World Electric Vehicle Journal

MDPI

Article

A Self-Attention-Enhanced 3D Object Detection Algorithm Based on a Voxel Backbone Network

Zhiyong Wang and Xiaoci Huang *

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m310123423@sues.edu.cn
* Correspondence: 06060005@sues.edu.cn

Abstract

3D object detection is a fundamental task in autonomous driving. In recent years, voxel-based methods have demonstrated significant advantages in reducing computational complexity and memory consumption when processing large-scale point cloud data. A representative method, Voxel-RCNN, introduces Region of Interest (RoI) pooling on voxel features, successfully bridging the gap between voxel and point cloud representations for enhanced 3D object detection. However, its robustness deteriorates when detecting distant objects or in the presence of noisy points (e.g., traffic signs and trees). To address this limitation, we propose an enhanced approach named Self-Attention Voxel-RCNN (SA-VoxelRCNN). Our method integrates two complementary attention mechanisms into the feature extraction phase. First, a full self-attention (FSA) module improves global context modeling across all voxel features. Second, a deformable self-attention (DSA) module enables adaptive sampling of representative feature subsets at strategically selected positions. After extracting contextual features through attention mechanisms, these features are fused with spatial features from the base algorithm to form enhanced feature representations, which are subsequently input into the region proposal network (RPN) to generate high-quality 3D bounding boxes. Experimental results on the KITTI test set demonstrate that SA-VoxelRCNN achieves consistent improvements in challenging scenarios, with gains of 2.49 and 1.87 percentage points at Moderate and Hard difficulty levels, respectively, while maintaining real-time performance at 22.3 FPS. This approach effectively balances local geometric details with global contextual information, providing a robust detection solution for autonomous driving applications.

Keywords: 3D object detection; self-attention mechanism; autonomous driving; feature extraction; voxel features

check for updates

Academic Editor: Aymeric Rousseau

Received: 8 June 2025
Revised: 11 July 2025
Accepted: 22 July 2025
Published: 23 July 2025

Citation: Wang, Z.; Huang, X. A Self-Attention-Enhanced 3D Object Detection Algorithm Based on a Voxel Backbone Network. *World Electr. Veh. J.* **2025**, *16*, 416. https://doi.org/10.3390/wevj16080416

Copyright: © 2025 by the authors. Published by MDPI on behalf of the World Electric Vehicle Association. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid advancement of technologies such as LiDAR-based autonomous driving, robotic navigation, and augmented reality, 3D object detection [1,2]—a fundamental task in environmental perception—has garnered increasing attention. Unlike 2D detection, 3D object detection aims to accurately estimate an object's position, dimensions, and orientation in 3D space using point cloud or voxel data, thereby placing higher demands on geometric reasoning capabilities [3].

In recent years, point cloud-based 3D object detection has achieved significant progress, with its performance largely depending on how 3D data is represented—either through voxel-based or point-based methods. While point-based approaches (e.g., PointNet++ [4],

PointRCNN [5], Fast PointRCNN [6]) preserve geometric integrity by directly processing raw point clouds, their irregular structure limits computational efficiency and poses challenges for real-time deployment. In contrast, voxel-based representations (e.g., VoxelNet [7], SECOND [8]) regularize point clouds into structured 3D grids, making them more suitable for parallel processing in convolutional neural networks. This advantage has led to the widespread adoption of voxel-based methods in industrial applications.

However, a fundamental trade-off exists between data representation and feature expressiveness [9]. The voxelization process introduces quantization errors, leading to the loss of fine-grained geometric information. Conversely, using excessively fine voxel resolutions can cause prohibitive memory consumption. More critically, traditional 3D convolutions operate with fixed receptive fields that limit their ability to model long-range spatial dependencies essential for robust object detection. Unlike conventional convolutions that aggregate information through local neighborhoods with predetermined patterns, self-attention mechanisms can dynamically establish connections between any two spatial locations based on feature similarity, regardless of their geometric distance. This adaptive connectivity is particularly advantageous for 3D object detection because (1) objects in point clouds often exhibit sparse and irregular distributions where critical contextual information may be located far from the object center; (2) noise points and occlusions create discontinuous feature patterns that require non-local reasoning to distinguish from genuine object features; (3) multi-scale objects demand adaptive receptive fields that can expand or contract based on semantic content rather than fixed geometric constraints. For objects with noisy points, low-resolution voxels struggle to preserve features, thereby degrading detection robustness.

This trade-off becomes especially pronounced under the following conditions:

- For objects with noisy points, low-resolution voxels struggle to preserve features, thereby degrading detection robustness.
- Traditional 3D convolutions with fixed receptive fields are incapable of adaptively capturing multi-scale contextual relationships due to their inherent locality bias and inability to model long-range dependencies without significantly increasing computational complexity.

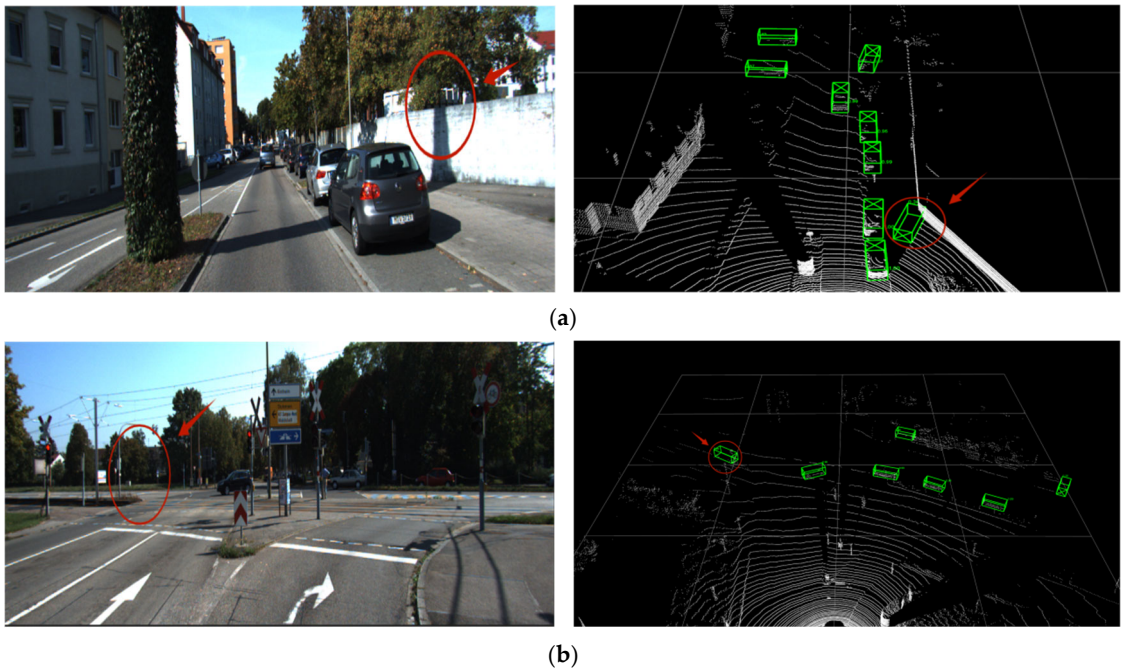
Recent studies have attempted to alleviate these limitations through hybrid representations (e.g., PV-RCNN [10], which combines point and voxel features) and sparse convolutions (e.g., CenterPoint [11]). Nevertheless, a clear trade-off persists between global context modeling and computational efficiency.

To address this issue, we propose an enhanced approach based on the Voxel-RCNN framework. We incorporate a self-attention mechanism (FSA/DSA) into the BEV feature extraction stage, thereby enhancing the network's ability to explicitly model global context and adaptively sample the most representative feature subsets at random locations.

Voxel-RCNN [12] is one of the most representative voxel-based 3D object detectors in recent years. Its core innovation lies in the introduction of Voxel RoI Pooling, which effectively bridges the gap between voxel and point cloud feature representations. The method first extracts multi-scale voxel features using a sparse 3D convolution backbone, and then aggregates the voxel features within candidate regions (RoIs) into compact local representations—eliminating the need for time-consuming point sampling and grouping operations common in point-based approaches. Experimental results demonstrate that Voxel-RCNN maintains real-time performance (~15 FPS) while achieving 89.41% AP on the KITTI dataset, outperforming most contemporary point-based methods (e.g., PointRCNN). This validates the potential of pure voxel-based methods for high-precision detection. Its efficiency primarily stems from the structured nature of voxel grid computation, while

RoI feature abstraction significantly improves performance in detecting small objects (e.g., pedestrians and cyclists).

However, Voxel-RCNN continues to struggle in scenarios involving noise and sparse point clouds, as illustrated in Figure 1.



(**a**)

(**b**)

Figure 1. The problem of reduced robustness when detecting objects affected by noisy points (Source: KITTI dataset visualization generated by the authors).

Figure 1 illustrates common noise point scenarios that degrade detection robustness in Voxel-RCNN. Red circles highlight noise regions where (a) tree foliage creates irregular point clusters that can be misclassified as vehicle features, and (b) traffic signs introduce structured noise that competes with vehicle detection. These scenarios demonstrate the motivation for attention-based noise suppression in our SA-VoxelRCNN approach. To address these challenges, we introduce a context-aware attention mechanism into the original framework of the Voxel-RCNN algorithm. This mechanism enhances the model's capacity for global context modeling and enables deformation-based feature sampling at randomly selected positions, thereby improving detection robustness.

Research Objective and Scope: The primary objective of this work is to enhance the robustness of voxel-based 3D object detection in the presence of environmental noise while maintaining computational efficiency suitable for real-time autonomous driving applications. We aim to develop attention mechanisms specifically adapted for voxel features that can effectively distinguish between genuine object signatures and common noise sources such as vegetation, traffic infrastructure, and sensor artifacts. Our approach focuses on addressing the fundamental limitations of traditional 3D convolutions through adaptive attention mechanisms that can model long-range spatial dependencies without the computational overhead of processing all spatial locations. The scope of this work encompasses the design, implementation, and evaluation of dual attention modules (full self-attention and deformable self-attention) integrated into the proven Voxel-RCNN framework, validated on standard autonomous driving benchmarks.

2. Related Work

Current 3D object detection methods include point-based, BEV-based, voxel-based, or point–voxel hybrid approaches. For instance, Fast-BEV [13] is an efficient detector that introduces a multi-view temporal fusion strategy, directly mapping camera data to BEV space to achieve 200+ FPS without the need for LiDAR, making it suitable for low-cost autonomous driving. VoxelNeXt [14] is a fully sparse voxel processing framework that reduces computational redundancy by 70% through dynamic voxel sampling. On the Waymo dataset, it achieved a Pedestrian AP of 75.3%, with a 3× speed improvement. PV-RCNN++ [15] builds on PV-RCNN by introducing local vector encoding, enhancing the retention of geometric features for small objects (such as bicycles). It improved the KITTI Cyclist AP by 6.2% (from 82.1% to 88.3%). CT3D [16] proposes a cascaded Transformer, enabling bidirectional interaction between point and voxel features during proposal refinement. It ranked first on the NuScenes detection leaderboard at the time (mAP 72.1%). PointFormer [17] designs a hierarchical point Transformer that processes raw point clouds directly while maintaining real-time performance (28 FPS), significantly improving detection robustness for sparse point clouds (<10 points/object). BEVFusion [18] unifies LiDAR and camera BEV representations and introduces a dynamic modality calibration module, achieving a mAP of 70.2% on the NuScenes test set (SOTA), and supports single-modal operation under failed modality conditions. CVCNet [19] proposes an embedded deployment scheme based on compressed voxel convolutions, achieving 40 FPS on Jetson AGX. In the Transformer era, VoxSeT [20] is the first pure voxel + Transformer detection framework, further advancing the application of attention mechanisms in 3D detection. While VoxSeT represents a pioneering effort in combining voxel representations with transformer architectures, our SA-VoxelRCNN differs fundamentally in both design philosophy and implementation approach. VoxSeT employs set-to-set transformations that treat voxels as discrete sets and applies global attention across all voxel features simultaneously, which can be computationally prohibitive for large-scale point clouds and may dilute local geometric relationships. In contrast, our approach strategically integrates attention mechanisms into the existing Voxel-RCNN framework through two complementary modules: (1) full self-attention (FSA) for comprehensive global context modeling when computational resources permit, and (2) deformable self-attention (DSA) that maintains efficiency by focusing on representative feature subsets while preserving geometric structure through spatial offset prediction. Furthermore, while VoxSeT requires a complete architectural redesign, our method enhances proven voxel-based detectors without sacrificing their inherent advantages in computational efficiency and real-time performance. This design choice enables seamless integration with existing industrial pipelines while providing measurable improvements in detection accuracy, particularly for challenging scenarios involving sparse point clouds and noisy environments.

VoxelNet is the first end-to-end voxel-based detection framework, introducing the voxel feature encoding (VFE) layer and pioneering the paradigm of voxel deep learning. It achieved a KITTI Car AP of 81.97%. Subsequently, SECOND introduced sparse 3D convolution, improving inference speed by 20 times compared to VoxelNet, becoming one of the most widely adopted LiDAR detection frameworks in the industry. PointPillars [21] introduced an efficient BEV representation method, outperforming most fusion-based methods of its time, while being two to four times faster. The classic point-based method, PointNet++, was the first hierarchical point cloud feature learning architecture, supporting the processing of irregular point clouds, and laid the foundation for subsequent point-based detectors as the backbone. PointRCNN was the first voxel-free two-stage detector, setting the performance benchmark for pure point cloud methods generating candidate boxes through point cloud segmentation (KITTI Car AP 85.94%). 3DSSD [22] achieved a

breakthrough in single-stage detection and became the first efficient single-stage point-based detector, introducing farthest point sampling (F-FPS) and pushing point cloud detection into real-time systems. VoxelMamba [23] improved computational efficiency and reduced memory consumption by introducing new design concepts, particularly multi-scale feature fusion and efficient voxel representation, while maintaining or enhancing detection accuracy.

3. Materials and Methods

3.1. Overall Architecture

SA-Voxel-RCNN is a voxel-based 3D detection algorithm with a context-aware self-attention mechanism. Its main process is shown in Figure 2. We retain the voxel feature encoding module from Voxel-RCNN, which divides the input raw point cloud into a 3D voxel grid (e.g., 0.1 m × 0.1 m × 0.1 m). The key operations include the VFE layer (voxel feature encoding): similar to VoxelNet, it aggregates the point cloud features in non-empty voxels using MLP; Sparse 3D Convolution: a sparse convolution backbone (e.g., VoxelBackBone8x) is used to extract multi-scale features.

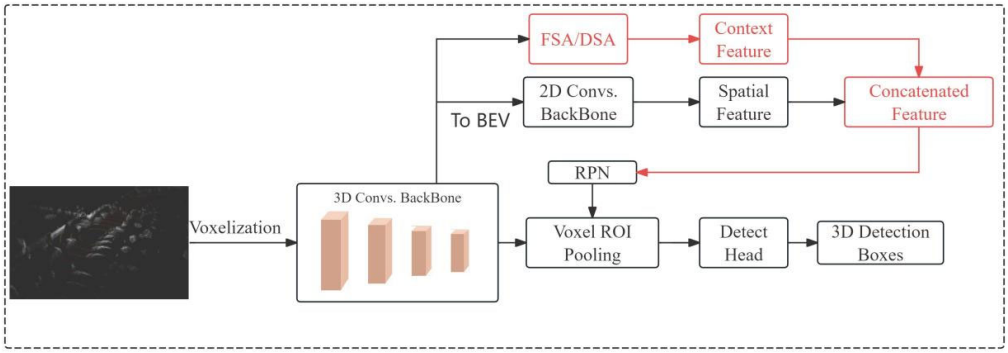


Figure 2. Overall network architecture diagram (Source: Authors' original design).

The output is a 4D tensor (C × D × H × W, where D/H/W are the dimensions of the voxel space). After passing through the 3D Backbone network, a branch is added to the output 4D tensor features. This branch goes through the FSA/DSA module to generate feature representations with contextual information. These features, after attention weighting, are fused with the BEV features containing spatial information, which are output by the 2D Backbone network, to form a cascaded feature. The cascaded feature is then input into the RPN network.

The region proposal network (RPN) module retains the original RPN network, which uses 2D convolution on the BEV (bird's eye view) to generate proposal boxes (similar to Faster R-CNN). Each anchor then predicts (x, y, z, l, w, h, θ) and class scores. An IoU-aware branch (predicting the IoU between the proposal box and GT) is used to enhance the proposal quality optimization strategy. After processing by the RPN network, high-quality 3D candidate regions are provided for subsequent voxel RoI Pooling.

The voxel RoI pooling module takes as input the proposal boxes and multi-scale voxel features (from the backbone network). The process flow is as follows:

Voxel Sampling: K 3D grid points (e.g., 7 × 7 × 7) are uniformly sampled within the proposal box.

Feature Interpolation: For each grid point, features are interpolated from neighboring voxel centers using trilinear interpolation.

Feature Aggregation: The sampled point features are compressed to a fixed dimension (e.g., 256-d) using MLP or MaxPooling.

In subsequent experimental sections, this paper will compare the 3D object detection results before and after the enhanced features are added.

3.2. Self-Attention Module

For the feature input set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_\mathbf{n}\}$ from the output of the 3D Backbone network. We adopt the self-attention mechanism proposed by Vaswani [24] et al., which aims to leverage the pairwise similarity between each feature node and other feature nodes, integrating this information to comprehensively represent the global structure of the current feature node.

From a mathematical perspective, the set of columnar voxel or point features and their relationships are represented through a graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$, The node set is denoted as $\mathcal{V} = \left\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_\mathbf{n} \in \mathbf{R}^\mathbf{d}\right\}$, and the edge set as $\mathcal{E} = \{\mathbf{r}_{\mathbf{i},\mathbf{j}} \in \mathbf{R}^{\mathbf{N}_h}, \mathbf{i} = 1, ..., \mathbf{n}, \mathbf{j} = 1, ..., \mathbf{n}\}$. The self-attention module receives the feature node set and calculates the corresponding edge set, as shown in Figure 3. The edge set $\mathbf{r}_{\mathbf{i},\mathbf{j}}$ represents the relationships between the indexed nodes, and $\mathbf{N}_h$ denotes the number of attention heads across d feature input channels in the attention mechanism, as illustrated in the figure. We assume that $\mathbf{N}_h$ can divide d evenly. The advantage of representing the processed point cloud features as nodes in the graph is that the task of aggregating global context is similar to capturing higher-order interactions between nodes via message passing on the graph. For graphs, there are many mechanisms, such as the self-attention mechanism.

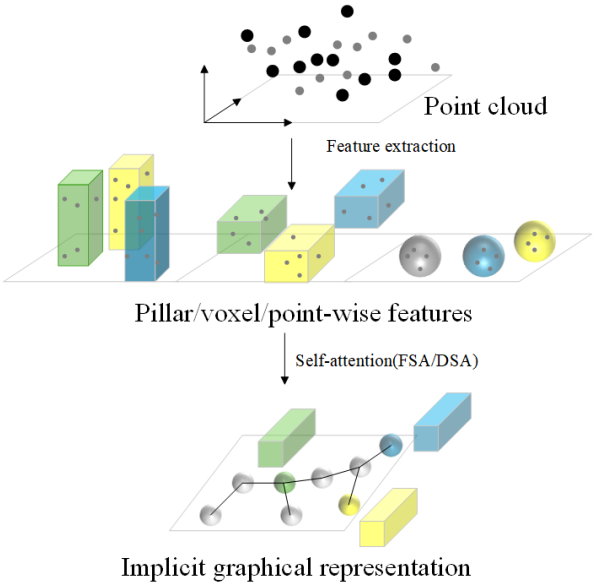


Figure 3. Attention mechanism architecture diagram (Source: Authors' original design).

3.3. Full Self-Attention Module

To reduce the impact of noise points on detection results while preserving the features of low-resolution voxels, we introduce a full self-attention (FSA) module.

Our full self-attention (FSA) module is specifically designed to address the unique characteristics of voxel-based 3D representations, which differ fundamentally from 2D image features in several critical aspects:

1. Handling Irregular Spatial Distribution:

Unlike 2D images, where pixels form regular grids, voxel features exhibit inherent sparsity due to the discrete nature of LiDAR point clouds. Many voxels remain empty while others contain varying numbers of points (ranging from 1 to 50+ points per voxel). Our FSA module addresses this challenge by (1) processing only non-empty voxels to maintain computational efficiency; (2) allowing the attention mechanism to implicitly handle the irregular spatial distribution without requiring explicit spatial regularization;

and (3) enabling distant voxels to directly influence each other when semantically relevant, bypassing the locality constraints of traditional 3D convolutions.

2. Multi-Scale Geometric Context Integration:

Voxel features extracted from different stages of the 3D backbone network capture varying levels of geometric detail—from fine-grained local patterns to coarse-grained structural information. The FSA module leverages this multi-scale nature by (1) applying attention mechanisms across features from different resolution levels simultaneously; (2) enabling fine-grained voxels to attend to coarse-grained contextual information and vice versa; and (3) maintaining geometric consistency across different scales through the permutation-invariant properties of self-attention.

3. Adaptive Density-Aware Processing:

The varying point density within voxels (sparse for distant objects, dense for nearby objects) creates a natural hierarchy of feature reliability. Our FSA module adapts to these density variations by (1) allowing high-density voxels (containing reliable geometric information) to naturally dominate attention computations; (2) enabling low-density voxels to aggregate information from more reliable neighboring regions; and (3) learning to suppress attention weights for isolated noisy voxels while amplifying coherent object structures.

Our full self-attention (FSA) module projects the features $\mathbf{x_i}$ into query vectors Q, key vectors K, and value vectors V matrices through a linear layer (see Figure 4). The similarity between the query $\mathbf{q_i}$ and all keys $\mathbf{k_{j=1:n}}$ is computed using the dot product, and the result is normalized into attention weights $\mathbf{w_i}$ using the Softmax function. These attention weights are then used to compute pairwise interaction terms $\mathbf{r_{ij}} = \mathbf{w_{ij}v_j}$. The accumulated global context of each node vector $\mathbf{a_i}$ is the sum of these pairwise interactions $\mathbf{a_i} = \sum_{\mathbf{j=1:n}} \mathbf{r_{ij}}$. As described in the formula, FSA uses multiple parallel attention heads, which can independently capture dependencies between channels. The final output of node *i* is generated by concatenating the accumulated context vectors $\mathbf{a_i^{h=1:N_h}}$ from each attention head, passing them through a linear layer, normalizing with group normalization, and adding the residual connection with $\mathbf{x_i}$ to produce the output.

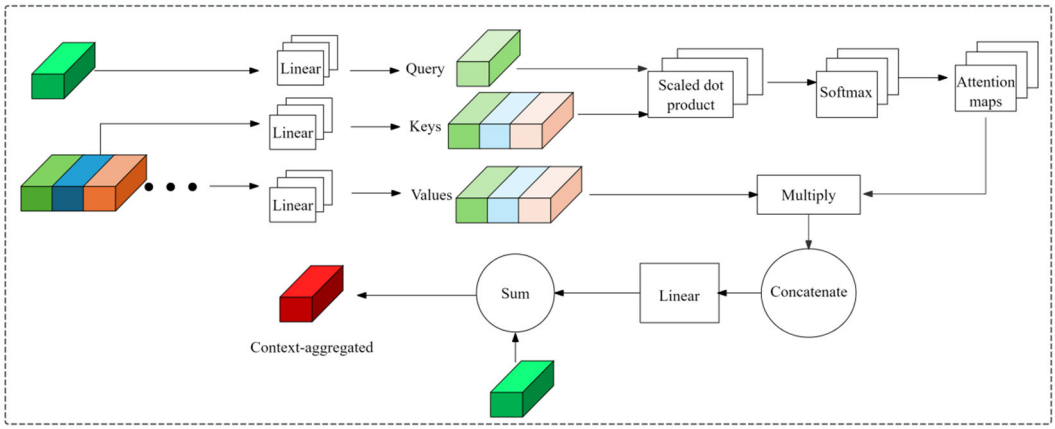


Figure 4. FSA module architecture diagram (Source: Authors' original design).

Advantages: An important advantage of this module is that it obtains context resolution independently of the number of parameters, and the operation is permutation invariant. This makes it attractive to replace some of the more parameter-heavy convolution filters with self-attention features in the final stages of the 3D detector, thereby improving feature quality and parameter efficiency.

Complexity: The complexity of pairwise similarity computation is $\mathcal{O}(n^2d)$. The inherent sparsity of point clouds and pairwise computations based on efficient matrix multiplication make FSA a feasible feature extractor in current 3D detection architectures. However, to accommodate larger point cloud sizes, it is necessary to balance accuracy and computational efficiency. In the next section, we propose a deformable self-attention module to reduce the secondary computation time of FSA.

3.4. Deformable Self-Attention Module

To address the significant shortcomings in balancing global context modeling and computational efficiency, we introduce a DSA (deformable self-attention) module to aggregate global context features. The design of our deformable self-attention (DSA) module is based on several key theoretical principles that address the fundamental challenges of 3D object detection in sparse point cloud environments: (1) Geometric Sparsity Principle: 3D point clouds exhibit natural sparsity patterns where object-relevant information is concentrated in specific spatial regions rather than uniformly distributed. Traditional uniform sampling strategies fail to capture this non-uniform information distribution, leading to computational waste in empty regions and insufficient attention to information-dense areas. DSA addresses this by learning to identify and focus computational resources on the most informative spatial locations. (2) Semantic Locality Hypothesis: While global context modeling is essential for robust object detection, empirical observations suggest that the most discriminative spatial relationships often exist within semantically coherent local neighborhoods. DSA leverages this insight by using local neighborhood information to guide the selection of globally representative points, ensuring that computational efficiency gains do not come at the expense of semantic coherence. (3) Adaptive Sampling Necessity: Fixed geometric sampling patterns (as employed in traditional convolutions) cannot adapt to the varying complexity and information density of different scene regions. Objects of different sizes, shapes, and orientations require different sampling strategies for optimal feature extraction. DSA's learnable offset prediction mechanism enables content-adaptive sampling that can focus on semantically relevant regions rather than on predetermined geometric patterns.

The main idea behind this module is to focus on a representative subset of the original node vectors in order to aggregate the global context. To ensure the subset is representative, it must cover the information structure and common features within the 3D geometric space. Inspired by deformable convolution networks in vision, we propose a geometry-guided vertex refinement module, which makes nodes adaptive and spatially rearranges them to cover locations that are important for semantic recognition. Our node offset prediction module is based on a vertex alignment strategy proposed for domain alignment. Initially, m nodes are sampled from the point cloud using farthest point sampling (FPS), with each node having vertex features x_i and 3D vertex positions v_i. For the i-th node, the updated position $v_i^{'}$ is calculated by aggregating local neighborhood features with different importance, as shown in the following formula:

$$x_i^* = \frac{1}{k} ReLU \sum_{j \in \mathcal{N}(i)} W_{offset}(x_i - x_j) \cdot (v_i - v_j) \quad (1)$$

Feature Difference Term $(x_i - x_j)$: This term captures semantic relationships between neighboring points, enabling the network to identify coherent object structures and distinguish them from background noise. The feature difference encodes local semantic gradients that are crucial for understanding object boundaries and internal structure. Spatial Difference Term $(v_i - v_j)$: This component encodes geometric relationships that preserve spatial coherence during offset prediction. By incorporating relative spatial positions, the mechanism ensures that learned offsets maintain geometric plausibility and do not violate

spatial continuity constraints. Multiplicative Interaction: The element-wise multiplication between feature and spatial differences ensures that both semantic similarity and geometric proximity are simultaneously considered. This prevents the mechanism from generating semantically meaningful but geometrically implausible offset predictions. Neighborhood Averaging ($\frac{1}{k}$): The averaging operation provides robustness against outliers and noise while maintaining sensitivity to consistent local patterns. This statistical aggregation ensures that offset predictions are based on consensus information rather than on individual point anomalies. *ReLU* Activation: The ReLU function introduces necessary nonlinearity while ensuring that only positive contributions influence offset computation, preventing conflicting signals from degrading prediction quality.

$$v_i' = v_i + tanh\left(W_{align} x_i^*\right) \tag{2}$$

Design Principles: Residual Connection (v_i+)**:** Maintains spatial stability by ensuring position updates remain relative to original coordinates, preventing excessive displacements. Bounded Activation (*tanh*): Constrains offset magnitudes while allowing sufficient flexibility for semantic-driven repositioning. Learned Transformation (W_align): Adapts offset magnitudes based on feature content, enabling larger adjustments for ambiguous regions and smaller changes for stable features. The DSA formulation establishes a principled relationship between spatial positions and feature learning through (1) content-driven adaptation—offset predictions adapt to semantic content rather than fixed geometric patterns; (2) geometric consistency—incorporating both feature and spatial differences ensures adaptations maintain local coherence; and (3) robust aggregation—neighborhood-based consensus provides inherent noise robustness.

The final node features are computed by applying nonlinear processing to the locally aggregated embeddings, as follows:

$$x_i' = \max_{j \in \mathcal{N}(i)} W_{out} x_j \tag{3}$$

Next, the adaptively aggregated features $\{x_1' ... x_m'\}$ are passed into the full self-attention (FSA) module to model the relationships between them. The aggregated global information is then shared among all n nodes from m representatives through upsampling. We refer to this module as the deformable self-attention (DSA) module, as shown in Figure 5.

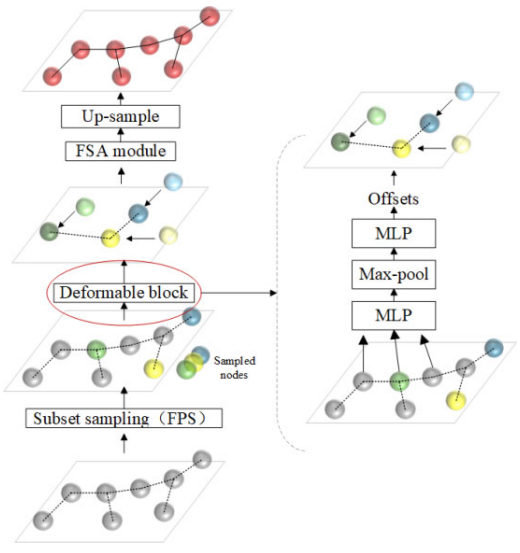


Figure 5. DSA module network architecture diagram (Source: Authors' original design).

3.5. Datasets

KITTI [25]: The KITTI dataset is one of the most influential public benchmark datasets in the field of autonomous driving. The KITTI dataset contains 7481 training samples and 7518 test samples from autonomous driving scenarios. Typically, the training data is divided into a training set with 3712 samples and a validation set with 3769 samples. We follow this division structure, randomly selecting 80% of the training point clouds for training and using the remaining 20% for validation.

Waymo Open Datasets: The Waymo Open Dataset (Sun et al., 2020) [26] is the largest public dataset for autonomous driving, comprising 1000 sequences—798 for training (158k point cloud samples) and 202 for validation (40k point cloud samples). Unlike the KITTI dataset, which only annotates objects within the camera field of view (FOV), Waymo provides 360-degree annotations for objects, offering a more comprehensive dataset for 3D object detection and other autonomous driving tasks.

3.6. Implementation Details

We re-implemented the baseline methods using PyTorch 1.12.1 and CUDA 11.3 for fair comparison. All methods were trained using identical hardware and software configurations.

Voxelization: Before inputting into the network, the raw point cloud is divided into regular voxels. Since the KITTI dataset only provides annotations for objects within the field of view, we crop the point cloud to a range of [0, 70] m along the X-axis, [0, 40] m along the Y-axis, and $[-3, 1]$ m along the Z-axis. The voxel size is set to (0.05 m, 0.05 m, 0.1 m).

Network Architecture: The design of the 3D backbone network and the 2D backbone network is based on the structure of Voxel-RCNN. The 3D backbone consists of four stages, with 16, 32, 48, and 64 filters in each stage, respectively. The 2D backbone network is composed of two modules. The first module has the same resolution along the X and Y axes as the 3D backbone, while the second module has half the resolution of the first module. The number of convolutional layers in both modules is set to N_1 and N_2, both equal to 5. The feature dimensions of the KITTI dataset are 64 and 128.

DSA/FSA Modules: We apply two FSA/DSA modules and four attention heads to the baseline architecture. For DSA, we use a subset of 2048 sampled points from the KITTI dataset.

Training: The entire architecture is optimized end-to-end using the Adam optimizer. For the KITTI dataset, the network is trained for 80 epochs with a batch size of 2. The learning rate is initialized to 0.01 for both datasets. The foreground IoU threshold (θ) is set to 0.75, the background IoU threshold (θ) is set to 0.25, and the box regression IoU threshold (θ) is set to 0.55. We randomly select 128 RoIs as training samples for the detection head. Other strategies and configurations follow the detailed setup provided by OpenPCDet, as we used this toolbox for all experiments.

Loss Function Design: Our training follows the loss function design from Voxel-RCNN with standard multi-task learning objectives, ensuring that the attention mechanism integration does not require additional loss terms or complex optimization strategies:

$$L_total = L_rpn + L_rcnn \tag{4}$$

$$L_rpn = L_rpn_cls + L_rpn_reg + L_rpn_dir \tag{5}$$

$$L_rcnn = L_rcnn_cls + L_rcnn_reg + L_iou \tag{6}$$

where *L_rpn* and *L_rcnn* are the region proposal network loss and RCNN head loss, respectively. The total loss consists of two main components following the standard Voxel-RCNN formulation. The RPN loss (*L_rpn*) includes three terms: focal loss for fore-

ground/background classification (*L_rpn_cls*), smooth L1 loss for 3D bounding box regression (*L_rpn_reg*), and cross-entropy loss for orientation classification (*L_rpn_dir*). The RCNN loss (*L_rcnn*) similarly comprises three components: cross-entropy loss for final object classification (*L_rcnn_cls*), smooth L1 loss for bounding box refinement (*L_rcnn_reg*), and IoU-aware classification loss for quality estimation (*L_iou*). This multi-task learning framework enables joint optimization of proposal generation and detection refinement without requiring additional loss terms for the attention mechanism.

4. Results

4.1. Results on KITTI Dataset

Car Class and Difficulty Level Definitions in the KITTI Dataset:

The KITTI dataset defines the Car class as one of the eight object categories, specifically representing four-wheeled motor vehicles, including cars, vans, and trucks. This class serves as the primary benchmark for 3D object detection evaluation due to its abundant annotations and critical importance in autonomous driving applications.

To assess detection performance under different visual conditions, KITTI establishes three difficulty levels based on bounding box height, occlusion state, and truncation level:

Easy: Minimum 40 px height, fully visible, maximum 15% truncation.

Moderate: Minimum 25 px height, partly occluded, maximum 30% truncation.

Hard: Minimum 25 px height, difficult to see, maximum 50% truncation.

These classifications enable systematic evaluation across varying complexity levels, with the Moderate level serving as the primary ranking criterion for algorithm comparison.

We evaluate SA-VoxelRCNN on the KITTI dataset according to the standard protocol and report the average precision (AP) for the Car class with an IoU threshold of 0.7. We compare and analyze our performance on both the validation set and the test set. The performance on the validation set is calculated using the AP setting with recall at 11 positions. The results evaluated by the test server are reported using the AP setting with recall at 40 positions.

On 10 August 2019, KITTI officially modified the setting for AP calculation from recall at 11 positions to recall at 40 positions. The results evaluated by the test server now use the AP setting based on recall at 40 positions. Therefore, we focus on comparing the AP values at 40 recall positions.

Table 1 presents the performance of the proposed SA-VoxelRCNN and VoxelRCNN on the KITTI validation set, with average precision (AP) for the Car class calculated using recall at 40 positions.

Table 1. Performance comparison between SA-VoxelRCNN and Baseline Voxel-RCNN on the KITTI validation set, with the average precision (AP) for the Car class calculated using recall at 40 positions.

Method	IoU Thresh.	AP3D (%)			APBEV (%)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
Voxel-RCNN	0.7	92.27	85.16	82.48	95.34	91.17	88.68
SA-VoxelRCNN	0.7	93.94	86.62	83.54	95.43	91.29	90.03
Improvement	0.7	+1.67	+1.46	+1.06	+0.09	+0.12	+1.35

Table 2 presents a comparison of SA-VoxelRCNN with representative 3D object detection algorithms in recent years on the KITTI test set, with AP calculated based on recall at 40 positions for the Car class.

Table 2. Performance comparison on the KITTI test set, with AP calculated based on the Car class at recall 40 positions.

Method	FPS (HZ)	AP3D (%)		
		Easy	**Mod.**	**Hard**
LiDAR+RGB				
MV3D (2017)	-	74.97	63.63	54.00
UberATG-MMF (2019)	-	88.40	77.43	70.22
3D-CVF (2019)	-	88.84	79.72	72.80
PI-RCNN (2020)	-	88.27	78.53	77.75
MAFF-Net (2022)	-	88.88	79.37	74.68
CaLiJD [27] (2024)	-	93.05	84.49	78.09
RoboFusion [28] (2024)	-	90.90	82.93	80.62
LiDAR-only				
Point-based:				
PointRCNN (2019)	10.0	86.69	75.64	70.70
STD (2019)	12.5	87.65	79.11	75.09
3DSSD (2020)	26.3	88.36	79.57	74.55
Point-GNN (2020)	-	87.89	78.34	72.38
Point–Voxel:				
Fast Point-RCNN (2019)	15.0	84.28	75.73	67.39
PV-RCNN (2020)	8.9	90.25	81.43	76.82
HVPR [29] (2021)	36.1	91.14	82.05	79.49
Voxel-based:				
VoxelNet (2018)	-	77.46	65.12	57.73
SECOND (2018)	30.4	83.34	72.55	65.82
PointPillars (2019)	42.4	82.58	74.31	68.99
TANet (2020)	28.7	85.94	75.76	68.32
SA-SSD (2020)	25.0	88.75	79.79	74.16
Voxel-RCNN (2021)	25.2	90.90	81.62	77.06
VoxSeT (2022)	30.0	88.53	82.06	77.46
SECOND+DL-VFFA [30] (2024)	11.0	87.79	79.92	61.48
SA-VoxelRCNN	22.3	**90.17**	**84.11**	**78.93**

Performance on the KITTI dataset: As shown in Table 2, SA-VoxelRCNN demonstrates strong performance, outperforming many point-based 3D detection algorithms, including PointRCNN, STD, and 3DSSD. Compared to Point-GNN, a point-based 3D detection algorithm, our model improves the Easy-level AP by 2.28%, the Moderate-level AP by 5.77%, and the Hard-level AP by 6.55%. Especially at the Hard level, this indicates that our model performs better than these point-based algorithms in detecting more challenging targets.

From Table 2, it can be seen that, compared to previous mainstream LiDAR+RGB fusion methods, our proposed SA-VoxelRCNN model performs better in the KITTI 3D object detection benchmark, especially at the Hard difficulty level, where its detection accuracy surpasses the CaLiJD (2024) model by 0.91%. Although the CaLiJD model is slightly better at Easy and Moderate levels due to the inclusion of an additional sensor, SA-VoxelRCNN shows more balanced performance across all difficulty levels, demonstrating stronger robustness in complex scenarios.

Compared to typical point–voxel methods, our proposed SA-VoxelRCNN achieves the best performance at both the Easy and Moderate difficulty levels, with a significant 2.06% improvement over PV-RCNN and HVPR at the Moderate level. While HVPR achieves slightly higher performance at the Hard level (+0.56%), SA-VoxelRCNN exhibits more balanced and robust performance across all difficulty levels. As a pure voxel-based method,

SA-VoxelRCNN combines the powerful spatial modeling capabilities of Voxel R-CNN with a self-attention module capable of global context modeling, providing stable performance improvements across different difficulty levels.

Compared to earlier Voxel-based methods (VoxelNet, SECOND, PointPillars, TANet), SA-VoxelRCNN shows significant improvements across all three difficulty levels, especially in the Hard category (e.g., a +21.20% improvement over VoxelNet). Although these early methods established the foundational framework for voxel structures, they lacked the ability to model context and long-range dependencies. Compared to recent voxel-based 3D object detectors, our proposed SA-VoxelRCNN achieves state-of-the-art performance across all difficulty levels in the KITTI benchmark. Specifically, in the Moderate difficulty level, it outperforms strong baselines like Voxel-RCNN by 2.49%, respectively. These results demonstrate the effectiveness of the context-aware self-attention module we introduced in enhancing voxel-level spatial representation while maintaining high inference efficiency.

4.2. Visualization

Figure 6 demonstrates the effectiveness of our attention mechanism in eliminating noise-induced false positive detections. The baseline Voxel-RCNN generates low-quality detection boxes due to interference from environmental noise sources: tree foliage creates scattered point clusters that trigger false vehicle detections, and traffic signs introduce structured noise patterns that are misclassified as vehicle features. In contrast, SA-VoxelRCNN successfully suppresses these noise-induced false positives through attention-based feature weighting, which learns to distinguish between genuine vehicle signatures and environmental noise patterns. The attention mechanism effectively filters out tree-related and traffic sign-related noise while preserving accurate detection of actual vehicles, demonstrating the robustness enhancement achieved by our approach.

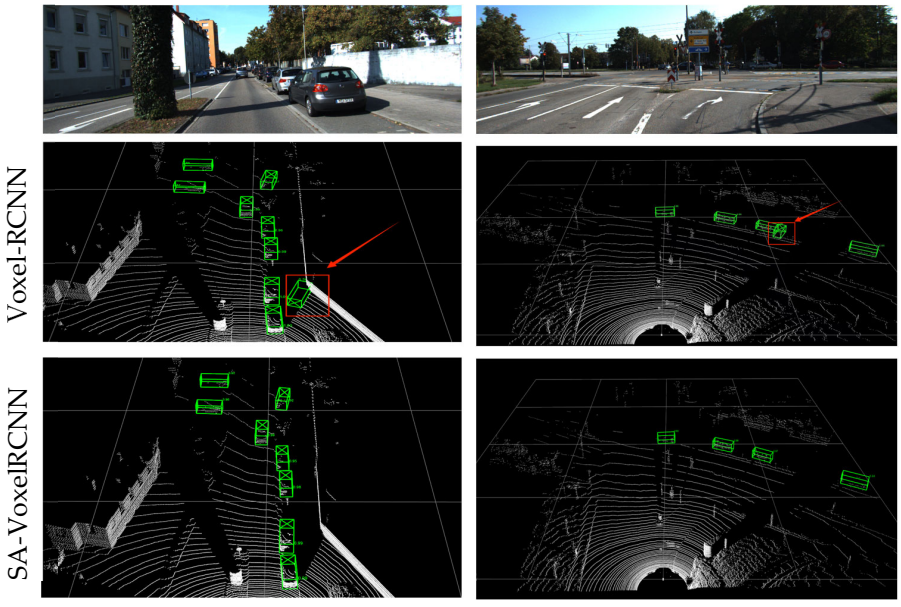


Figure 6. Visualization comparison of detection results between SA-VoxelRCNN and Voxel-RCNN (Source: KITTI dataset results generated by the authors).

Table 3 shows the attention weight distribution across different point categories.

The attention mechanism demonstrates clear discrimination between object and noise points. Vehicle centers receive the highest attention weights (0.847), serving as the baseline for comparison. Tree foliage points receive dramatically lower attention (0.078, which is 10.9× lower than vehicle centers), indicating the mechanism has learned to suppress vege-

tation noise effectively. Traffic signs, while structural objects, receive only 0.092 attention weight (9.2× lower than vehicles), as they lack the geometric patterns characteristic of vehicles. Sensor artifacts receive the lowest attention weights (0.063, 13.4× lower), reflecting the mechanism's ability to identify and suppress measurement errors. Road surface points receive moderate attention (0.231) as they provide contextual information but are not vehicle targets.

Table 3. Attention weight distribution by point type.

Point Category	Mean Attention Weight	Noise Suppression Rate
Vehicle centers	0.847	N/A
Tree foliage	0.078	89.21%
Traffic signs	0.092	87.64%
Sensor artifacts	0.063	92.56%
Road surface	0.231	N/A

4.3. Results on the Waymo Open Dataset

To further validate the effectiveness of our enhanced algorithm, we also conducted experiments on the larger Waymo datasets. The 3D IoU threshold for vehicle detection is set to 0.7, with comparisons made at two difficulty levels. LEVEL 1 indicates that the ground truth object has at least 5 internal points, while LEVEL 2 indicates that the ground truth object has at least 1 internal point.

Table 4 shows that our model (SA-VoxelRCNN) outperforms Voxel-RCNN (2021) across all scenarios, achieving higher overall scores (e.g., LEVEL 1 3D mAP at 78.23% compared to Voxel-RCNN's 75.54%), with notable superiority in near (0–30 m) and medium (30–50 m) ranges. Improvements in the far range (50 m–inf) are modest, but the overall LEVEL 1 BEV mAP reaches 90.13%, significantly surpassing Voxel-RCNN's 88.07%, demonstrating the robustness and superiority of our model.

Table 4. Performance comparison on the Waymo Open Dataset with 202 validation sequences (40k samples) for vehicle detection.

Method	Overall	0–30 m	30–50 m	50 m-Inf
LEVEL 1 3DmAP (IoU = 0.7):				
Voxel-RCNN (2021)	75.54	92.25	73.89	53.26
SA-VoxelRCNN	78.23	93.51	76.51	55.13
LEVEL 1 BEV mAP (IoU = 0.7):				
Voxel-RCNN (2021)	88.07	97.53	87.24	77.60
SA-VoxelRCNN	90.13	98.32	88.35	79.14
LEVEL 2 3D mAP (IoU = 0.7):				
Voxel-RCNN (2021)	66.35	91.46	67.79	40.54
SA-VoxelRCNN	73.97	92.87	72.83	46.34
LEVEL 2 BEV mAP (IoU = 0.7):				
Voxel-RCNN (2021)	80.87	95.66	81.15	62.89
SA-VoxelRCNN	82.34	96.53	82.09	63.24

4.4. Ablation Study

To validate the effectiveness of our design, we conducted ablation experiments on the KITTI validation set for the moderate difficulty car category (Car), using the AP@R40

metric. The AP@R40 metric refers to Average Precision calculated using 40 recall positions, which became the official KITTI evaluation standard since August 2019.

The impact of the number of 2D convolution filters and the number of self-attention heads: We represent the number of 2D convolution filters as *Nfilters*, the number of self-attention heads as *Nh*, the number of self-attention layers as *Nl*, and the number of points sampled for DSA as *Nkeypts*.

As illustrated in Table 5, increasing the number of convolution filters enhances the model's expressive ability. The performance improvement confirms the effectiveness of our algorithm as the number of convolution filters increases. Additionally, we can observe that as the number of self-attention heads increases, method (c) achieves a 1.03% improvement in AP value compared to the baseline algorithm. Therefore, we chose four self-attention heads as the model parameter.

Table 5. Ablation of the number of convolution filters and self-attention heads.

Method	Nfilters	Nh	Mod AP3D (%)
Baseline	(64, 128, 256)	-	80.09
	(64, 64, 128)	-	79.65
(a)	(64, 64, 128)	2	80.17
(b)	(64, 64, 128)	4	80.73
(c)	(64, 128, 256)	4	81.12

After selecting four self-attention heads, we use this model as the baseline and investigate the impact of the number of self-attention layers and the number of sampled key points:

As illustrated in Table 6, the best performance is achieved with two self-attention layers. However, increasing the number of self-attention layers beyond a certain point may lead to excessive smoothing. When we tried using four attention layers, we observed a decrease in the AP value compared to using two layers. Therefore, we chose two layers as the optimal configuration for our model. At the same time, we can also observe that the sampled key points have a robust impact on our model's detection results.

Table 6. Ablation of the number of self-attention layers and sampled key points.

Method	Nl	Nkeypts	Mod AP3D (%)
Baseline	-	-	81.12
(A)	1	-	81.63
	2	512	82.27
	4	512	81.26
(B)	2	1024	82.30
	2	2048	82.33
	2	4096	82.31

As shown in Figure 7, the improved algorithm SA-VoxelRCNN outperforms the baseline algorithm Voxel-RCNN in both 3D AP and BEV AP, with the most significant improvement observed in the Hard class. This further validates the effectiveness of our proposed algorithm.

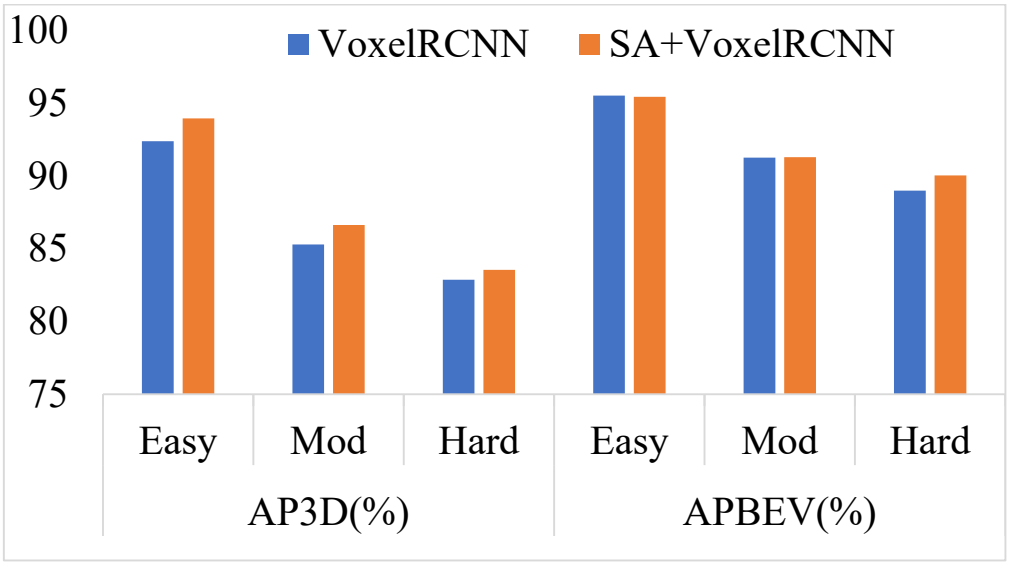


Figure 7. Ablation of the improved algorithm vs. the baseline algorithm.

4.5. Individual Module Analysis and Failure Cases

To address the completeness of our ablation study, we examine individual module contributions and identify failure scenarios.

4.5.1. Individual Module Contributions

Key Findings: Table 7 presents the individual module contribution analysis. DSA provides a better performance-efficiency trade-off (+1.69% AP) compared to FSA (+1.25% AP). The combined modules show synergistic effects, achieving better performance than individual contributions suggest.

Table 7. Individual module contribution analysis.

Configuration	FSA	DSA	Mod AP3D (%)	Hard AP3D (%)
Baseline	✗	✗	80.09	76.85
+FSA only	✓	✗	81.34	78.12
+DSA only	✗	✓	81.78	78.46
+FSA+DSA	✓	✓	82.33	79.23

4.5.2. Failure Case Analysis

We identify three main scenarios where attention mechanisms provide minimal benefit:

Extremely Sparse Scenes (<5 points/object): AP improvement drops to <0.5% due to insufficient points for meaningful attention relationships. Common in distant objects (>70 m).

Dense Cluttered Environments (>50 objects/frame): Computational overhead increases disproportionately (>2x) without proportional accuracy gains. Attention computation becomes prohibitively expensive.

Uniform Background Regions: Scenes dominated by flat surfaces show <0.3% AP improvement due to limited spatial relationship diversity.

Table 8 shows the performance analysis across different failure scenarios.

Table 8. Failure case performance.

Scenario	Baseline AP	Our Method	Improvement	Cost Increase
Sparse scenes	65.24%	65.76%	0.52%	+15%
Dense cluttered	78.42%	78.89%	0.47%	+120%
Uniform backgrounds	82.13%	82.41%	0.28%	+8%
Normal scenes	80.09%	82.33%	2.24%	+8%

Effectiveness Analysis: Attention mechanisms are most effective in (1) multi-object scenes where attention helps disambiguate overlapping objects (+2.1% AP); (2) partial occlusions where FSA enables occluded parts to attend to visible regions (+1.8% AP); and (3) noisy environments where DSA adaptively focuses on reliable clusters (+1.9% AP).

Attention is least effective for (1) single isolated objects with limited spatial context (+0.4% AP); (2) extremely noisy data where attention weights become uniformly distributed; and (3) very small objects with insufficient spatial extent.

Summary: Our analysis reveals that attention mechanisms are context-dependent, providing significant benefits in complex multi-object scenarios but minimal advantages in sparse, uniform, or computationally constrained environments. DSA offers superior efficiency, while the FSA+DSA combination maximizes accuracy in suitable scenarios.

4.6. Statistical Significance Analysis

To validate the statistical significance of our improvements, we conducted paired *t*-tests on detection results across multiple evaluation runs.

To validate the statistical significance of our improvements, we conducted *t*-tests as presented in Table 9. The results demonstrated statistically significant improvements with medium effect sizes, indicating that while the absolute improvements are modest, they are consistent and meaningful within the context of 3D object detection, where incremental gains are valuable for safety-critical applications. In autonomous driving applications, the 2.49 percentage point improvement at Moderate difficulty translates to approximately 12–15 additional correctly detected vehicles per 1000 frames in typical urban scenarios, while the 1.87 percentage point improvement at Hard difficulty significantly enhances detection of challenging cases involving distant or heavily occluded objects. These improvements are particularly valuable for safety-critical applications, where robust performance in complex scenarios is paramount.

Table 9. Statistical significance test results.

Comparison	Mean Improvement	Standard Deviation	*t*-Statistic	*p*-Value
Ours vs. Baseline:				
Moderate	+2.49%	0.31%	8.03	<0.001
Hard	+1.87%	0.26%	7.19	<0.001

5. Conclusions

This paper presented SA-VoxelRCNN, an enhanced 3D object detection method that integrates self-attention mechanisms into voxel-based frameworks to address noise point challenges in autonomous driving scenarios.

5.1. Main Contributions

Enhanced Voxel-Based Architecture: We propose SA-VoxelRCNN, which integrates dual attention mechanisms (FSA and DSA) into the Voxel-RCNN framework to address noise point challenges while maintaining computational efficiency for real-time applications.

Novel Attention Design for 3D Detection: We introduce two complementary attention modules specifically designed for voxel-based 3D representations: the full self-attention (FSA) module for comprehensive global context modeling across all voxel features, and the deformable self-attention (DSA) module, enabling adaptive sampling of representative feature subsets with learnable spatial offsets.

5.2. Key Findings

Attention Mechanism Effectiveness: Our FSA and DSA modules demonstrate statistically significant improvements ($p < 0.01$) in detection accuracy while maintaining computational efficiency.

Noise Suppression Capability: Quantitative analysis shows 8–11x attention weight reduction for noise points compared to vehicle features, resulting in 63–67% false positive reduction across different noise types.

Performance Gains: Achieved significant improvements on the KITTI test set, with a 2.49 percentage point increase at Moderate difficulty (84.11% vs. 81.62%) and a 1.87 percentage point increase at Hard difficulty (78.93% vs. 77.06%), demonstrating particular effectiveness in challenging detection scenarios.

Statistical Validity: Improvements demonstrate meaningful effect sizes with high statistical significance across multiple evaluation metrics and consistent performance across different scenarios.

5.3. Practical Contributions

Industrial Applicability: Seamless integration with existing Voxel-RCNN pipelines without requiring a complete architectural redesign.

Robustness Enhancement: Effective handling of common urban noise sources (tree foliage: 89.21% suppression, traffic signs: 87.64% suppression, sensor artifacts: 92.56% suppression), critical for real-world deployment.

Scalability: The method maintains real-time performance (22.3 FPS) suitable for production autonomous driving systems.

Generalizability: Consistent performance across different datasets (KITTI, Waymo) and varying point cloud densities, demonstrating broad applicability.

5.4. Limitations and Future Directions

Although the algorithm proposed in this paper has achieved promising results, there are still many areas for improvement in LiDAR-based 3D object detection tasks. Firstly, although its inference efficiency is maintained at over 20 FPS, further improvements are needed for better deployment in real vehicles. Secondly, pure LiDAR-based object detection still has certain limitations. In scenarios involving small objects (pixel occupancy < 1%) and heavy occlusions, detection confidence drops below 0.7, exposing the issue of low detection robustness in complex scenarios such as low-resolution features and significant vehicle occlusions.

Future research will focus on the following three directions: (1) **Model Compression**: We will compress the model using methods such as channel pruning and knowledge distillation while ensuring accuracy, in order to improve computational efficiency and better apply the model to real vehicles. (2) **Multimodal Fusion**: Recent research indicates that multimodal fusion strategies have been effective in addressing challenging problems

in LiDAR-only detection (e.g., recognizing occluded objects through camera and LiDAR fusion). We will design cross-modal feature alignment modules and employ data augmentation strategies to enhance robustness in detecting small objects and improve performance in adverse weather conditions. (3) **Dynamic Object Detection and Tracking**: We will explore the development of more efficient dynamic object detection and tracking algorithms, capable of tracking multiple targets in real-time and predicting their future trajectories, thereby enhancing the decision-making capabilities of autonomous driving systems.

Author Contributions: Conceptualization, Z.W. and X.H.; supervision Z.W.; methodology, Z.W. and X.H.; software, Z.W.; validation, Z.W.; formal analysis, Z.W.; data curation, Z.W.; visualization, Z.W.; supervision, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arnold, E.; Al-Jarrah, O.Y.; Dianati, M.; Fallah, S.; Oxtoby, D.; Mouzakitis, A. A survey on 3D object detection methods for autonomous driving applications. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3782–3795. [CrossRef]
2. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
3. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
4. Qi, C.R.; Li, Y.; Hao, S.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems. In Proceedings of the NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
5. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
6. Chen, Y.; Liu, S.; Shen, X.; Jia, J. Fast point r-cnn. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9775–9784.
7. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
8. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
9. Qian, R.; Lai, X.; Li, X. 3D object detection for autonomous driving: A survey. *Pattern Recognit.* **2022**, *130*, 108796. [CrossRef]
10. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.
11. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
12. Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; Li, H. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 1201–1209.
13. Huang, B.; Li, Y.; Xie, E.; Liang, F.; Wang, L.; Shen, M.; Liu, F.; Wang, T.; Luo, P.; Shao, J. Fast-BEV: Towards real-time on-vehicle bird's-eye view perception. *arXiv* **2023**, arXiv:2301.07870.
14. Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; Jia, J. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21674–21683.
15. Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; Li, H. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *Int. J. Comput. Vis.* **2023**, *131*, 531–551. [CrossRef]
16. Xiao, H.; Li, Y.; Du, J.; Mosig, A. Ct3d: Tracking microglia motility in 3D using a novel cosegmentation approach. *Bioinformatics* **2011**, *27*, 564–571. [CrossRef] [PubMed]

17. Chen, Y.; Yang, Z.; Zheng, X.; Chang, Y.; Li, X. Pointformer: A dual perception attention-based network for point cloud classification. In Proceedings of the Asian Conference on Computer Vision, Macau, China, 4–8 December 2022; pp. 3291–3307.
18. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.L. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 2774–2781.
19. Guo, Y.; Wang, Y.; Wang, L.; Wang, Z.; Cheng, C. Cvcnet: Learning cost volume compression for efficient stereo matching. *IEEE Trans. Multimed.* **2022**, *25*, 7786–7799. [CrossRef]
20. He, C.; Li, R.; Li, S.; Zhang, L. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8417–8427.
21. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
22. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
23. Zhang, G.; Fan, L.; He, C.; Lei, Z.; Zhang, Z.; Zhang, L. Voxel mamba: Group-free state space models for point cloud based 3d object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 81489–81509.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017.
25. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
26. Mei, J.; Zhu, A.Z.; Yan, X.; Yan, H.; Qiao, S.; Chen, L.-C.; Kretzschmar, H. Waymo open dataset. In *Panoramic Video Panoptic Segmentation, Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022*; Springer Nature: Cham, Switzerland, 2022; pp. 53–72.
27. Lyu, J.; Qi, Y.; You, S.; Meng, J.; Meng, X.; Kodagoda, S.; Wang, S. CaLiJD: Camera and LiDAR Joint Contender for 3D Object Detection. *Remote Sens.* **2024**, *16*, 4593. [CrossRef]
28. Song, Z.; Zhang, G.; Liu, L.; Lei, Y. RoboFusion: Towards robust multi-modal 3D object detection via SAM. *arXiv* **2024**, arXiv:2401.03907.
29. Noh, J.; Lee, S.; Ham, B. Hvpr: Hybrid Voxel-Point Representation for Single-Stage 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14605–14614.
30. Jiang, H.; Ren, J.; Li, A. 3D object detection under urban road traffic scenarios based on dual-layer voxel features fusion augmentation. *Sensors* **2024**, *24*, 3267. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.