTI Dataset: The proposed model is assessed on the KITTI dataset under three categories-Easy, Moderate, and Hard. Several experiments were carried out on the validation set, and finally tested the model. Table II demonstrates the object detection performance with average precision for the mentioned three objects at 40 recall points. The IoU threshold is 0.7 for Car and 0.5 for Pedestrian and Cyclist. From the results obtained, the proposed model has improved the average precision with 82.3% for Car and 44.9% for Pedestrian on the Moderate difficulty, i.e., an improvement of 1.02% and 1.9% on the most recent methods, Pass-PVRCNN++ [30] and VoxelNeXt [10] methods on the "Car" object. 2.95% and 2.2% improvement for the pedestrian object compared to the same methods. In comparison with the state-of-the-art approaches, overall mAP is improved to 0.53% by achieving the improved precision of 65.45%. Table III shows the improvement of results on the validation split with the average precision metric for the mentioned three objects.

To explore the visualizations on the KITTI dataset, the feature maps from various channels are depicted for both the 3D Resnet backbone and the 3D Resnet backbone with CA, as shown in Fig. 6 (a) and (b). In comparison with the above two variants of backbone, there is a notable difference in channel 6, marked green for the object activation with the SE integrated backbone. However, sparse activations in a few channels suggest that the features they capture are not crucial for the current input but could become relevant for other inputs with different spatial arrangements and object configurations. Finally, by combining low-level features extracted from earlier layers with higher-level abstractions from deeper layers, the model constructs a comprehensive understanding of the scene, enabling it to form a spatial hierarchy that integrates information across all levels and supports robust decision-making. Also, the individual feature maps are depicted in Fig. 7.

Fig. 8, 9, and 10 depict the qualitative detections of the proposed model in the scenes under sparse regions, dense regions, and comparative analysis with the VoxelNeXt method. Fig. 8 (a) and (b) represent the predictions of Car and Pedestrian objects in less crowded areas or sparse regions. Fig. 9 (a) and (b) show the detections in crowded and dense regions with objects Car, Pedestrian, and Cyclist, which has high occurrence and overlapping of objects. From this, it is evident that the proposed model is capable of detecting objects under occluded conditions also. Fig. 10 shows the comparison of detections of the proposed model with the VoxelNeXt model, where it is evident that a few missed detections of VoxelNeXt are detected by the proposed method, which signifies the improvement of the proposed method. The representations for Car, Pedestrian, and Cyclist are in green, blue, and yellow bounding boxes, respectively.

Results on the NuScenes Dataset: The proposed 3D Sparse ResNet with channel attention framework has been evaluated on the NuScenes dataset, which is a large-scale autonomous driving dataset. All the results are based on LIDAR-only inputs without leveraging external data. The objects used in evaluating on NuScenes dataset for the mean average precision metric are Car, Bus, Construction Vehicle, Trailer, Truck, Bicycle, Barrier, Motor Cycle, Pedestrian, and Traffic Cone. Training and testing are performed using center-based detection head aligned with [10]. Table IV discusses the detailed results on the NuScenes test split. With the mentioned ten objects, the proposed model achieved mAP of 66.1%, which is improved by 1.6% compared to the most recent method, VoxelNeXt. For the objects Car and Mot., it has improved the average precision results have improved on the proposed model. However, there is a significant improvement for the Car Class with 2.9%. The underlined scores represent the second-highest average precision values. Table V shows the improvement of average precision on the validation split. The highlighted bold ones represent the improved detections in terms of average precision.

C. Ablation Studies

To verify the effectiveness of the feature recalibration module, 3D ResNet has been taken into consideration. Table VI describes the improvement of the average precision of all three classes, Car, Pede, and Cyclist, on the Moderate Category, which implies that the slightly overlapped objects are present in the scene. Notably, the pedestrian detection accuracy improves by 5.0%, indicating the CA module's ability to amplify subtle and discriminative features in sparse and challenging input regions. Similarly, the car and cyclist categories show respective improvements of 3.1% and 3.9%, reflecting better spatial focus and inter-channel sensitivity in the feature learning process.

To validate the effectiveness of stage-wise placement of the channel attention model, layered positioning of channel attention plays a key role. Specifically, the proposed study evaluated three configurations: (i) inserting CA blocks only after the initial convolutional layer (early-stage), (ii) inserting CA blocks after every convolutional block (full-stage). Results on the validation set reveal that full-stage integration improves performance gain, improving mAP to 84.6%. Early-stage CA provides modest gains, likely due to its limited ability to capture high-level semantic dependencies. These findings suggest that distributed CA placement across the network enables better feature recalibration, contributing to both spatial precision and precise localization of objects under dense or occluded regions. Tables VII and VIII show the variants of positioning channel attention blocks. Fig. 11 depicts the 3D mAP of Car, Ped, and Cycl classes with the CA module.

TABLE II
3D OBJECT DETECTION PERFORMANCE: AVERAGE PRECISION METRIC ON KITTI TEST SPLIT AT 40 RECALL POINTS. DIFFICULTY LEVELS: E-EASY, M-MODERATE, AND H-HARD.

Model Name	**YoP**	**Car (R40) %**			**Pedestrian (R40) %**			**Cyclist (R40) %**			**mAP**
		E	*M*	*H*	*E*	*M*	*H*	*E*	*M*	*H*	
Voxel-Net [7]	2018	77.50	65.10	57.70	39.50	33.70	31.50	61.20	48.40	44.40	51.00
SECOND [8]	2018	83.10	73.70	66.20	51.10	42.60	37.30	70.50	53.90	46.90	58.37
Point-RCNN [4]	2019	85.90	75.80	68.30	49.40	41.80	38.60	73.90	59.60	53.60	60.77
PointPillar [21]	2019	79.10	75.00	68.30	52.10	43.50	41.50	75.80	59.10	52.90	60.81
PointGNN [6]	2020	88.30	79.40	72.20	51.90	43.70	40.10	78.60	63.40	57.10	63.86
Part-A2-Free [9]	2020	85.90	77.90	72.00	54.50	44.50	42.40	78.60	62.70	57.70	64.02
Voxel-RCNN [19]	2021	90.70	81.60	77.40	52.50	44.80	39.00	77.50	64.00	53.10	64.51
PV-RCNN [28]	2020	90.20	81.40	76.80	52.10	43.20	40.20	78.60	63.70	57.60	64.87
PVRCNN++ [29]	2022	87.72	81.29	76.78	47.50	40.31	38.15	80.34	67.46	60.48	64.45
Pillar-Trans-LGS [12]	2023	88.07	78.96	74.05	49.52	42.75	39.62	78.47	66.24	58.91	64.07
VoxelneXt [10]	2024	89.10	80.40	76.90	52.10	42.70	39.10	81.30	65.30	57.40	64.92
Pass-PVRCNN++ [30]	2025	87.65	81.28	76.79	47.66	41.95	38.90	80.43	68.45	60.93	64.89
CA-Resnet (Ours)	2025	88.80	**82.30**	77.02	51.00	**44.90**	39.60	**81.40**	66.10	57.90	**65.45**

TABLE III
3D OBJECT DETECTION PERFORMANCE: AVERAGE PRECISION METRIC ON KITTI VAL. SPLIT AT 40 RECALL POINTS. DIFFICULTY LEVELS: E-EASY, M-MODERATE, AND H-HARD

Model Name	**Car (R40) %**			**Pedestrian (R40) %**			**Cyclist (R40) %**		
	E	*M*	*H*	*E*	*M*	*H*	*E*	*M*	*H*
VoxelNet [7]	84.10	72.54	68.38	56.75	51.16	46.81	78.16	59.79	51.03
SECOND [8]	85.25	75.66	71.64	54.72	50.65	45.93	77.72	61.92	58.69
PointRCNN [4]	86.18	75.94	73.27	57.53	50.20	46.42	83.15	64.04	59.50
PointPillars [21]	87.74	78.38	75.52	53.92	48.43	43.58	81.10	62.36	58.24
Part-A^2 [9]	87.81	78.89	73.51	58.10	52.55	48.06	79.97	64.82	57.93
PointGNN [6]	88.33	79.47	72.29	51.92	43.77	40.14	78.60	63.48	57.08
SP-Pillars [27]	88.93	79.93	76.16	58.16	52.94	48.32	86.28	66.78	62.65
CA-Resnet (ours)	**91.58**	**82.45**	**79.69**	**65.76**	**57.82**	**52.55**	**83.29**	**69.76**	**63.67**

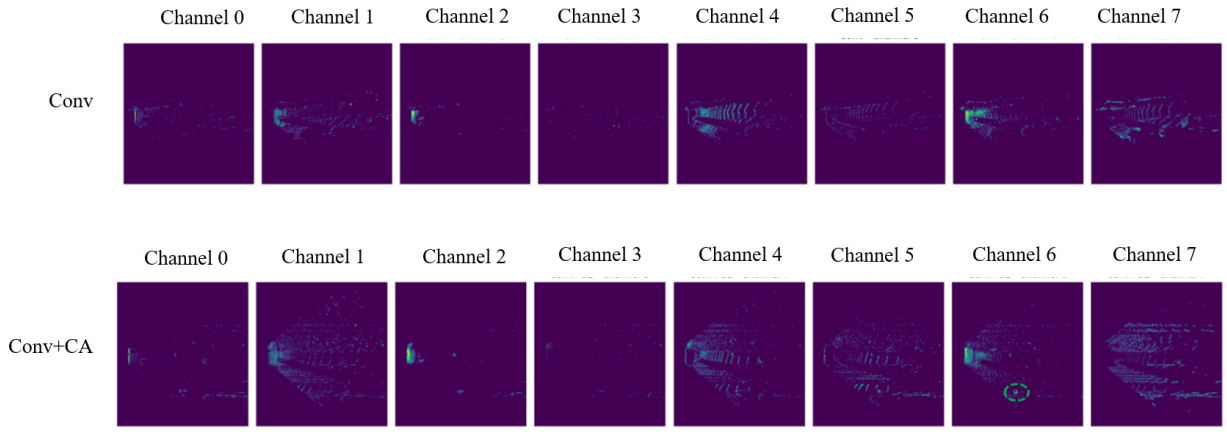

Fig. 6. (a)Feature Maps for 3D Resnet (b)Feature Maps for 3D Resnet+CA

Table IX shows the importance of stage-wise positioning of the CA blocks contributing to the mean average precision. This describes the mAP of Car, Pedestrain(pede.) and Cyclist(Cycl) classes of each difficulty level, Easy, Moderate, and Hard. A progressive ablation study evaluating the impact of stage-wise integration of CA modules into a 3D ResNet (3DR) architecture across the input layer, along with four convolutional layers(L1–L5). Beginning with the baseline 3DR, a steady improvement in detection accuracy is observed as SE modules are incrementally introduced at deeper layers. Specifically, initial inclusion at L1 yields modest gains, while cumulative placements up to L5 result in significant performance boosts, notably improving the moderate difficulty category from 65.33% to 69.05%. Full deployment of SE across all layers (L1–L5 and final output) achieves the improved scores, with 80.21% on easy, 70.01%

TABLE IV
COMPARISON OF MAP METRIC ON NUSCENES TEST SPLIT FOR 3D OBJECT DETECTION. THE '-' INDICATES THAT DATA IS NOT PUBLICLY AVAILABLE.

Method	Car	T.C.	Ped.	Mot.	C.V	Trailer	Bar.	Byc.	Bus	Truck	mAP
Point Pillar [21]	68.4	30.8	59.7	27.4	4.1	23.4	38.9	1.1	28.2	23.0	30.5
itKD [32]	79.4	45.90	73.91	30.21	3.5	26.4	53.7	5.39	54.3	40.3	41.3
CBGS [15]	81.1	70.9	80.1	51.5	10.5	42.9	65.7	22.3	54.9	48.5	52.8
CenterPoint [14]	84.6	76.7	83.4	53.7	17.5	53.2	70.9	28.7	60.2	51.0	58.0
VPSNet [33]	84.8	–	–	–	22.9	38.9	68.2	–	72.5	58.6	61.7
Focals-Conv [13]	86.7	81.4	**87.5**	64.5	23.8	59.5	74.1	36.3	**67.7**	56.3	63.8
PillarNet [23]	87.4	**82.1**	87.2	67.4	**30.4**	**61.8**	**76.0**	40.3	60.9	**56.7**	65.0
VoxelNeXt [10]	84.6	79.0	85.8	73.2	28.7	55.8	74.6	**45.7**	64.7	53.0	64.5
Ours	**87.5**	81.5	87.3	**74.2**	28.9	61.7	75.1	43.4	65.1	56.3	**66.1**

TABLE V
COMPARISON OF AVERAGE PRECISION METRIC ON NUSCENES VALIDATION SPLIT FOR 3D OBJECT DETECTION

Method	Car	Ped.	C.V	Bar.	T.C.	Truck	Bus	Byc.	Trailer	Mot.
SECOND [8]	81.8	77.7	15.0	59.2	57.4	51.7	66.9	17.5	37.3	42.5
CenterPoint [14]	85.0	85.3	15.5	67.1	70.0	58.2	69.5	40.9	35.7	58.8
Transfusion [18]	86.9	87.5	25.2	70.3	77.2	60.8	73.1	**57.3**	43.4	72.9
WYSIWYG [16]	80.0	66.9	7.5	34.5	27.9	35.8	54.1	0.0	28.5	18.5
PillarNeXt [24]	84.8	86.1	21.8	68.2	74.2	58.0	68.3	56.5	37.1	68.4
AGONet [20]	81.5	72.2	13.3	51.2	48.1	50.1	62.2	5.9	34.0	32.5
VoxelNeXt [10]	85.6	85.4	17.9	68.1	70.0	58.4	71.6	43.4	38.6	59.7
Ours	**88.1**	**89.1**	**30.1**	**78.1**	**83.9**	57.2	66.0	44.5	**63.0**	**74.6**

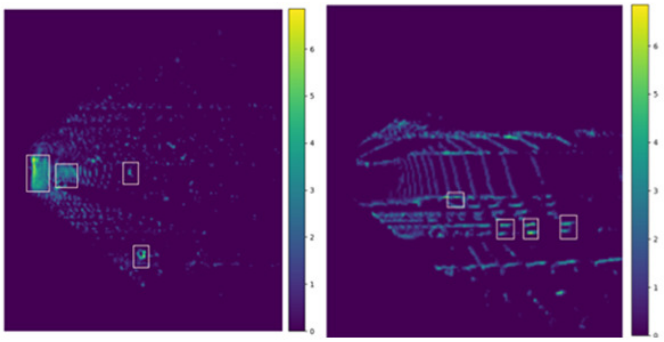

Fig. 7. Individual Feature Maps

TABLE VI
AVERAGE PRECISION RESULTS ON THE MODERATE CATEGORY IMPROVEMENTS OVER THE 3D RESNET MODEL

Model Name	Car% (R40)	Pede. % (R40)	Cycl. % (R40)
3D Resnet	79.2	39.9	62.2
3D Resnet + CA	82.3	44.9	66.1
Improvement	(+)3.1	(+)5	(+)3.9

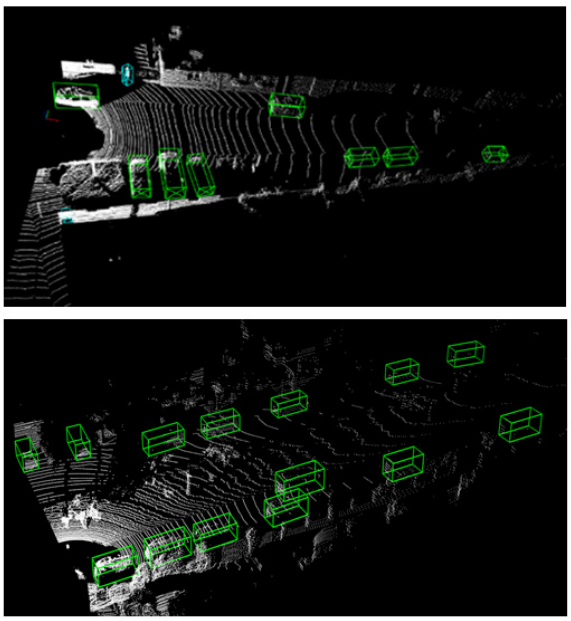

Fig. 8. Visualization of Qualitative Predictions on the KITTI dataset in the sparse and less crowded regions.

on moderate, and 65.37% on hard difficulty levels. These results substantiate the effectiveness of the proposed stage-wise CA placement strategy in progressively refining deep feature representations for 3D object detection.

Fig. 12 showcases a comparison of mean Average Precision (mAP) for three classes—Car, Pedestrian (Ped), and Cyclist (Cyc)—measured at recall thresholds R11 and R40. The bar graph illustrates that while both R11 and R40 yield high mAP values for cars and cyclists, a noticeable improvement is observed in the cyclist category during the transition from R11 to R40. Conversely, pedestrian detection shows only a marginal difference between the two settings. A line graph representing the relative gain highlights that the cyclist category experiences the most significant performance boost, while pedestrian detection shows minimal gain. This indicates that increasing the recall value from R11 to R40 particularly benefits categories with more spatial variability, such as cyclists. Collectively, the results emphasize the effectiveness of denser proposal sampling with improved

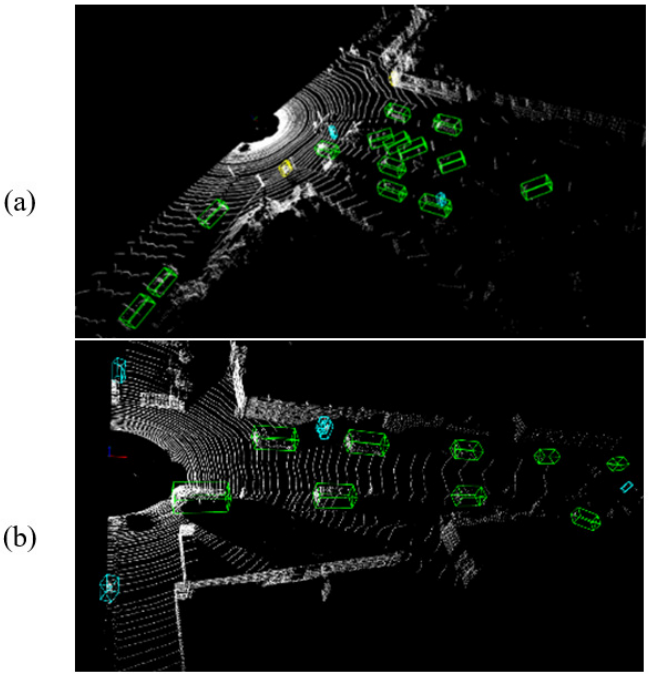

Fig. 9. Visualization of Qualitative Predictions on the KITTI dataset in dense regions.

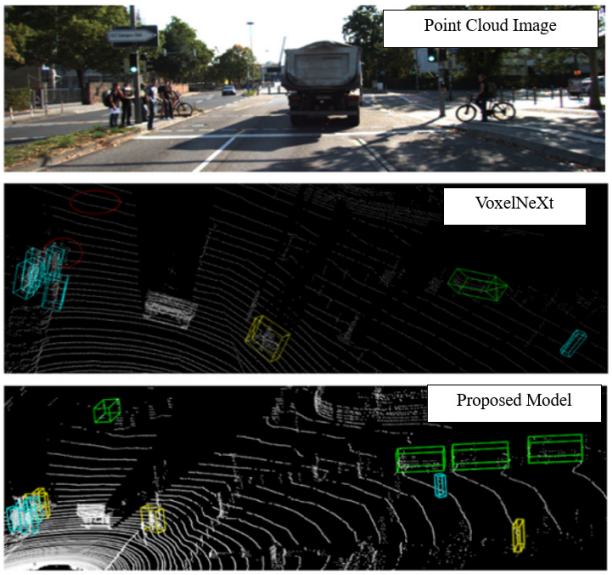

Fig. 10. Comparison of predictions between VoxelneXt and the proposed model in dense regions.

TABLE VII
ABLATION EXPERIMENTS ON THE MODEL VARIANTS WITH THE POSITIONING OF THE CHANNEL ATTENTION FOR EASY AND MODERATE DIFFICULTY

Model Variants	Easy Diff.			Mod. Diff.		
	Car	Ped.	Cycl.	Car	Ped.	Cycl.
3D Resnet	89.67	54.42	79.78	78.65	48.16	69.18
3DR+CA	90.02	55.71	81.44	79.23	50.29	69.21
3DR+CA& Conv	91.58	65.76	83.29	82.45	57.82	69.76

performance in detection, especially for complex object classes.

TABLE VIII
ABLATION EXPERIMENTS ON THE MODEL VARIANTS WITH THE POSITIONING OF THE CHANNEL ATTENTION FOR HARD DIFFICULTY

Model Variants	Hard Diff.		
	Car	Ped.	Cycl.
3D Resnet	75.92	45.09	65.09
3D Resnet + CA Input	76.50	47.31	63.55
3DR + CA Input & Conv	79.90	52.55	63.67

TABLE IX
ABLATION EXPERIMENTS ON MAP OF THE MODEL VARIANTS WITH THE LAYERED POSITIONING OF THE CHANNEL ATTENTION. L1 . . . L5 REPRESENTS THE LAYERED CA MODULE AT EACH POSITION

3DR	L1	L2	L3	L4	L5	Easy Diff. (%)	Mod. Diff. (%)	Hard Diff. (%)
✓						74.62	65.33	62.03
✓	✓					75.72	66.24	62.45
✓	✓	✓				75.80	66.58	62.39
✓	✓	✓				76.06	67.30	63.02
✓	✓	✓	✓	✓		78.76	69.05	63.93
✓	✓	✓	✓	✓	✓	80.21	70.01	65.37

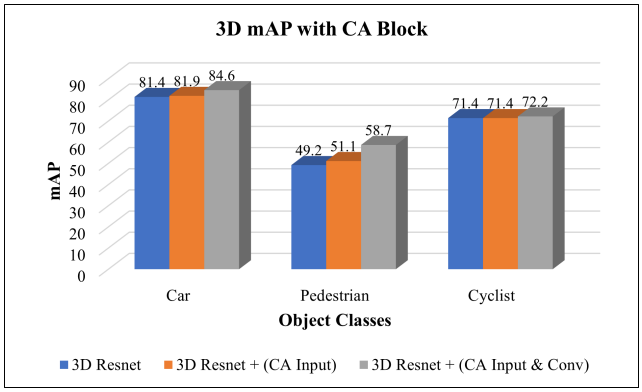

Fig. 11. 3D mAP comparison with CA-block at different stages on the validation set

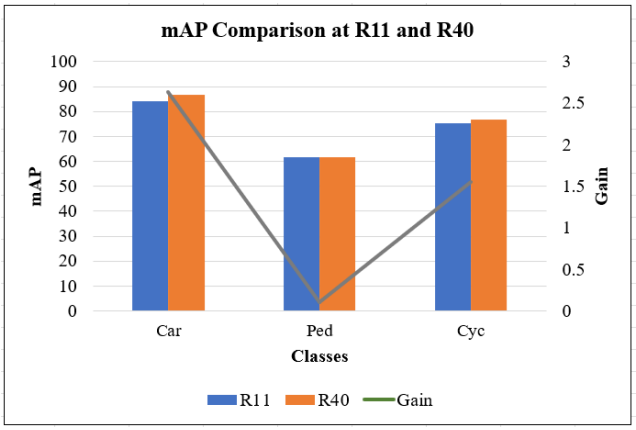

Fig. 12. 3D mAP comparison at different recall values

The evaluation results shown in tables X and XI across three object categories—Car, Pedestrian, and Cyclist—are summarized under three difficulty levels: Easy (E), Moderate (M), and Hard (H), using 2D detection, Bird's Eye View (BEV), and Average Orientation Similarity (AOS) metrics. For the Car category, performance remains consistently high,

TABLE X
MAP RESULTS OF 2D, BEV, AND AOS ON THE CAR CLASS FOR CATEGORIES – EASY, MODERATE, AND HARD

Task	E	M	H
2D	96.55	93.36	90.71
BEV	93.10	89.40	86.65
AOS	96.53	93.23	90.48

TABLE XI
MAP RESULTS OF 2D, BEV, AND AOS ON THE PEDESTRIAN AND CYCLIST CLASSES FOR CATEGORIES – EASY, MODERATE, AND HARD

Task	Pedestrian			Cyclist		
	E	M	H	E	M	H
2D	70.5	59.3	55.3	87.6	78.8	72.1
BEV	54.1	46.3	43.0	82.9	70.1	63.6
AOS	63.3	52.2	48.5	87.2	77.7	71.0

with 2D detection achieving 96.55% for Easy, 93.36% for Moderate, and 90.71% for Hard. BEV and AOS scores follow closely, indicating reliable spatial and orientation estimates even under challenging conditions. Pedestrian detection shows a notable drop in all metrics as difficulty increases, with 2D detection ranging from 70.5% to 55.3%, and BEV and AOS reflecting similar declines. In contrast, Cyclist detection remains strong, particularly in 2D and AOS, where easy-level scores exceed 87%, though there is a gradual decrease in performance under harder conditions. These trends highlight that while the model is highly robust for vehicles and cyclists, pedestrian detection remains more sensitive to increased complexity and occlusion in the scenes. These results collectively demonstrate that high accuracy in 2D, BEV, and AOS directly supports more reliable and comprehensive 3D object detection. Strong performance across views and orientation metrics indicates the model's ability to localize the objects accurately in three-dimensional space, even under challenging conditions.

The Precision-Recall (PR) curves depicted in Fig. 13, Fig. 14, and Fig. 15 for Car, Pedestrian, and Cyclist classes reflect the model's strong detection capabilities across varying difficulty levels. For the Car category, the curves are tightly grouped and remain consistently high, indicating reliable detection even in complex scenarios. Pedestrian detection shows a smooth progression across Easy, Moderate, and Hard settings, demonstrating the model's ability to adapt to changes in object visibility and scene density. Cyclist detection maintains strong precision values across a wide recall range, particularly under simpler conditions, and retains balanced performance as complexity increases. The clear separation of curves across difficulty levels provides valuable insight into how the model responds to different environmental and occlusion challenges. Overall, the PR curves reinforce the effectiveness of the detection framework, highlighting its robustness, scalability, and generalization across object types and its deployment potential for robust 3D perception systems.

Table XII depicts the hyperparameters of the model which

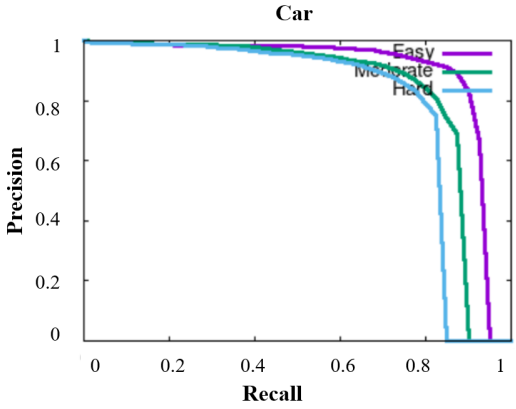

Fig. 13. Precision-Recall Curve for Car Class

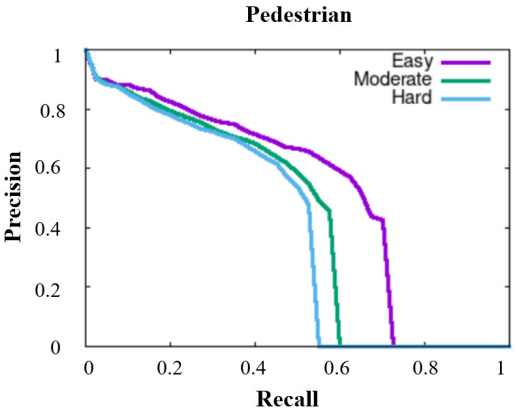

Fig. 14. Precision-Recall Curve for Pedestrian Class

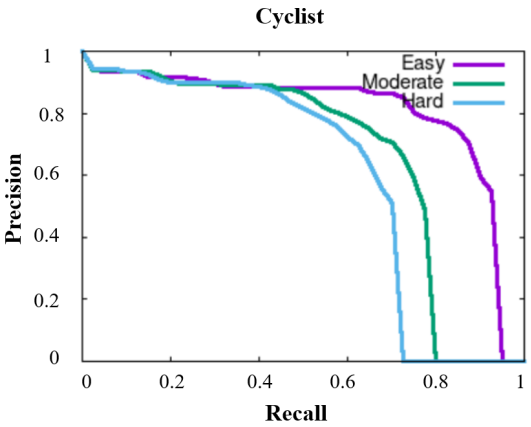

Fig. 15. Precision-Recall Curve of Cyclist Class

are emprirically chosen based on the default parameters of the OpenPCDet's standard. Table XIII presents a comparative summary of the parameter efficiency and accuracy achieved by different 3D object detection models. Part-A2, with 59.23M parameters, attains an accuracy of 77.9%, whereas PV-RCNN, despite having considerably fewer parameters (13.12M), delivers a higher accuracy of 81.4%. SparseDet balances parameter count and performance with 25.10M parameters and 78.45% accuracy. In contrast, the proposed model demonstrates a clear advantage by requiring only 11.17M parameters while achieving the highest accuracy of 82.30%. This highlights the model's ability to achieve

TABLE XII
HYPERPARAMETERS CONFIGURATION OF THE MODEL

HyperParameter	Value
Batch Size	2
Optimizer	Adam
Learning Rate	0.01
Weight Decay	0.01
Momentum	0.9
No. of Epochs	100

TABLE XIII
PARAMETER EFFICIENCY AND PERFORMANCE ACROSS DIFFERENT MODELS

Model Name	No. of Parameters (in Million)	Car-Mod.(%)
Part-A2[9]	59.23	77.9
PV-RCNN[28]	13.12	81.4
SparseDet[35]	25.10	78.45
Ours	**11.17**	**82.30**

superior detection performance with significantly reduced computational complexity, making it more practical for deployment in resource-constrained environments.

The key findings of this study emphasize the effectiveness of integrating channel-focused attention mechanisms into 3D convolutional architectures for point cloud-based object detection. By systematically inserting channel attention modules at varying convolution layers of the 3D ResNet backbone, the model demonstrates a consistent improvement in class-wise detection performance. Notably, introducing CA modules enhances the model's ability to prioritize informative features while suppressing less relevant activations, leading to measurable gains, especially for the medium difficulty category of the objects. These outcomes suggest that attention-based recalibration, when applied in a structured, stage-wise manner, substantially strengthens the model's feature discrimination (from Fig. 6 and Table VIII) and improves the detection capability across different object categories.

IV. CONCLUSION

This work deliberates a two-stage voxel-based approach with sparse 3D Resnet combined with Channel Attention for 3D object detection, leveraging two key enhancements. Firstly, channel attention mechanism was adopted for feature recalibration, and secondly, systematic integration of layer-wise CA across 3D Resnet for deeper refinement of selective features. The experimentation results showed an improvement in the detection performance of the moderate category of the three objects: Car to 82.3%, Pedestrian to 44.9%, and Cyclist to 66.1% in LiDAR-only modality on the KITTI dataset. Highlighting the potential of voxel representation in incorporating the Channel attention mechanism, enhancing the overall mean average precision of the model on the KITTI dataset to 65.45% and on the NuScenes dataset to 66.1% within the sparse networks with improved feature discrimination.

The proposed approach fills the performance gap between traditional 3D object detection methods and sparse point cloud detection, pertaining to accurate and precise detection and recall of objects. However, even though the current approach achieved better improvements, there is a necessity to improve the precision. In this direction, future works focus on extending this model to improve the precision by developing the region proposal network, which is of more importance in detecting objects in dynamic and cluttered environments, along with the sparsity of the data.

ACKNOWLEDGMENT

The authors wish to extend gratitude to the institute management towards the continuous encouragement and for providing the required computational resources essential for progress of the research.

REFERENCES

[1] O. Bouazizi, C. Azroumahli, A. E. Mourabit, and M. Oussouaddi, "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.

[2] 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.

[3] 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.

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

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

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

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

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

[9] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From Points to Parts:3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020, doi: 10.1109/tpami.2020.2977026.

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

[11] Ziying Song, Guoxin Zhang, Jun Xie, *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.

[12] M. Uzair, J. Dong, R. Shi, H. Mushtaq, and I. Ullah, "Trans-LGS: Transformer-Based Local Graph Structure for 3D Object Detection in Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, pp. 1–18, 2025, doi: 10.1109/tvt.2024.3506580.

[13] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal Sparse Convolutional Networks for 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5418–5427, 2022, doi: 10.1109/cvpr52688.2022.00535.

[14] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3D Object Detection and Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, doi: 10.1109/cvpr46437.2021.01161.

[15] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," *CoRR*, abs/1908.09492, 2019.

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

[17] 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. [Online]. Available: https://arxiv.org/abs/2206.00630

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

[19] 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.

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

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

[22] Y. Zhang and Z. Zhou, "SAE-PointPillars: Adaptive Spatial Feature Fusion Based PointPillars 3D Target Detection Algorithm," *IAENG International Journal of Computer Science*, vol. 52, no. 6, pp. 1627–1636, 2025.

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

[24] X. Li, C. Wang, S. Wang, Z. Zeng, and J. Liu, "PillarNeXt: Improving the 3D Detector by Introducing Voxel2Pillar Feature Encoding and Extracting Multi-Scale Features," *arXiv preprint arXiv:2405.09828*, 2024. [Online]. Available: https://arxiv.org/abs/2405.09828

[25] Y. Gao, Peng Wang, Xiaoyan Li, *et al.*, "PillarFocusNet for 3D Object Detection with Perceptual Diffusion and Key Feature Understanding," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-92338-5.

[26] L. Sun, Y. Li, and W. Qin, "PEPillar: A Point-Enhanced Pillar Network for Efficient 3D Object Detection in Autonomous Driving," *The Visual Computer*, 2024, doi: 10.1007/s00371-024-03481-5.

[27] T. Chen, Y. Yuan, B. Yin, and Y. Liao, "SP-Pillars: An Efficient LiDAR 3D Objects Detection Framework with Multi-Scale Feature Perception and Optimization," *IEEE Access*, p. 1, 2025, doi: 10.1109/access.2025.3564665.

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

[29] S. Shi,Li Jiang, Jiajun Deng, *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.

[30] S. Chen, H. Zhang, and N. Zheng, "Leveraging Anchor-Based LiDAR 3D Object Detection via Point Assisted Sample Selection," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2025, doi: 10.1109/tits.2025.3555229.

[31] A. Pravallika, M. F. Hashmi, and A. Gupta, "Deep Learning Frontiers in 3D Object Detection: A Comprehensive Review for Autonomous Driving," *IEEE Access*, vol. 12, pp. 173936–173980, 2024, doi: 10.1109/ACCESS.2024.3456893.

[32] H. Cho, J. Choi, G. Baek, and W. Hwang, "ITKD: Interchange transfer-based knowledge distillation for 3D object detection," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 33, pp. 13540–13549, 2023, doi: 10.1109/cvpr52729.2023.01301.

[33] J. Wen, Q. Zhang, and G. Zhang, "VPSNet: 3D object detection with voxel purification and fully sparse convolutional networks," *The Journal of Supercomputing*, vol. 81, no. 3, 2025, doi: 10.1007/s11227-024-06890-4.

[34] B. Ma, "Point Cloud Classification and Segmentation Using Channel-Aware Dynamic Convolutional Neural Network," *Engineering Letters*, vol. 30, no. 2, pp. 711–717, 2022.

[35] 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.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018, doi: 10.1109/CVPR.2018.00745.

[37] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012, doi: 10.1109/cvpr.2012.6248074.

[38] Holger Caesar, Varun Bankiti, Alex H. Lang, *et al.*, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, doi: 10.1109/cvpr42600.2020.01164.

G. Ramesh Chandra is a Professor in the Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology affiliated to JNT University Hyderabad, Telangana, India. His expertise encompasses 3D image processing, point cloud processing, data mining and computer vision.