Neural Processing Letters (2024) 56:23
https://doi.org/10.1007/s11063-024-11447-w

Improving 3D Object Detection with Context-Aware and Dimensional Interaction Attention

Jing Zhou[1] · Zixin Gong[1] · Junchi Zhang[1]

Accepted: 25 December 2023 / Published online: 9 February 2024

Abstract
Recently, 3D object detection technology based on point clouds has developed rapidly. However, too few points of distant and occluded objects are scanned by the sensor, and thus these objects suffer from too insufficient features to be detected. This case damages the detection accuracy. Therefore, we constitute a novel 3D object detection with Context-aware and dimensional Interaction Attention Network (CIANet) to explore vital geometric cues for enriching the feature representation of the object, thus boosting the overall detection performance. Specifically, in the first stage, we employ the 3D sparse convolution to extract voxel features, and then construct a Channel-Spatial Hybrid Attention (CSHA) module and a Contextual Self-Attention (CSA) module to enhance voxel features for generating proposals. The CSHA module aims to enhance the key information of the channel and spatial domains of 2D Bird's Eye View (BEV) features, and the CSA module is applied to supplement contextual information to the enhanced BEV features, thus generating accurate proposals. In the second stage, we construct a Dimensional Interaction Attention (DIA) module to refine Region of Interest (RoI) features within the proposals. It enhances the interactions among the channel and spatial dimensions of RoI features to learn accurate boundaries of objects for proposal refinement. Extensive experiments on the KITTI and Waymo benchmarks show the superior detection performance of CIANet compared to recent methods, especially for objects such as pedestrians and cyclists.

Keywords 3D object detection · Attention mechanism · Contextual information · Dimensional interaction attention

1 Introduction

3D object detection is one of the key technologies in the field of autonomous driving and has received wide attention. Recently, the points scanned by the LiDAR sensor have become the main input for 3D object detection [1]. However, in real scenes, the distribution of the point

✉ Jing Zhou
zhj131@jhun.edu.cn

[1] School of Artificial Intelligence, Jianghan University, Wuhan 430056, China

cloud is sparse, irregular, and unbalanced, and thus LiDAR-based 3D object detection still faces great challenges.

The current advanced 3D object detection methods can be classified into view-based [2–4], point-based [5–8], and voxel-based [9–12] approaches. The view-based methods project point clouds onto 2D views and then apply well-developed 2D Convolutional Neural Networks (CNN) networks on different views for 3D detection. However, the 3D spatial information is compressed and lost due to these projection operations, which limits the detection performance of 3D objects. Subsequently, point-based methods are proposed to reserve the spatial information of the original point cloud. They extract point-wise features directly from the original point cloud to generate accurate detection boxes. However, these methods spend too much computation to reconstruct the neighborhood points, thus leading to low inference speed. To reduce the computation and accelerate the inference speed, the voxel-based methods convert the point clouds into regular voxels and then extract the voxel feature with 3D sparse convolution. Such operations preserve the 3D spatial information and reduce the computational complexity at the same time.

However, in real scenes, some weak objects are scanned with too few points to depict their complete boundary, and they cannot provide sufficient spatial features. In this case, there is not enough attention to focus on the boundaries, making it difficult for the above methods to detect these weak objects, and thus damaging the overall detection performance.

To address the above issue, we introduce the attention mechanism to compute the spatial contextual correlation among different parts of objects, so as to guide the network focus on those parts that lack points to depict the boundary, and thus determining the boundaries of the objects more clearly. That is, we establish an attention-based 3D object detection network CIANet, which effectively improves the overall detection performance, especially for weak objects with small sizes like pedestrians and cyclists.

Specifically, for the first stage of the efficient voxel-based network Voxel-RCNN [12], which compresses the 3D spatial features into BEV to generate the initial proposals and thus ignores the global spatial contextual association, we first construct the Channel-Spatial Hybrid Attention (CSHA) module to model channel interdependencies and explore important spatial information of BEV features, followed by a 2D CNN to further extract the enhanced features. Then we construct the Contextual Self-Attention (CSA) module to extract the global spatial contextual cues among different parts of objects, which is supplied to the enhanced BEV features. In this way, the sparse boundaries with too few points of objects are enhanced to generate high-quality proposals. In addition, in the second stage of the network, we first employ the voxel RoI pooling operation to capture the RoI feature of the object, and then we construct the Dimensional-Interaction Attention (DIA) module to extract the internal association among different dimensions of the feature, thus highlighting the RoI feature for refining the proposal. Attributed to the above well-designed modules based on the attention mechanism, CIANet can focus on the boundary information of weak objects in real scenes, and localize these objects accurately, thus improving the overall detection performance.

The main contributions of this work are as follows:

- We present the Channel-Spatial Hybrid Attention (CSHA) module and Contextual Self-Attention (CSA) module in the first stage, which highlight vital channel-spatial features and aggregate rich global contextual information of objects to generate more accurate proposals.
- We design a Dimensional-Interaction Attention (DIA) module in the second stage. It integrates the interactions between the channel dimension and the spatial dimensions of

objects to enhance RoI features, thus refining the proposals to generate the final accurate detection boxes.

- Our proposed novel CIANet achieves better 3D object detection performance than other advanced methods on the KITTI and Waymo benchmarks, especially for weak objects with small sizes like pedestrians and cyclists. Experimental results show that our proposed attention modules can enrich the feature representation of the object and improve detection performance.

2 Related Work

2.1 3D Object Detection with LiDAR Point Clouds

Existing 3D object detection methods based on the LiDAR point cloud can be divided into three categories, including view-based, point-based, and voxel-based methods. The view-based methods [2–4] convert the point cloud into 2D views, which are then fed into 2D CNN for detection. In particular, MV3D [2] converts the point cloud into the bird's eye view and front view and then integrates the RGB image as input data for object detection. However, MV3D is difficult to be widely applied in real scenarios. To further improve the detection speed, PIXOR [3] implements quick and efficient object detection based on the bird's eye view of the scene. Furthermore, considering that the point cloud has a highly variable point density in the bird's eye view, MVF [4] fuses the perspective view that provides dense observation into the bird's eye view to utilize complementary information to generate accurate detection boxes. Overall, these view-based methods project the point cloud into 2D views, which result in the loss of point cloud depth information, thus affecting the detection performance.

Point-based methods [5–8] employ PointNet [13] or PointNet + + [14] to extract features from the original point cloud, avoiding information loss due to projection operation. The typical two-stage PointRCNN [5] employs PointNet + + to extract features and segment the foreground points to generate high-quality 3D proposals, and then it combines semantic features and local spatial features to refine the proposals. Subsequently, to reduce the excessive computational complexity of the two-stage network, single-stage detector 3DSSD [6] removes the time-consuming upsampling layer and refinement stage, and then proposes a new fusion sampling strategy to improve detection efficiency. However, these point-based methods take lots of computation to search neighboring points during the feature extraction process, which result in slow inference speed.

Voxel-based methods [9–12] convert the point cloud into regular voxels and employ the well-developed sparse CNN to extract features, thus improving the computation speed of the network. Typical method SECOND [9] adopts the 3D sparse convolution to extract the voxel feature, followed by a region proposal network to generate the detection box, which maintains fast detection speed. In addition, PointPillars [10] voxelized point clouds into pillars, which are then further encoded into 2D pseudo-images, followed by a 2D CNN to extract features, thus further accelerating the detection speed. However, SECOND and PointPillars lose spatial geometry information due to voxelization, which affects the detection accuracy. Some methods [15–17] address such problem by integrating the voxel features into key points, which lead to a rich feature representation with 3D structure information but increase the computation. Considering that the precise localization of the original points is not necessary, Voxel-RCNN [12] designs a voxel RoI pooling operation to aggregate 3D structure context from 3D voxel features, thus improving detection efficiency while preserving 3D spatial knowledge. Moreover, to further improve detection accuracy, CT3D [18] refines the proposals with a channel-wise Transformer architecture consisting of the proposal-to-point embedding operation, the self-attention-based encoder, and the channel-wise decoder, thus providing rich spatial semantic features to the detection head for generating detection boxes. These voxel-based methods have higher computational efficiency, and they adopt detailed refinement schemes to improve the detection performance of the network.

However, in real scenes, some weak objects are scanned with too few points to depict their complete boundary, and they cannot provide sufficient spatial features. In this case, there is not enough attention to focus on the boundaries, which makes it difficult for the above methods to detect these weak objects, decreasing overall detection accuracy. Therefore, we apply attention mechanisms upon the voxel-based backbone network to enhance critical boundary information of objects, thus further improving the overall detection performance of the network.

2.2 Attentions in Computer Version

In recent years, the attention mechanism has been successfully applied in computer vision fields, which can focus on key information and suppress irrelevant interference [19]. Traditional channel and spatial attention mechanisms [20–22] build attention masks to select the vital channels and spatial regions that need to be concerned, thus enhancing the representation of key features for objects. These traditional attention mechanisms only focus on local regions, hence, the self-attention mechanism is proposed to acquire global contextual information of objects. Furthermore, the Transformer structure based on self-attention [23] is proposed to capture long-range contextual information and highlight critical feature information of objects.

Due to the complexity of the scene in the 3D object detection task, it is essential to focus on the critical features of objects, so attention mechanisms are also introduced in 3D object detection. TANet [24] employs channel attention, point attention, and voxel attention strategies to extract the key information of objects in the scene, achieving convincing detection performance. SA-Det3D [25] establishes two self-attention modules to model the contextual information of the 3D object, which achieves superior object detection performance. Pointformer [26] explores and integrates local and global context-aware information to obtain dependencies between multi-scale representations. VoTr [27] adopts the Transformer to construct a backbone network for aggregating the information of the empty and non-empty voxels, thus expanding the non-empty voxel space. Then it utilizes the self-attention mechanism of the Transformer to capture the global context information among the expanded non-empty voxels, thus achieving the attention-weighted features with enhanced context in a larger receptive field.

The above attention mechanism can effectively improve the object detection performance. Therefore, in this work, we apply the attention mechanism on the baseline detector to enhance the critical boundary information of weak objects, thus accurately locating them and improving the overall detection performance.

3 CIANet for 3D Object Detection

In this section, we present the detailed design of the two-stage voxel-based detection network CIANet, where the first stage contains a 3D backbone network and a proposal generation

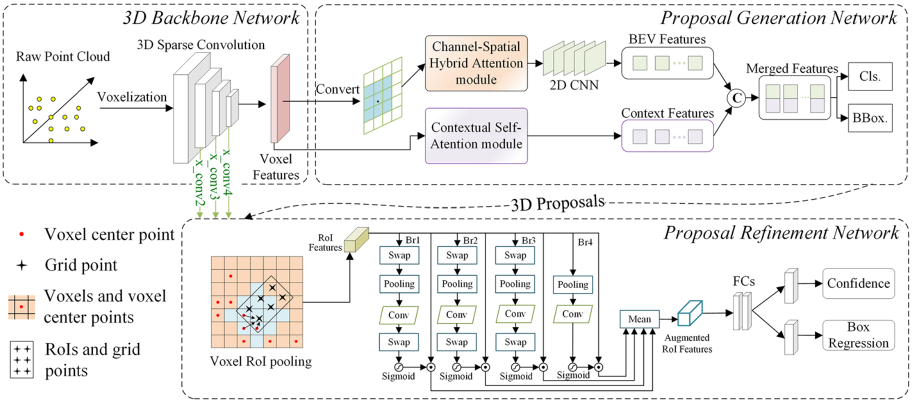

Fig. 1 An overview of CIANet. The whole network consists of two stages: the first stage extracts voxel features with a 3D backbone network, and then applies a proposal generation network composed of a Channel-Spatial Hybrid Attention (CSHA) module, a 2D CNN and a Contextual Self-Attention (CSA) module to generate high-quality proposals. In the second stage, the proposals are refined via voxel RoI pooling and the Dimensional-Interaction Attention (DIA) module to generate accurate 3D detection boxes

network, and the second stage contains a proposal refinement network. Figure 1. illustrates an overview of the CIANet, and in the first stage, the 3D backbone network first extracts voxel features from the point clouds. Then in the proposal generation network, we convert the voxel features into 2D BEV features and then construct the CSHA module to enhance the 2D BEV features, followed by the CSA module to further supplement the 2D BEV features with spatial contextual information, thus generating the proposals. In the second stage, we first take the voxel RoI pooling operation to extract the RoI features within the proposals, and then we design the DIA module to further highlight the RoI features for proposal refinement, thus producing accurate detection boxes. We will describe specific attention modules in the following sections.

3.1 Channel-Spatial Hybrid Attention Module

In the 3D backbone network of the first stage, we divide the original point cloud into m regular voxels, then employ the 3D sparse convolution to extract the high-level voxel feature with rich semantic information, which is fed into the proposal generation network to be converted into 2D BEV feature for proposal generation. Actually, the dependencies between channels are ignored by the 3D sparse convolution operation. Meanwhile, 3D spatial information is compressed during the converting process, resulting in that the key cues of the 2D spatial region are not distinct. Hence, we construct a novel Channel-Spatial Hybrid Attention (CSHA) module, which combines channel and spatial attention mechanisms to explore interdependencies among channels of the 2D BEV feature and highlight vital spatial information. The architecture of the CSHA module is shown in Fig. 2.

Specifically, we send the 2D BEV feature $F \in \mathbb{R}^{C\times H\times W}$ to the Channel-Spatial Hybrid Attention (CSHA) module, which contains a channel domain branch and a spatial domain branch. In the channel domain branch, we first apply the Global Average Pooling (GAP) operation with the size of $H \times 1$ to condense the spatial dimension H for achieving the feature $F_1 \in \mathbb{R}^{C\times 1\times W}$, and then utilize the max pooling operation with the size of $1 \times$

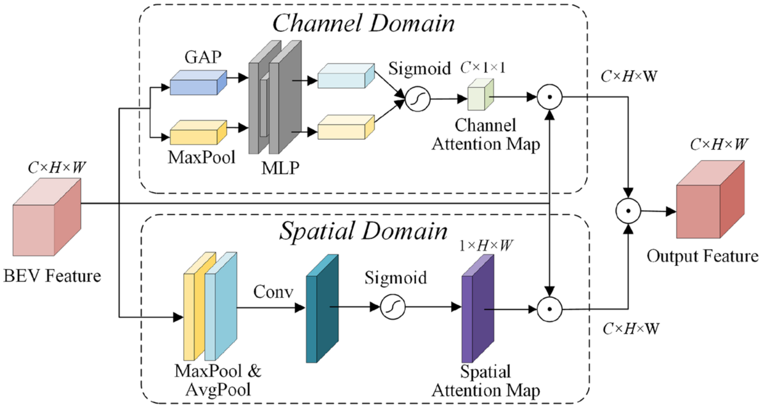

Fig. 2 Illustration of the architecture of the CSHA module, which consists the channel-domain branch and spatial-domain branch. It combines channel and spatial attention mechanisms to explore interdependencies among channels and highlight vital spatial information of the BEV feature

W along the spatial dimension W to generate the feature $F_2 \in \mathbb{R}^{C \times H \times 1}$. Then we send F_1 and F_2 to the Multi-Layer Perceptron (MLP) consisting of a dimension-reducing layer and a dimension-raising layer to generate the pooled feature $F_g \in \mathbb{R}^{C \times 1 \times 1}$. Then a sigmoid activation function is utilized to obtain the channel attention map, which is integrated with the feature F through element-wise multiplication to generate the channel-reweighted feature $F_c \in \mathbb{R}^{C \times H \times W}$, as shown in Eq. (1).

$$\begin{aligned} F_c &= \sigma(MLP(GAP(F))+MLP(MaxPool(F))) \odot F \\ &= \sigma(W_2(W_1(F_1))+W_2(W_1(F_2))) \odot F \end{aligned} \tag{1}$$

where $W_1 \in \mathbb{R}^{(C/r) \times C}$ and $W_2 \in \mathbb{R}^{C \times (C/r)}$ are the MLP weights, r is the reduction ratio.

In the spatial domain branch, we first update the feature $F \in \mathbb{R}^{C \times H \times W}$ by the max pooling and average pooling with the size of $C \times 1$ to achieve two feature maps of $F_m \in \mathbb{R}^{1 \times H \times W}$ and $F_a \in \mathbb{R}^{1 \times H \times W}$. They are concatenated on the channel dimension, followed by a convolution layer and a sigmoid activation function to obtain the spatial attention map $F_p \in \mathbb{R}^{1 \times H \times W}$. Then we compute the product between the spatial attention map F_p and F to get the region-reweighted feature $F_s \in \mathbb{R}^{C \times H \times W}$, as shown in Eq. (2).

$$F_s = \sigma(Conv([AvgPool(F); MaxPool(F)])) \odot F \tag{2}$$

Finally, we employ the element-wise multiplication to integrate the channel-reweighted feature F_c and region-reweighted feature F_s, thus achieving the output feature F_{out} with explicit channel dependencies and enhanced spatial cues, as shown in Eq. (3).

$$F_{out} = F_c \odot F_s \tag{3}$$

And then the feature F_{out} is updated by the 2D CNN referring to [9, 12]. In this way, the vital channel and spatial information of the 2D BEV feature is strengthened based on the CSHA module.

3.2 Contextual Self-Attention Module

In the proposal generation network, we employ the CSHA module to obtain the enhanced 2D BEV feature. However, it lacks global spatial contextual associations. Considering that high-level voxel features in the 3D backbone network contain uncompressed 3D spatial information, we further construct a CSA module, which captures the spatial context among different parts of objects that are embedded in the high-level voxel features, and then supplies it to the enhanced BEV feature, thus enhancing the insufficient boundary of the weak object with sparse points. The architecture of the CSA module is shown in Fig. 3.

Specifically, for m voxels divided from the original point cloud in the 3D backbone, we calculate their center points based on the index number of each voxel, thus obtaining a set of center points denoted as $Y_{center} \in \mathbb{R}^{m\times 3}$. Then we utilize the Farthest Point Sampling (FPS) operation to select n sampled points from the center points $Y_{center} \in \mathbb{R}^{m\times 3}$. For the sampled point $\overline{p}$, it has the point feature $U_{\overline{p}} \in \mathbb{R}^{C}$ and the 3D position $L_{\overline{p}} = (x^{(\overline{p})}, y^{(\overline{p})}, z^{(\overline{p})}) \in \mathbb{R}^3$, and we propose the bias estimation strategy to calculate the position bias from its neighboring points for updating its position. In this way, the neighboring contextual information is aggregated upon the point $\overline{p}$.

In detail, we first calculate the feature variation $\Delta U^q \in \mathbb{R}^C$ and the position variation $\Delta L^q \in \mathbb{R}^3$ between point $\overline{p}$ and the neighboring point q, as illustrated in Eq. (4).

$$\begin{cases} \Delta U^q = U_{\overline{p}} - U_q,\ q \in \varphi(\overline{p}) \\ \Delta L^q = L_{\overline{p}} - L_q,\ q \in \varphi(\overline{p}) \end{cases} \tag{4}$$

where $\varphi(\overline{p})$ indicates the set of d neighbor points for point $\overline{p}$, U_q is the point feature of point q, and L_q is the 3D position of point q.

Then we compute a weighted combination of the feature and position variations, thus achieving the final position bias $\Delta L_{\overline{p}}$ for the point $\overline{p}$, as denoted in Eq. (5).

$$\Delta L_{\overline{p}} = \frac{\sum_{q\in\varphi(\overline{p})} MLP(\Delta U^q)\cdot(\Delta L^q)}{\sum_{q\in\varphi(\overline{p})} MLP(\Delta U^q)} \tag{5}$$

After that, we add the final position bias to the initial position of the sampled point $\overline{p}$ to get the renewed position $L'_p \in \mathbb{R}^3$ of the updated point p, as shown in Eq. (6). The updated

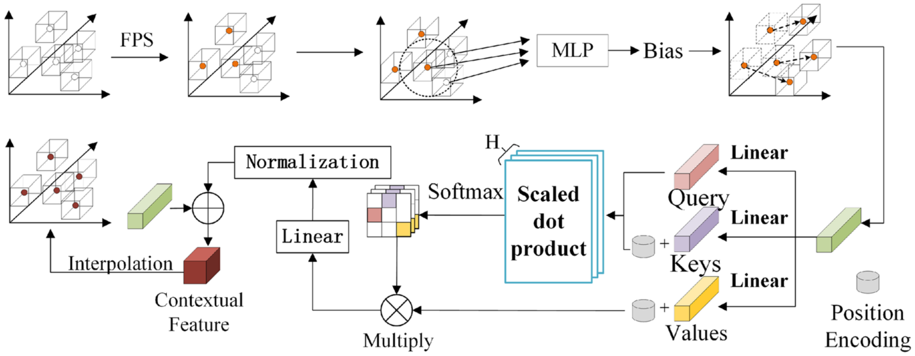

Fig. 3 The architecture of the CSA module. The voxelized original point cloud is sampled by the farthest point sampling operation, followed by the bias estimation strategy and self-attention operation with relative position encoding strategy to capture the spatial context embedded in high-level voxel features

points can cover common feature structures in the 3D space.

$$L'_p = L_{\overline{p}} + \Delta L_{\overline{p}} \tag{6}$$

Next, we aggregate the features of the neighboring points to achieve the renewed feature for the updated point, which is calculated as Eq. (7).

$$U'_p = \sum_{q \in \varphi(\overline{p})} \omega U_q \tag{7}$$

In this way, the features of all updated points are adaptively updated to achieve the aggregated feature $U' = \{u'_1, u'_2, \cdots, u'_n \in \mathbb{R}^C\}$.

To further explore the global contextual information, we employ the self-attention mechanism to calculate semantic interactions between the pair-wise aggregated features.

Exploring global contextual information for updated points by the self-attention mechanism is comparable to capturing semantic correlation between feature nodes while passing messages in a graph. Hence, we employ the graph $G = (\upsilon, \lambda)$ to describe the collection of aggregated features and their relationship, where $\upsilon = \{u'_1, u'_2, \cdots, u'_n \in \mathbb{R}^C\}$ represents a feature node set and $\lambda = \{r_{pg} \in \mathbb{R}^H\}$ represents an edge set. The r_{pg} denotes the relationship between feature node p and node g, and H represents the number of attention heads across C input channels.

Furthermore, considering that the relative position information between nodes contains accurate spatial dependencies, we introduce the relative position-coding strategy into the self-attention mechanism to extract the global contextual information more accurately. In detail, we multiply the aggregated feature of the updated point p of the single attention head with projection matrices $W_Q \in \mathbb{R}^{n\times n}$ to achieve query vector $Q_p \in \mathbb{R}^{n\times(C/H)}$, and then we multiply the semantic feature of the updated point g with the projection matrices $W_K \in \mathbb{R}^{n\times n}$, and $W_V \in \mathbb{R}^{n\times n}$ to achieve key vector $K_g \in \mathbb{R}^{n\times(C/H)}$ and value vector $V_g \in \mathbb{R}^{n\times(C/H)}$, respectively. Then we propose a relative position encoding strategy, it first restricts the maximum value of the Euclidean distance between coordinates of node p and node g as b, and then applies a linear layer to encode the coordinate distance into relative position information. The detailed calculation of position encoding is shown in Eq. (8).

$$a_{pg} = \text{linear}\Big(\min\Big(\Big\| L'_p - L'_g \Big\|, b\Big)\Big), g \in \psi \tag{8}$$

where $a_{pg} \in \mathbb{R}^{n\times(C/H)}$ is the encoded relative position feature and $\|\cdot\|$ denotes Euclidean distance, and ψ denotes the set of update points except p.

Subsequently, we embed the encoded relative position feature in K_g and V_g, and calculate the contextual correlation term r_{pg} between the feature node p and the feature node g, as shown in Eq. (9).

$$r_{pg} = \text{softmax}\left(\frac{Q_p\left(K_g + a_{pg}\right)^T}{\sqrt{C/H}}\right) \cdot \left(V_g + a_{pg}\right) \tag{9}$$

For the node p, we calculate the sum of its contextual correlation terms with other nodes to obtain the accrued term S_p, as shown in Eq. (10).

$$S_p = \sum_{g \in \psi} r_{pg} \tag{10}$$

Then we concatenate the accrued term S_p across attention heads, followed by a linear layer, a group normalization, and a residual connection to generate the global contextual feature of node p. Next, we share the global contextual features of n nodes to m nodes by interpolation, thus achieving the context features of m nodes. And each node feature represents the feature of the corresponding voxel where the node lies in.

At last, we merge the context features with the enhanced BEV features to achieve the merged features with global contextual association information, which are sent to a Region Proposal Network [28] for generating the proposals.

3.3 Dimensional-Interaction Attention Module

To refine the proposals, we construct a refinement network in stage 2, which consists of a voxel RoI pooling operation and a Dimensional-Interaction Attention (DIA) module. Specifically, we first divide the proposal into different grids and utilize the voxel RoI pooling operation to extract RoI features. Then, to further highlight the vital grid features that contribute to detection, we intend to employ the attention mechanism for feature enhancement, thus strengthening the RoI features for proposal refinement.

Considering the traditional channel attention mechanism compresses the spatial dimension to capture the dependencies among channel dimensions, which results in the loss of spatial information and thus the internal correlation between the spatial and channel dimensions is ignored. Although some approaches utilize spatial attention as a supplementary module to channel attention, they calculate channel attention and spatial attention separately, resulting in the internal correlation between the two attention dimensions not being captured. Thus, we construct a DIA module to capture the internal correlation among different dimensions of the feature, thus enhancing RoI features for proposal refinement. The architecture of the DIA module is shown in Fig. 4.

Specifically, we extract the RoI feature $T \in \mathbb{R}^{C \times L \times W \times H}$ within proposals by employing the voxel RoI pooling operation [12]. It divides the proposals into $L \times H \times W$ grids. For each layer of the 3D sparse convolution, the neighboring voxels of the center point in each grid are determined based on the Manhattan distance, and then a PointNet [13] is utilized to aggregate neighboring voxel features onto the center of the grid to obtain the aggregated features of this layer. Finally, the aggregated features of the last three layers are concatenated to obtain the RoI features $T \in \mathbb{R}^{C \times L \times W \times H}$.

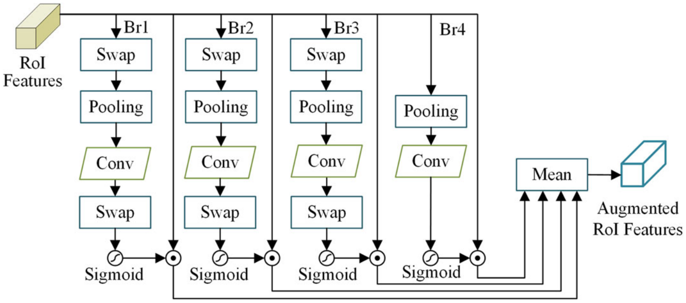

Fig. 4 The architecture of the DIA module. It consists of four branches, which include swap, pooling, convolution, and activation operations. The first three branches capture the internal correlation between the channel and spatial dimensions of RoI features, and the last branch explores the dependencies among spatial dimensions. Finally, we calculate the mean of the outputs of four branches to obtain augmented RoI features for proposal refinement

And then the RoI feature T is fed into the DIA module with four branches. The first branch establishes the interactions between the channel dimension C and spatial dimensions W,H. In detail, we first swap the channel dimension C and spatial dimension L of the RoI feature to get the feature $T_1 \in \mathbb{R}^{L\times C\times W\times H}$, followed by a max pooling and an average pooling operation to compress the first dimension of T_1 as two, thus obtaining the pooled feature $\widehat{T_1} \in \mathbb{R}^{2\times C\times W\times H}$. After that, we employ a 3D convolution layer and a batch normalization layer on $\widehat{T_1}$ to capture the contextual connection between the channel dimension C and the spatial dimensions W, H, generating the interaction attention feature $\widehat{T_1^*} \in \mathbb{R}^{1\times C\times W\times H}$. Then we swap the two dimensions to achieve the feature $T_1^* \in \mathbb{R}^{C\times 1\times W\times H}$, followed by a sigmoid activation function to obtain the interaction attention weight. At last, we multiply the weight by the feature T to get the re-weighted RoI feature B_1 with the same shape of T. The above process is described as Eq. (11).

$$B_1 = \sigma\big(P_1^*(\gamma(\chi(Pool(P_1(T)))))\big) \cdot T \tag{11}$$

where σ indicates the sigmoid activation function, χ and γ indicate the 3D convolution layer and batch normalization layer, respectively. P_1 and P_1^* represent two swap operations.

In the same way, we aim to build interactions between the channel dimension C and spatial dimensions L, H in the second branch. We first swap the dimensions of the RoI feature $T \in \mathbb{R}^{C\times L\times W\times H}$ and then compact its spatial dimension W by the pooling operations to achieve the feature $\widehat{T_2} \in \mathbb{R}^{2\times L\times C\times H}$. Subsequently, a 3D convolution layer and a batch normalization layer are employed on $\widehat{T_2}$ to generate the interaction attention feature $\widehat{T_2^*} \in \mathbb{R}^{1\times L\times C\times H}$. Furthermore, we swap the dimensions of $\widehat{T_2^*}$ and then adopt the sigmoid activation function to achieve the interaction attention weight, which is multiplied by T to obtain the re-weighted RoI feature B_2. The above process is denoted in Eq. (12).

$$B_2 = \sigma\big(P_2^*(\gamma(\chi(Pool(P_2(T)))))\big) \cdot T \tag{12}$$

where P_2 and P_2^* represent two swap operations.

Similarly, in the third branch, we utilize the same approach adopted in the above two branches to build interactions between the channel dimension C and spatial dimensions L, W, thus obtaining the re-weighted RoI feature B_3, as shown in Eq. (13).

$$B_3 = \sigma\big(P_3^*(\gamma(\chi(Pool(P_3(T)))))\big) \cdot T \tag{13}$$

where P_3 and P_3^* represent two swap operations.

In the last branch, we capture the spatial dependencies among spatial dimensions L, W and H. Firstly, we compress the channel dimension C of the RoI feature T through pooling operations to obtain the feature $T_4 \in \mathbb{R}^{2\times L\times W\times H}$, followed by a 3D convolution layer and a batch normalization layer to get the feature $T_4^* \in \mathbb{R}^{1\times L\times W\times H}$. And then the T_4^* is activated by the sigmoid function to get the attention weight, which is multiplied by T to achieve the re-weighted RoI feature B_4, as shown in Eq. (14).

$$B_4 = \sigma(\gamma(\chi(Pool(T)))) \cdot T \tag{14}$$

Subsequently, we obtain the final augmented RoI features by calculating the mean of the output features of the four branches, as described in Eq. (15).

$$D_{out} = \frac{1}{4}\sum_{i=1}^{4} B_i \tag{15}$$

Finally, we send the augmented RoI features into a 2-layer MLP, followed by two branches for confidence prediction and box regression to generate the accurate detection boxes.

3.4 Training Losses

Our CIANet is a two-stage detection network, which is trained in an end-to-end fashion. The overall loss includes two parts, one is the region proposal loss $\mathcal{L}_{RPN}$ in the first stage, and the other is the proposal refinement loss $\mathcal{L}_{RCNN}$ in the second stage, as shown in Eq. (16).

$$\mathcal{L}_{ALL} = \mathcal{L}_{RPN} + \mathcal{L}_{RCNN} \tag{16}$$

The $\mathcal{L}_{RPN}$ includes the loss of classification and box regression, as shown in Eq. (17).

$$\mathcal{L}_{RPN}=\frac{1}{\varepsilon}\left[\sum_i \mathcal{L}_{cls}\left(c_i^a, l_i^*\right) + \vartheta\left(l_i^* \geq 1\right)\sum_i \mathcal{L}_{reg}\left(r_i^a, o_i^*\right)\right] \tag{17}$$

where ε is the number of foreground anchors. For classification loss $\mathcal{L}_{cls}$, we apply the Focal Loss function [29] to calculate the loss between the classification output c_i^a and the classification label l_i^*. For the box regression loss $\mathcal{L}_{reg}$, we apply $\vartheta\left(l_i^* \geq 1\right)$ to select the foreground anchors and calculate their loss by Huber loss, which calculates the loss between the regression output r_i^a and the regression target o_i^*, as shown in Eq. (18).

$$\mathcal{L}_{reg}\left(r_i^a, o_i^*\right) = \begin{cases} \frac{1}{2}\left(r_i^a - o_i^*\right)^2, \ \left|r_i^a - o_i^*\right| \leq \delta \\ \delta \cdot \left(\left|r_i^a - o_i^*\right| - \frac{1}{2}\delta\right), otherwise \end{cases} \tag{18}$$

where δ is a hyperparameter calculated from r_i^a and o_i^* during training phase.

The $\mathcal{L}_{RCNN}$ includes the box regression loss and the IoU-guided confidence prediction loss of the second stage, as shown in Eq. (19).

$$\mathcal{L}_{RCNN} = \frac{1}{\rho}\left[\sum_i \mathcal{L}_{cls}\left(c_i, \partial_i^*(IoU_i)\right)+\vartheta\left(IoU_i \geq \mu_{reg}\right)\sum_i \mathcal{L}_{reg}\left(r_i, o_i^*\right)\right] \tag{19}$$

where ρ is the number of sampled 3D proposals during the training phase. The $\mathcal{L}_{reg}$ refers to box regression loss, which is implemented by the Huber Loss. The IoU_i represents the IoU score corresponding to the i^{th} proposal and its ground truth. We adopt $\vartheta\left(IoU_i \geq \mu_{reg}\right)$ to calculate the regression loss of the high-quality proposals, which are selected with $IoU_i \geq \mu_{reg}$. In addition, $\mathcal{L}_{cls}$ represents the classification loss based on the IoU score, which is implemented by the Binary Cross Entropy Loss function. The c_i denotes the output of classification and $\partial_i^*(IoU_i)$ denotes the target of classification loss, which is calculated as shown in Eq. (20).

$$l\partial_i^*(IoU_i) = \begin{cases} 0 \quad IoU_i < \mu_B, \\ \frac{IoU_i-\mu_B}{\mu_F-\mu_B}\ \mu_B \leq IoU_i < \mu_F, \\ 1 \quad IoU_i > \mu_F, \end{cases} \tag{20}$$

where μ_F and μ_B represent the foreground and background IoU thresholds, respectively.

4 Experiment

4.1 Datasets and Evaluation Metrics

We first train and test our CIANet on the KITTI dataset, which provides 7481 training LiDAR samples and 7518 test samples. Following the experimental scheme in [9, 12], we further divide the 7481 training samples into a training set of 3712 samples and a validation set of

3769 samples. Meanwhile, the objects in the KITTI dataset are classified into three categories of cars, pedestrians, and cyclists. Each category is divided into three difficulty levels of easy, moderate, and hard due to the degree of occlusion and truncation.

We evaluate the detection performance with the evaluation metrics of Average Precision (AP) and mean Average Precision (mAP), where AP is computed by averaging the precision of all predicted detection boxes at a given difficulty level and mAP is calculated by meaning the average precisions of predicted boxes among three difficulty levels. We calculate the AP and mAP values on the validation set with 11 recall positions and the AP values on the test set with 40 recall positions, respectively. Moreover, the IoU threshold is set to 0.7 for cars and 0.5 for pedestrians and cyclists.

Furthermore, we also conduct experiments on a large-scene dataset Waymo [30]. It has 798 training sequence data that contain 158,361 point cloud samples, and 202 validation sequence data that contain 40,077 samples. The samples of Waymo are split into two difficulty levels: LEVEL_1 and LEVEL_2, where LEVEL_1 objects have at least five LiDAR points and LEVEL_2 objects have at least one LiDAR point. In the Waymo dataset, we utilize the IoU of 0.7 for vehicles and 0.5 for pedestrians and cyclists. Meanwhile, we adopt the mean Average Precision (mAP) and the mean Average Precision weighted by Heading (mAPH) as evaluation criteria.

4.2 Implementation Details

Network Architecture. For the KITTI dataset, the ranges of X, Y, and Z axes of the 3D scene are [0, 70.4] m, [-40, 40] m, and [-3, 1] m. The scene is divided into 16,000 voxels for training, and 40,000 voxels for testing, and the size of each voxel is set to (0.05 m, 0.05 m, 0.1 m). For the Waymo dataset, the ranges of X, Y, Z axes of the 3D scene are within [-75.2, 75.2] m, [-75.2, 75.2] m, and [-2, 4] m, and the voxel size is set as (0.1 m, 0.1 m, 0.15 m). In the first stage of the network, we first apply the 3D backbone network that stacks four sparse convolution layers to recover the voxel feature dimensions as 16–32-64–64. Then in the CSHA module, since the smaller value of reduction ratio r will increase the computation of the model and the larger value may decrease accuracy, we set r to 16 to balance the complexity and accuracy. And then we apply two CSA modules with 4 attention heads to capture global spatial contextual associations of objects. In each CSA module, we sample 2048 points from the center points of the voxels, and for each sampled point, we select its 32 neighboring points within a radius of 4 m to calculate the biases to update its position and feature. When utilizing the relative position encoding strategy, the distance b is set as 16. In addition, the feature dimension of the self-attention is 64, and the interpolation radius is set as 1.6 m to select 16 downsampled points to propagate their features for the original points.

In the second stage, we divide the proposal into $6 \times 6 \times 6$ grids, and then set two Manhattan distance thresholds of 2 and 4 to select neighboring voxels for the grid center point during voxel RoI pooling. Then in the DIA module, the kernel size of the 3D convolution layer is $3 \times 3 \times 3$, and the stride is set as 1.

Training and Inference Details. For the KITTI dataset, our CIANet is trained on the GeForce RTX 3090 Ti GPU in an end-to-end manner for 80 epochs with batch size 4. For the Waymo dataset, we train our CIANet on the same device with batch size 4 for epochs 30. We adopt the ADAM optimizer and set the initial learning rate as 0.01, which is updated by the cosine annealing strategy. In the training phase, we adopt the Non-Maximum Suppression (NMS) method with a threshold of 0.8 to select 512 initial proposals. Then we employ the IoU threshold 0.55 to sample 128 proposals for classification and regression, where the positive and negative proposals have a ratio of 1:1. And the positive samples have IoU > 0.55 with respect to the ground truth box. For classification loss, we set the foreground IoU threshold μ_F as 0.75 and the background IoU threshold μ_B as 0.25. For regression loss, we set the box regression IoU threshold μ_{reg} as 0.55. In the testing phase, we remove the redundant proposals by NMS threshold 0.85 to retain the top 100 proposals for box refinement. After that, we further remove redundant detection boxes with the NMS threshold of 0.1 to obtain the final 3D detection boxes.

Moreover, to improve the generality of the model, we adopt several common data augmentation techniques. In detail, we select half of the scenes to flip along the X axis and rotate the scene around the Z axis with an angle scope of [−π/4, π/4], and scale the scene with a random scaling factor between 0.95 and 1.05. Also, we conduct the ground-truth sampling augmentation mechanism, which randomly pastes some new ground-truth objects from other scenes to the current scene. For each object category, we paste at least 15 ground-truth objects.

4.3 3D Detection on the KITTI Dataset

We train the CIANet model on the training set of the KITTI dataset and evaluate it on the validation and test sets. And then we compare the detection accuracy of CIANet with other state-of-the-art methods, as shown in Table 1 and Table 2. Furthermore, we evaluate the computational complexity of CIANet and compare it with other models on the KITTI validation set, as shown in Table 3.

Comparison with other methods. We display the detection results of our CIANet and other advanced methods on the validation set in Table 1, and it is clear that the AP values of our CIANet outperform other methods for pedestrians and cyclists that are hard to be detected. Compared with baseline network Voxel-RCNN, our method achieves AP improvements of 2.36%, 2.53%, and 1.65% for pedestrians at easy, moderate, and hard levels, and the AP gains of 2.89%, 1.67%, and 2.46% for cyclists at three difficulty levels. Notably, our method also surpasses Voxel-RCNN by 0.42%, 0.73%, and 0.79% AP gains at three difficulty levels for cars, respectively. Compared to the voxel-based method CT3D, which employs the Transformer structure composed of the encoder and decoder, our method achieves distinct AP improvements of 3.72%, 2.7%, and 1.15% for pedestrians and 2.08%, 2.07%, and 2.64% for cyclists at easy, moderate, and hard levels. In addition, the AP values for car detection are also improved by 0.57% and 0.56% at easy and hard levels than CT3D. Then, compared with the advanced detector VoTr, which adopts the Transformer to extract the enhanced features in a larger receptive field, our CIANet achieves significant AP gains for car category at three difficulty levels. In recent years, point-voxel combined methods have also achieved great detection performance. For the typical point-voxel-based network PV-RCNN, our CIANet has significant AP advantages at three categories of objects, especially the AP gains of 7.7% for pedestrians at moderate level and 6.19% for cyclists at hard level. Compared with the latest Octree-based Transformer network OcTr, our CIANet gains substantial advantages. In detail, CIANet gets AP gains by (1.25%, 6.32%, 2.16%) for cars, and (6.46%, 5.37%, 4.56%) for pedestrians, and (1.83%, 3.34%, 4.52%) for cyclists at three difficulty levels.

We further evaluate our CIANet and recent methods on the KITTI test set and display their detection results in Table 2. By comparing their AP values, it is obvious that our CIANet outperforms other methods for pedestrians and cyclists at moderate and hard levels, and it also ranks first for cars at the hard level.

Compared with the baseline network Voxel-RCNN, our method obtains significant AP improvement by 3.55%, 5.72%, and 4.11% for pedestrians at three difficulty levels, and

Table 1 Comparison of AP(%) values with different methods on KITTI validation set at easy, moderate, and hard levels for car, pedestrian, and cyclist categories, and all results are 11 recall positions (The bold values denote the best results)

Method	Reference	Car 3D (IoU = 0.7)			Pedestrian 3D (IoU = 0.5)			Cyclist 3D (IoU = 0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND [9]	Sensors 2018	88.61	78.62	77.22	56.55	52.98	47.73	80.59	67.16	63.11
PointPillars [10]	CVPR 2019	86.46	77.28	74.65	57.75	52.29	47.91	80.06	62.69	59.71
STD [16]	ICCV 2019	**89.70**	79.80	79.30	–	–	–	–	–	–
PV-RCNN [15]	CVPR 2020	89.35	83.69	78.70	63.12	54.84	51.78	86.06	69.48	64.50
Part-A^2 [11]	TPAMI 2020	89.56	79.41	78.84	65.69	60.05	55.45	85.50	69.90	65.49
CT3D [18]	ICCV 2021	89.11	85.04	78.76	64.23	59.84	55.76	85.04	71.71	68.05
VoTr [27]	ICCV 2021	87.86	78.27	76.93	–	–	–	–	–	–
RDIoU [31]	ECCV 2022	89.16	**85.24**	78.41	63.26	57.47	52.53	83.32	68.39	63.63
VoxSeT [32]	CVPR 2022	88.45	78.48	77.07	60.62	54.74	50.39	84.07	68.11	65.14
OcTr [33]	CVPR 2023	88.43	78.57	77.16	61.49	57.17	52.35	85.29	70.44	66.17
Voxel-RCNN (3classes)*[12]	AAAI 2021	89.26	84.16	78.53	65.59	60.01	55.26	84.23	72.11	68.23
CIANet (Ours)	–	89.68	84.89	**79.32**	**67.95**	**62.54**	**56.91**	**87.12**	**73.78**	**70.69**

Table 2 Comparison of AP(%) values with different methods on KITTI test set at easy, moderate, and hard levels for car, pedestrian, and cyclist categories, and all results are 40 recall positions (The bold values denote the best results)

Method	Reference	Car 3D (IoU = 0.7)			Pedestrian 3D (IoU = 0.5)			Cyclist 3D (IoU = 0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND [9]	Sensors 2018	84.65	75.96	68.71	45.31	35.52	33.14	75.83	60.82	53.67
PointPillars [10]	CVPR 2019	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
PointRCNN [5]	CVPR 2019	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
PV-RCNN [15]	CVPR 2020	90.25	81.43	76.82	52.17	43.29	40.29	78.60	63.71	57.65
TANet [24]	AAAI 2020	84.39	75.94	68.82	**53.72**	44.34	40.49	75.70	59.44	52.53
VIC-Net [34]	ICRA 2021	88.25	80.61	75.83	43.82	37.18	35.35	78.29	63.65	57.27
SE-SSD [35]	CVPR 2021	**91.49**	**82.54**	77.15	–	–	–	–	–	–
IASSD [36]	CVPR 2022	88.87	80.32	75.10	49.01	41.20	38.03	80.78	66.01	58.12
M3DETR [37]	WACV 2022	90.28	81.73	76.96	45.70	39.94	37.66	**83.83**	66.74	59.03
EPNet + + [38]	TPAMI 2022	91.37	81.96	76.71	52.79	44.38	41.29	76.15	59.71	53.67
Voxel-RCNN (3 classes)*[12]	AAAI 2021	90.62	81.44	76.91	44.23	39.11	37.39	83.17	67.00	59.33
CIANet (Ours)	–	90.97	81.95	**77.39**	47.78	**44.83**	**41.50**	82.89	**67.97**	**59.78**

Table 3 Comparison of the computational complexity for different methods on KITTI validation set, and all results are 11 recall positions

Method	Car 3D (IoU = 0.7)	Pedestrian 3D (IoU = 0.5)	Cyclist 3D (IoU = 0.5)	Param(M)	FLOPs(G)
	mAP	mAP	mAP		
SECOND [9]	81.48	52.42	70.29	4.62	76.82
PointPillars [10]	79.46	52.65	67.49	4.83	63.53
PointRCNN[16]	81.63	–	–	4.04	27.71
PV-RCNN [15]	83.91	56.58	73.35	12.41	91.98
Part-A^2 [11]	82.60	60.40	73.63	61.64	82.60
CT3D [18]	84.30	59.94	74.93	6.09	101.09
RDIoU [31]	84.27	57.75	71.78	4.82	79.41
Voxel-RCNN (3classes)* [12]	83.98	60.29	74.86	6.89	23.40
CIANet (Ours)	84.63	62.47	77.20	12.30	31.74

0.35%, 0.51%, and 0.48% AP gains for cars. Also, our method achieves 0.97% and 0.45% AP boosts for cyclists at moderate and hard levels, respectively. For the attention-based detection network TANet which specializes in detecting pedestrians, our CIANet improves the AP values for pedestrians by 0.49% at the moderate level and 1.01% at the hard level. The IASSD is a new voxel-based detection network, and our CIANet exceeds it by 3.63% for pedestrians on the moderate level and 3.47% on the hard level. Meanwhile, the AP values of our network outperform IASSD by (2.1%, 1.63%, 2.29%) for cars and (2.11%, 1.96%, 1.66%) for cyclists at three difficulty levels. Compared to the network EPNet + + , which fuses point cloud and image information for object detection, our CIANet still achieves significant AP advantages by (6.74%, 8.26%, 6.11%) for cyclists at easy, moderate, and hard levels, and (0.45%, 0.21%) for pedestrians at moderate and hard levels.

Overall, the detection results on the validation and test sets demonstrate that our method achieves superior detection performance than other state-of-the-art methods, especially for pedestrians and cyclists. This is attributed to the attention mechanisms in our CIANet can compute the spatial contextual correlation among different parts of objects, thus guiding the network focus on the parts of the object that lack points to depict the boundary. That is, our CIANet can enhance the feature representation of weak objects with inadequate boundaries, thus it effectively boosts the detection accuracy of pedestrians and cyclists with small sizes.

Analysis of computational complexity. To evaluate the computational complexity of CIANet, we compare the number of Parameters (Params) and the number of FLOating-Point operations (FLOPs) of different methods under the same mAP metric in Table 3. It can be seen that our CIANet consumes comparable Params and FLOPs compared to the baseline Voxel-RCNN, but it significantly outperforms Voxel-RCNN in terms of mean detection precision for the three object categories. Moreover, in comparison to the Params and FLOPs of all methods, we observe that Voxel-RCNN has the lowest FLOPs and fewer Params, *i.e.*, the baseline Voxel-RCNN has the highest computational efficiency. Our CIANet inherits the advantage of moderate parameters and high computational efficiency of this baseline network, and at the same time achieves the highest mAP values among all methods. This demonstrates that our

CIANet can effectively improve the detection accuracy while maintaining low computational complexity.

4.4 3D Detection on the Waymo Dataset

We train the CIANet model on the training sequence of the Waymo dataset and evaluate its detection performance for vehicles, pedestrians, and cyclists on the validation sequence. Subsequently, we compare the detection accuracy of CIANet with some recent methods, as shown in Table 4.

From the detection results on the Waymo validation sequence in Table 4, it is noted that CIANet has the highest detection accuracy for vehicles, pedestrians, and cyclists over other typical detection networks. Compared with the voxel-based network Part-A^2, which has a convincing performance on the Waymo dataset, our CIANet obtains significant mAP and mAPH gains at different difficulty levels, especially for mAPH of pedestrians with gains of 1.84% and 1.36% at LEVEL_1 and LEVEL_2, and mAP of cyclists with improvements of 1.45% at LEVEL_1. The mAP and mAPH values of LEVEL_1 vehicles also significantly surpass Part-A^2 by 1.58% and 1.44%, respectively. Then compared to the point-voxel-based network PV-RCNN, our CIANet achieves remarkable mAP and mAPH advantages for the detection of vehicles, pedestrians, and cyclists. Concretely, the mAP values of our network outperform PV-RCNN by (1.12%, 1.44%, 2.24%) at LEVEL_1, respectively, and (0.67%, 1.31%, 1.53%) at LEVEL_2. And our mAPH values outperform PV-RCNN by (1.06%, 3.06%, 2.03%) at

Table 4 Comparison of the detection performance for different methods on the Waymo validation sequence (The bold values denote the best results)

Difficulty level	Method	ALL	Vehicle 3D (IoU = 0.7)		Pedestrian 3D (IoU = 0.5)		Cyclist 3D (IoU = 0.5)	
		mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
LEVEL_1	SECOND [9]	63.05	72.72	71.69	68.70	58.18	60.62	59.28
	PointPillars [10]	63.33	71.60	71.00	70.60	56.70	64.40	62.30
	PV-RCNN [15]	69.63	77.51	76.89	75.01	65.65	67.81	66.35
	Part-A^2 [11]	70.25	77.05	76.51	75.24	66.87	68.60	67.36
	LiDAR RCNN [39]	66.20	73.50	73.00	71.20	58.70	68.60	66.90
	3D-MAN [40]	-	69.03	68.52	71.71	67.74	-	-
	IASSD [36]	64.48	70.53	69.67	69.38	58.47	67.67	65.30
	CIANet (Ours)	**71.68**	**78.63**	**77.95**	**76.45**	**68.71**	**70.05**	**68.38**
LEVEL_2	SECOND [9]	57.23	63.85	63.33	60.72	51.31	58.34	57.05
	PointPillars [10]	57.53	63.10	62.50	62.90	50.20	61.90	59.90
	PV-RCNN [15]	63.33	68.98	68.41	66.04	57.61	65.39	63.98
	Part-A^2 [11]	63.84	68.47	67.97	66.18	58.62	66.13	64.93
	LiDAR RCNN [39]	60.10	64.70	64.20	63.10	51.70	66.10	64.40
	3D-MAN [40]	-	60.16	59.71	62.58	59.04	-	-
	IASSD [36]	58.08	61.55	60.80	60.30	50.73	64.98	62.71
	CIANet (Ours)	**65.00**	**69.65**	**69.23**	**67.35**	**59.98**	**66.92**	**65.80**

LEVEL_1, and (0.82%, 2.37%, 1.82%) at LEVEL_2. Moreover, our CIANet achieves substantial accuracy improvements than the recent IASSD for vehicles, pedestrians, and cyclists at two levels. In addition, from the 3rd column of Table 4, it can be seen that our CIANet's average mAPH value among the three categories surpasses other methods, which effectively proves that the overall detection performance of our CIANet on the Waymo dataset exceeds other detectors. This indicates that our proposed attention modules promote the network to perceive the sparse boundary features of objects in the large-scene dataset, thus significantly boosting the detection accuracy.

4.5 Ablation Studies

We conduct ablation experiments to validate the effectiveness of our designed attention modules in the CIANet. All ablation experiments are performed on the KITTI validation set with 11 recall locations and the mAP value is adopted as the evaluation criterion.

4.5.1 Ablation studies of attention modules

In this section, we first report the detection results of baseline Voxel-RCNN in the 1st row of Table 5 for comparison. Then we add the CSHA module upon the baseline to highlight the vital information in the channel and spatial domains of BEV features, as shown in the 2nd row, leading to the mAP gains of 0.2%, 0.65%, and 0.64% for cars, pedestrians, and cyclists. This proves that the CSHA module can effectively heighten BEV features to improve detection performance. Subsequently, we further add the CSA module to extract the global context information and supply it to the enhanced BEV features, as shown in the 3rd row. The 2nd and 3rd rows show that the CSA module boosts the mAP values with gains of 0.23%, 0.8%, and 0.79% for three categories of objects, which illustrates that the global contextual correlations among different parts of objects captured by the CSA module facilitate the detection. Finally, we further append the DIA module to capture the interactions between channel and spatial dimensions of the RoI feature, as shown in the 4th row. From the 3rd and 4th rows, we observe that the mAP values for three categories of objects are improved by 0.22%, 0.73%, and 0.91%, respectively. This proves that it is necessary for detection to further enhance RoI features with the attention mechanism.

Table 5 The results of ablation experiments for CIANet on KITTI validation set, here we report the mAP(%) with 11 recall position

Base	CSHA	CSA	DIA	Car 3D (IoU = 0.7)	Ped. 3D (IoU = 0.5)	Cyc.3D (IoU = 0.5)
				mAP	mAP	mAP
✓				83.98	60.29	74.86
✓	✓			84.18	60.94	75.50
✓	✓	✓		84.41	61.74	76.29
✓	✓	✓	✓	84.63	62.47	77.20

Table 6 The results of ablation experiments for CSHA module on KITTI validation set

Ablation protocols	Car 3D (IoU = 0.7)	Ped. 3D (IoU = 0.5)	Cyc.3D (IoU = 0.5)
	mAP	mAP	mAP
CSHA-N	84.42	61.72	76.54
CSHA-N + CBAM	84.49	61.89	76.87
CSHA-N + CD	84.58	62.15	76.89
CSHA-N + SD	84.60	62.36	77.07
CSHA-N + CD + SD	84.63	62.47	77.20

Overall, successively adding the above attention modules to the baseline can gradually improve the mAP values of the three object categories, and in particular, the mAP improvements of pedestrians and cyclists are more significant. This validates the effectiveness of our method based on elaborate attention modules.

4.5.2 Ablation studies based on the CSHA module

To further explore the effectiveness of the CSHA attention module, we conduct ablation experiments as shown in Table 6. Here, we first remove the CSHA module from CIANet to obtain a new model named CSHA-N, and its detection accuracies for three object categories are shown in 1st row. Then we add the classical attention-based module CBAM with serial channels and spatial attention branches to the CSHA-N model, as shown in the 2nd row. From the 1st and 2nd rows, we observe that adding the CBAM model alone only slightly promotes the mAP gains. Then we add the Channel Domain (CD) branch and Spatial Domain (SD) branch of the CSHA module to the CSHA-N model in turn, as denoted in the 3rd and 4th rows. From the 1st and 3rd rows, we find that the CD branch brings out mAP improvements for cars, pedestrians, and cyclists, which verifies that modeling the interdependence between feature channels by the CD branch can improve detection accuracy. From the 1st and 4th rows, it can be seen that the addition of the SD branch boosts the mAP values. This certifies that the SD branch can highlight crucial spatial information of the object to facilitate the detection. Lastly, we apply both the CD and SD branches to the CSHA-N baseline, as shown in the 5th row, which achieves mAP improvements of 0.21%, 0.75%, and 0.66% compared to CSHA-N. From the 3rd to 5th rows, it is noted that combining two branches contributes more to detection performance than utilizing one branch alone. And from the 2nd and 5th rows, we observe that our CSHA module with parallel channels and spatial attention branches achieves better detection performance than the traditional attention module CBAM. This demonstrates that the parallel manner is more efficient than the traditional serial manner to splice the channel and spatial branches.

4.5.3 Ablation studies based on the CSA module

We employ the CSA module which employs the position bias operation and the self-attention mechanism to extract global contextual information of objects. To verify the effect of the position bias operation, we conduct ablation experiments as shown in Table 7.

Table 7 The results of ablation experiments for position bias operation in CSA module on KITTI validation set

Ablation protocols	Car 3D (IoU = 0.7)	Ped. 3D (IoU = 0.5)	Cyc.3D (IoU = 0.5)
	mAP	mAP	mAP
CSA-N	84.50	61.78	76.50
CSA-N + position bias	84.63	62.47	77.20

Table 8 The results of ablation experiments for DIA module on KITTI validation set

Br1	Br2	Br3	Br4	Car 3D (IoU = 0.7)	Ped. 3D (IoU = 0.5)	Cyc.3D (IoU = 0.5)
				mAP	mAP	mAP
				84.41	61.74	76.29
✓				84.47	61.94	76.54
✓	✓			84.54	62.13	76.80
✓	✓	✓		84.59	62.31	77.05
✓	✓	✓	✓	84.63	62.47	77.20

Concretely, we remove the position bias operation from CIANet, as shown in the 1st row, resulting in the decrease of the mAP values by 0.13%, 0.69%, and 0.7% respectively. This drop verifies that the position bias operation updating the position information of the sampled points is conducive to detection.

4.5.4 Ablation studies based on the DIA module

To explore the contribution of the internal mechanism of the DIA module to the detection performance, we conduct ablation experiments on its four branches (Br1, Br2, Br3, Br4), respectively, and the experimental results are shown in Table 8. We first remove the DIA module from CIANet to form a detection model named DIA-N, and its detection results are displayed in the 1st row.

Then we introduce the first branch (Br1) upon DIA-N to explore the interactions between the channel dimension C and spatial dimensions W, H of RoI features, as shown in the 2nd row, resulting in mAP gains of 0.06%, 0.2%, and 0.25% for cars, pedestrians, and cyclists. Afterwards, as denoted in the 3rd row, we further add the second branch (Br2) to learn the interactions between the channel dimension C and spatial dimensions L, H of RoI features. From the 2nd and 3rd rows, it can be seen that mAP values are improved by 0.07%, 0.19%, and 0.26% for the three categories of objects. Similarly, as illustrated in the 4th row, we apply the third branch (Br3) to explore the correlations between the channel dimension C and spatial dimensions L, W. And from the 3rd and 4th rows, we can see that the mAP values are boosted by 0.05%, 0.18%, and 0.25%, respectively. At last, as shown in the 5th row, we add the fourth branch (Br4) to capture the dependencies among spatial dimensions L, W, and H, which further increases the mAP values by 0.04%, 0.16%, and 0.15%, respectively. And from the 1st to 5th rows, we observe that the detection accuracy of the network is

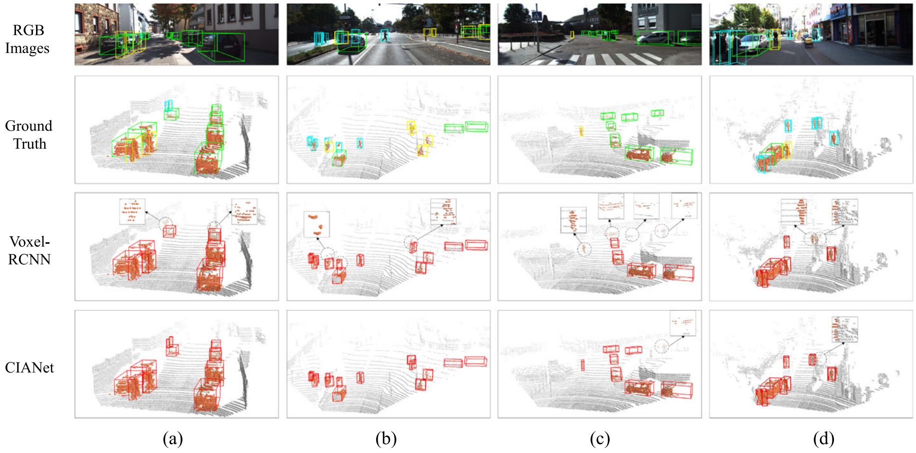

Fig. 5 Qualitative results on different scenes of the KITTI dataset. The first row displays the RGB images with 3D ground truth boxes, and the second row describes the corresponding point cloud scenes with 3D ground truth boxes, where cars, pedestrians and cyclists are labeled as green, blue and yellow, respectively. The third and fourth rows display the 3D object prediction boxes of the baseline network Voxel-RCNN and our CIANet, where all the prediction boxes are labeled as red. All the foreground objects in the above point cloud scenes are colored as orange

gradually increased with the sequential addition of the above four branches. This indicates that capturing the interaction information among different dimensions of RoI features can enhance the feature representation of the object, thus promoting detection accuracy.

4.6 Visualization of the Results

We visualize in Fig. 5 the detection results of baseline Voxel-RCNN and our CIANet on KITTI validation set. For scene a) of Fig. 5, Voxel-RCNN misses two pedestrians and one car with extremely sparse points, but our CIANet accurately detects them. For scene b), we observe that our CIANet detects two occluded pedestrians, but one of them is ignored by the Voxel-RCNN. Meanwhile, CIANet precisely detects one distant cyclist with sparse points, which is also missed by Voxel-RCNN. In scene c), Voxel-RCNN misses three long-distance cars and one small cyclist, while CIANet only misses one extremely distant car. Similarly, in scene d), Voxel-RCNN fails to detect two distant pedestrian objects, while CIANet only omits the one located on the right. These intuitive visualization results further demonstrate that our method can effectively detect objects in complex scenes, especially for weak small objects with sparse points. This means that our elaborate attention modules indeed facilitate the improvement of detection performance. Besides, we also find that although CIANet achieves better detection results than Voxel-RCNN in scene c) and scene d), it is unable to detect some extremely distant objects. Actually, the voxel-based backbone network of CIANet adopts the voxel centroids to represent the voxels of objects. And for extremely distant objects with too sparse boundaries, the centers of the voxels that correspond to their boundaries tend to fall outside the ground truth boxes, which causes these boundary voxels to be misclassified as background voxels, and thus the extremely distant objects are missed.

5 Conclusion

In this paper, we present a novel two-stage object detection network CIANet based on elaborate attention modules. In the first stage, we first explore the channel interdependence and crucial spatial information of the BEV feature, followed by a 2D CNN to obtain enhanced BEV features. Then we replenish the enhanced BEV features with the spatial contextual associations, which are captured from different parts of objects. In this way, we highlight the sparse boundary parts of weak objects to generate high-quality proposals. Finally, in the second stage, we further capture the interactions between the channel and spatial dimensions of RoI features to focus on the prominent voxel grid features, and then apply the enhanced RoI features for proposal refinement, thus generating accurate detection boxes. Extensive experiments on the KITTI and Waymo datasets validate that CIANet achieves significant improvement in detection performance over existing methods, especially for weak objects with small sizes like pedestrians and cyclists.

In addition, this work still has the following shortcoming that needs to be addressed. In the voxel-based backbone network of CIANet, we utilize the voxel centroids to represent the voxels of the object, which makes the centroids of the boundary voxels of some objects, especially the weak objects with extremely sparse boundaries, tend to fall outside the ground truth boxes and are misclassified as background voxels. Hence, in future work, we intend to add a point-wise auxiliary branch to the backbone network, and it captures the native point-wise features that contain complete boundary information of the object and supplements them to voxel features, thus leading the network to more accurately perceive the boundaries of objects and further improve detection accuracy.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 62106086), the Natural Science Foundation of Hubei Province (No. 2021CFB564), and the Research Fund of Jianghan University (No. 2021yb060).

Author contributions Jing Zhou (First Author and Corresponding Author): Conceptualization, Methodology, Software, Investigation, Formal Analysis, Funding Acquisition, Writing - Original Draft; Zixin Gong: Data Curation, Visualization, Writing - Original Draft; Junchi Zhang: Validation, Resources, Supervision.

Declarations

Conflict of interest The authors declare no conflict of interest.

Data Availability All data generated or analyzed during this study are included in this article. The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. Liu B, Tian B, Wang H, Qiao J, Wang Z (2022) Fusenet: 3d object detection network with fused information for lidar point clouds. Neural Process Lett 54(6):5063–5078

2. Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1907–1915
3. Yang B, Luo W, Urtasun R (2018) Pixor: real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7652–7660
4. Zhou Y, Sun P, Zhang Y, Anguelov D, Gao J, Ouyang T, Guo J, Ngiam J, Vasudevan V (2020) End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: Conference on robot learning. PMLR, pp 923–932
5. Shi S, Wang X, Li H (2019) Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 770–779
6. Yang Z, Sun Y, Liu S, Jia J (2020) 3dssd: point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11040–11048
7. Shi W, Rajkumar R (2020) 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, pp 1711–1719
8. Li J, Luo S, Zhu Z, Dai H, Krylov AS, Ding Y, Shao L (2020) 3d iou-net: Iou guided 3d object detector for point clouds. arXiv:2004.04962
9. Yan Y, Mao Y, Li B (2018) Second: sparsely embedded convolutional detection. Sensors 18(10):3337
10. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) Pointpillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12697–12705
11. Shi S, Wang Z, Shi J, Wang X, Li H (2020) From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE Trans Pattern Anal Mach Intell 43(8):2647–2664
12. Deng J, Shi S, Li P, Zhou W, Zhang Y, Li H (2021) Voxel r-cnn: towards high performance voxel-based 3d object detection. Proc AAAI Conf Artif Intell 35(2):1201–1209
13. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 652–660
14. Qi CR, Yi L, Su H, Guibas LJ (2017) “Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems, vol 30
15. Shi S, Guo C, Jiang L, Wang Z, Shi J, Wang X, Li H (2020) Pv-rcnn: point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10529–10538
16. Yang Z, Sun Y, Liu S, Shen X, Jia J (2019) Std: sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1951–1960
17. Noh J, Lee S, Ham (2021) Hvpr: hybrid voxel-point representation for single-stage 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14605–14614
18. Sheng H, Cai S, Liu Y, Deng B, Huang J, Hua X-S, Zhao M-J (2021) Improving 3d object detection with channel-wise transformer. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2743–2752
19. Rouhafzay G, Cretu A-M, Payeur P (2023) A deep model of visual attention for saliency detection on 3d objects. In: Neural processing letters, pp 1–21
20. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
21. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
22. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) Eca-net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11534–11542
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
24. Liu Z, Zhao X, Huang T, Hu R, Zhou Y, Bai X (2020) Tanet: robust 3d object detection from point clouds with triple attention. Proc AAAI Conf Artif Intell 34(07):11677–11684
25. Bhattacharyya P, Huang C, Czarnecki K (2021) Sa-det3d: self-attention based context-aware 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3022–3031
26. Pan X, Xia Z, Song S, Li Le, Huang G (2021) 3d object detection with pointformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7463–7472
27. Mao J, Xue Y, Niu M, Bai H, Feng J, Liang X, Xu H, Xu C (2021) Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3164–3173
28. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28

29. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
30. Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, et al. (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2446–2454
31. Sheng H, Cai S, Zhao N, Deng B, Huang J, Hua X-S, Zhao M-J, Lee GH (2022) Rethinking iou-based optimization for single-stage 3d object detection. In: European conference on computer vision. Springer, pp 544–561
32. He C, Li R, Li S, Zhang L (2022) 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, pp 8417–8427
33. Zhou C, Zhang Y, Chen J, Huang D (2023) Octr: octree-based transformer for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5166–5175
34. Jiang T, Song N, Liu H, Yin R, Gong Y, Yao J (2021) Vic-net: voxelization information compensation network for point cloud 3d object detection. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp 13408–13414
35. Zheng W, Tang W, Jiang L, Fu C-W (2021) Se-ssd: self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14494–14503
36. Zhang Y, Hu Q, Xu G, Ma Y, Wan J, Guo Y (2022) Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18953–18962
37. Guan T, Wang J, Lan S, Chandra R, Wu Z, Davis L, Manocha D (2022) M3detr: multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 772–782
38. Liu Z, Huang T, Li B, Chen X, Wang X, Bai X (2022) Epnet++: cascade bi-directional fusion for multi-modal 3d object detection. In: IEEE transactions on pattern analysis and machine intelligence (2022)
39. Li Z, Wang F, Wang N (2021) Lidar r-cnn: an efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7546–7555
40. Yang Z, Zhou Y, Chen Z, Ngiam J (2021) 3d-man: 3d multi-frame attention network for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1863–1872