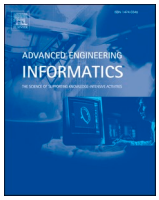
Advanced Engineering Informatics 65 (2025) 103370

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Alternating interaction fusion of Image-Point cloud for Multi-Modal 3D object detection

Guofa Li [a], Haifeng Lu [b], Jie Li [a,*], Zhenning Li [c], Qingkun Li [d], Xiangyun Ren [e], Ling Zheng [a]

[a] College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China
[b] Institute of Human Factors and Ergonomics, College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
[c] State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau 999078, China
[d] Beijing Key Laboratory of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[e] State Key Laboratory of Intelligent Vehicle Safety Technology, Chongqing Changan Automobile Co., Ltd, Chongqing 400023, China

ARTICLE INFO

Keywords:
3D object detection
Camera-LiDAR fusion
Autonomous Vehicles

ABSTRACT

A mainstream feature fusion method involves enhancing Lidar point cloud information by incorporating camera, but it fails to fully utilize the rich information in images. Another method uses a dual-channel parallel approach to fuse image and point cloud information, but it also faces issues such as excessive module stacking and high computational demands. Therefore, we propose a powerful alternating interaction fusion approach. Firstly, it resolves the problem of unilateral fusion schemes that overly rely on point cloud information and fail to fully utilize image data. Secondly, it tackles the problem of excessive module stacking and high computational demands in dual-channel parallel fusion schemes of point cloud and image data. Specifically, our alternate interactive fusion module implements a method where image and point cloud BEV features mutually enhance each other. Local attention interactions are engaged between image features containing point cloud information and regular image features. This enhances the expressiveness of image features. Subsequently, internal BEV attention interactions occur between point cloud BEV features with enriched image information and regular point cloud BEV features. This step improves the expressiveness of the point cloud BEV features. Experiments on the large-scale nuScenes dataset demonstrate that our proposed method outperforms both the unilateral point cloud-centric fusion and the parallel interactive fusion approaches.

1. Introduction

3D object detection is pivotal for autonomous driving, achieving automation by locating and recognizing objects in the real world. LiDAR and cameras serve as complementary sensors for 3D object detection. LiDAR provides high-precision spatial information with excellent resolution of object shapes and edges [1], while camera images contain color, texture, and other features crucial for object identification and scene comprehension [2]. Different forms of data offer complementary knowledge to achieve cross-modal information fusion, thereby enhancing the detection performance [3,4].

Currently, the methods for fusing LiDAR and camera data generally follow three approaches [5–7]. The first approach is input-level fusion based on mapping alignment, with PointPainting being a representative method. This method directly incorporates image features into LiDAR point clouds before the features enter the network, leading to the premature loss of important image feature information. Specifically, due to the difference in sparsity between image pixels and point clouds, most pixel feature points cannot be effectively mapped to the sparse point cloud points, resulting in the underutilization of rich image feature information. Additionally, due to the heterogeneity of features, images provide dense semantic information, while point clouds offer sparse 3D spatial information. Directly attaching image features to point clouds fails to fully leverage the semantic advantages of image features.

The second is feature-level fusion based on mapping alignment, with MV3D [9] and BEVFusion [10] as typical methods. The intrinsic fusion challenges are similar to those of input-level fusion, mainly involving reliance on precise calibration, modality heterogeneity, and alignment errors [11]. Firstly, the feature alignment relies heavily on calibrated feature mapping, making it highly susceptible to calibration errors. Secondly, due to the heterogeneity of the features, aligning sparse 3D point clouds with dense 2D image features is a significant challenge.

* Corresponding author.
E-mail address: jieli@cqu.edu.cn (J. Li).

https://doi.org/10.1016/j.aei.2025.103370
Received 16 December 2024; Received in revised form 19 March 2025; Accepted 14 April 2025
Available online 24 April 2025

Particularly during hard-association, the sparsity of LiDAR point clouds may result in the inability to fully utilize the rich contextual information provided by camera images, causing the fused features to lose critical details.

The third is fusion based on attention mechanisms, using the self-attention and cross-attention mechanisms of transformer to adaptively fuse data or features between the two modalities [12]. This soft-association mechanism can address many issues present in mapping alignment, such as calibration errors, but it also has its inherent limitations. FUTR3D [25] and TransFusion [13] are representative methods. FUTR3D [25] directly interacts with image and point cloud features in the 3D object query box without fusing the features of both modalities. While this simplifies the fusion process, it lacks dedicated modules to achieve modality fusion at the feature level, making it difficult to fully utilize the complementary information from both images and point clouds. As LiDAR-based fusion methods have consistently shown better detection results, unilateral fusion strategies like TransFusion [13] and DeepFusion [14] continue to adopt this approach, aiming to fully utilize LiDAR's advantages in spatial perception and distance measurement. However, this unilateral fusion strategy overly relies on LiDAR, with image features only added as auxiliary information, failing to fully exploit the advantages of images in semantic understanding. The fine-grained semantic information from images, like color and texture, is underutilized in the fusion process. This can cause inaccuracies in detection, especially in complex scenes where LiDAR's geometric data alone is not enough. Therefore, the potential of image features is not fully realized, negatively affecting the overall fusion performance.

DeepInteraction [15] proposes a parallel interaction method to fuse features while preserving the independence of the two modalities, interacting their features in the process. However, it also faces high computational demands and module stacking issues due to parallel and cascading prediction.

To address these shortcomings in fusion methods, this paper proposes an improved strategy of alternating interaction. Our key idea is to first enhance the features of one modality and then use these enhanced features to further improve those of the other modality. There are two conceptual approaches. One is to enhance image features first and then point cloud BEV features. The other is to enhance point cloud BEV features first and then image features. These approaches maintain the independence of the two types of features while leveraging their complementarity, thus improving feature expressiveness.

Specifically, we fuse the two modalities in an alternating interactive manner. Compared with the issue of insufficient interaction in unilateral fusion dominated by point clouds, this method initially focuses on images, fully utilizing image information. Compared with the computational burden of parallel fusion, this method enhances point cloud BEV features on the basis of the initially enhanced image features. The two stages interact in a cascading manner. This interaction extracts deeper features of the two modalities while using the same amount of interaction modules. It also reduces transformer interaction modules by half at the same module depth. This reduction contributes to the reduction of computational costs.

In summary, our contributions are twofold. Firstly, we explore the intrinsic challenges of LiDAR-camera fusion, highlighting the issues of insufficient utilization of image information and module stacking in multimodal fusion, which are the drawbacks of unilateral and parallel fusion mechanisms, respectively. Secondly, we propose a 3D object detection fusion model based on Transformer cross-attention, which alternately interacts and fuses features from the two modalities. Specifically, this method adopts a sequential fusion strategy, first optimizing image features and then optimizing point cloud features, thereby enhancing the expressiveness of the fused features while reducing the stacking of attention modules and lowering computational costs. Compared to the baseline method, our approach improves accuracy by + 1.91 % mAP and + 1.07 % NDS, while increasing inference speed from 2.99 FPS to 4.12 FPS.

The remainder of this paper is organized as follows: we review the related literature in Section 2, and introduce the proposed alternating interaction method in Section 3; experimental analysis and conclusions are presented in Section 4 and Section 5, respectively.

2. Related work

2.1. Camera-based 3D object detection methods

Due to the high cost of LiDAR sensors, researchers have devoted considerable effort to pure camera-based 3D perception. Previous methods mostly rely on single-view depth estimation [16], predicting image depth directly based on image features [17–19] or using intermediate feature representations [20,21] to estimate the 3D positions of objects. Multi-scale object detection based on disparity segmentation reduces redundant information transmission across scales [22]. For multi-view inputs, one approach leverages 3D space as an intermediary, with the assistance of a depth prediction network, LSS [23], to explicitly estimate the depth information of images, then transforms these 3D features from voxel space to BEV (Bird's Eye View) space, culminating in the construction of BEV features. However, the accuracy of predicted depth maps is significantly lower than that of LiDAR. This introduces semantic ambiguity into the BEV space. Another approach attempts to implicitly capture spatial or temporal information from multi-view images [24]. Typically, this method creates 3D object entities under BEV features and optimizes these entities using BEV features [25,26]. However, this method deprives 3D object entities of the opportunity to interact directly with the feature space. Additionally, camera-based methods are also susceptible to changes in lighting conditions and may perform poorly in low-light situations.

2.2. Lidar-based 3D object detection methods

Due to the unordered and irregular nature of point clouds, many 3D object detection methods handle them in a specific way. Firstly, point clouds are projected onto regular grids, such as 3D voxels [27,28], pillars [29], or range images [30]. Secondly, standard 2D or 3D convolutions are applied to compute features on the BEV plane. This process generates 3D boxes with different representations. Outdoor scene 3D detection models mostly utilize transformers for feature extraction [31,32]. However, the attention operation of each transformer layer has a computational complexity of $O(N^2)$ for N points, which is computationally intensive. Therefore, various methods [33,34] have been proposed to accelerate computation by applying Transformers to point cloud feature extraction. Meanwhile, schemes that use Transformer decoders or their variants as their detection heads [35] are increasingly adopted by researchers. 3DETR [36] employs a full Transformer decoder architecture with fewer design priors, simplifying the detection process and effectively eliminating the need for many manually designed components (e.g., NMS or anchor generation). However, point cloud-based methods have several drawbacks. These include susceptibility to weather conditions, complex data processing, and limited resolution.

2.3. Fusion based 3D object detection methods

In recent years, multi-sensor fusion technology has gained increasing attention in the field of 3D detection. Many autonomous vehicles with 3D object detection functions are typically equipped with LiDAR and multiple surround-view cameras. The existing 3D detection methods typically perform multi-modal fusion at one of three stages: input-level fusion, feature-level fusion and attention mechanisms fusion.

Input-level fusion methods mainly involve mapping image semantic features onto foreground LiDAR points and performing LiDAR-based detection on the augmented point cloud inputs. PointPainting [8] and PointAugmenting [37] are representative methods of input fusion. PointPainting [8] and Painted-PointRCNN [11] enhance point cloud

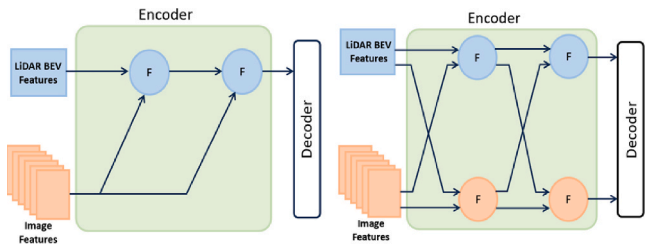


Fig. 1. The left subfigure is a unilateral fusion feature extraction module that primarily utilizes the point cloud channel, and the right subfigure is a parallel fusion module conducting feature optimization simultaneously through the point cloud and image channels.

data by associating each LiDAR point with corresponding image semantic labels generated by an image segmentation network. PointAugmenting [37] further improves this method by using pointwise CNN features extracted from a pre-trained 2D detection model to augment the point cloud. This approach significantly improves detection performance. The CNN features of the detection network are better adapted to the appearance variations of the targets compared with the highly abstract semantic segmentation scores used in PointPainting [8]. Besides, SFEMOS combines both moving object segmentation and scene flow estimation tasks and leverages geometric constraints to enhance their performance [38].

Feature-level fusion methods, which can also be applied in LiDAR-map feature matching [39], usually conduct multi-modal feature fusion at the backbone, candidate box generation or ROI refinement stages. These methods extract features from both image and LiDAR backbone networks and fuse them either at the proposal generation or ROI head stage. MV3D [9] and AVOD [40] are basic approaches in this domain. Alternatively, some approaches initiate by generating 3D object proposal boxes, then project these boxes onto image views and bird's-eye views to extract features and optimize the target boxes.

Attention mechanism-based fusion methods use Transformer decoders for multi-modal feature fusion [41–43]. FUTR3D [25] and TransFusion [13] define object queries in 3D space and integrate image features into these target boxes. Due to the advantages of point clouds in distance and spatial perception, these methods adopt a unilateral fusion strategy that is biased towards the 3D LiDAR modality, but this excessively neglects the role of images in fusion, failing to exploit the complementarity of multi-modal fusion. Concurrent methods like DeepInteraction [15] explores the drawbacks of such unilateral fusion strategies, proposing interacting features of the two modalities while preserving their unique feature representations. However, running both channels in parallel also leads to bulky modules and high computational costs. Fig. 1 shows the simplified structure of the feature fusion modules for unilateral fusion and parallel fusion.

The methods mentioned above either excessively neglect the role of images in the fusion process or suffer from module stacking and high computational consumption. In this study, we address these problems using a novel multi-modal interaction strategy. The key insight of our method is to alternately interact between modalities, fully exploring and utilizing image information while maintaining modal independence. This approach compensates for the shortcomings of a unilateral fusion strategy biased towards the 3D LiDAR modality, while keeping the modules streamlined to avoid unnecessary computational burden due to excessive stacking of modalities.

3. Method

In this section, we propose an Alternating Interaction Method (AIM) for 3D object detection using LiDAR and camera, with the overall structure of the model shown in Fig. 2. The AIM consists of two modules: Image Feature Enhancement Module (IFEM) and LiDAR Feature Enhancement Module (LFEM). Both modules include a cross-modal feature mapping and sampling module to achieve feature alignment, as well as a feature interaction module based on attention mechanisms

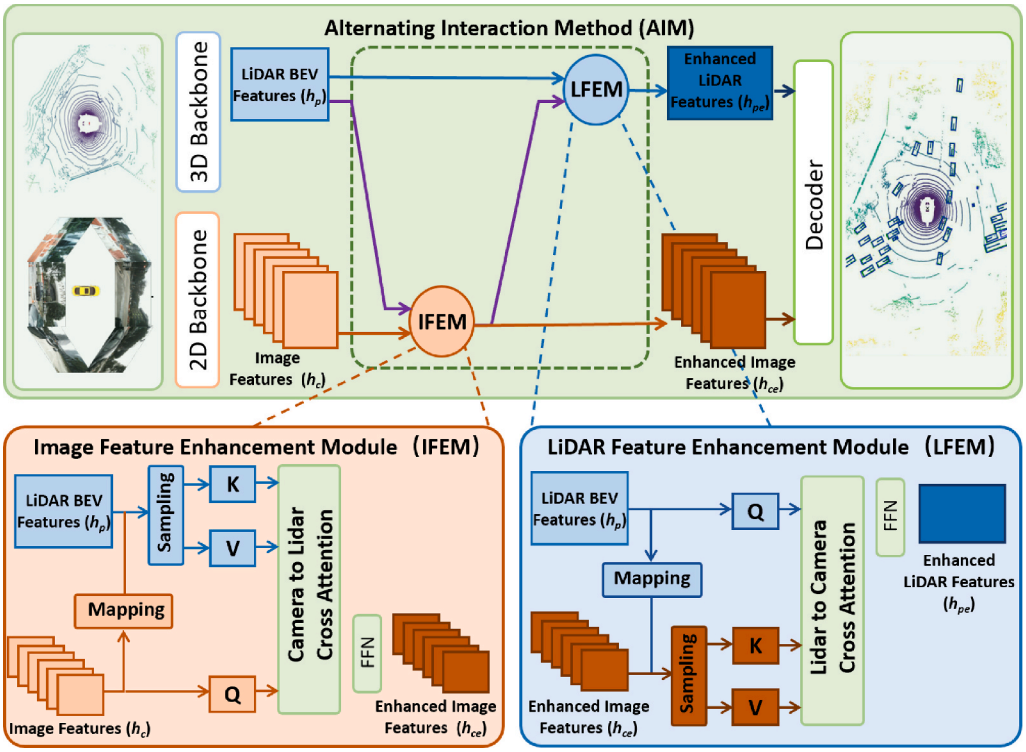


Fig. 2. Overall structure diagram of the proposed alternating interaction method. The 3D backbone and 2D backbone are utilized separately for feature extraction from LiDAR and images. Then, image and point cloud features are optimized through Image Feature Enhancement Module and LiDAR Feature Enhancement Module, respectively. Finally, the decoder outputs the result of multi-modal feature fusion.

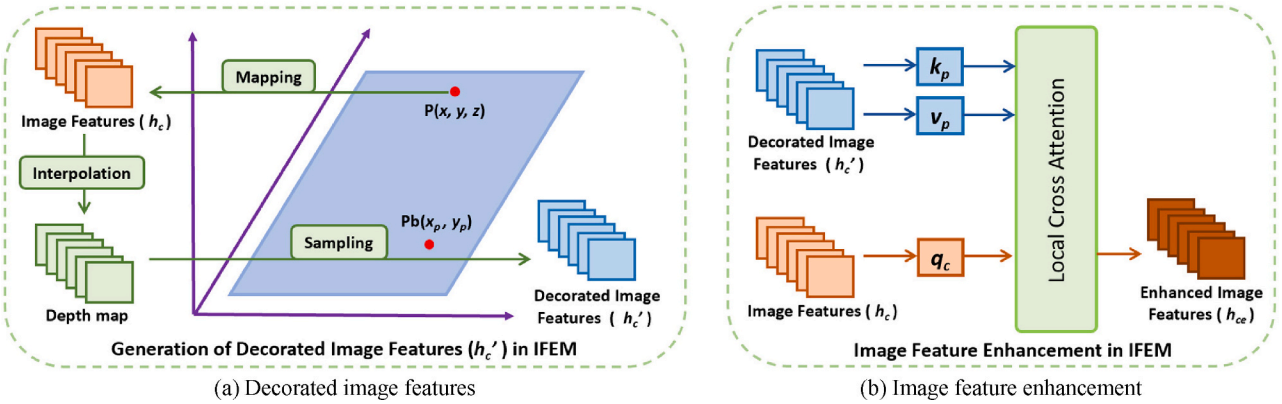


Fig. 3. Image Feature Enhancement Module (IFEM). (a) The process of constructing decorated image features $\boldsymbol{h_c}'$ through mapping and sampling. (b) The process of local cross-attention interaction between image features $\boldsymbol{h_c}$ and decorated image features $\boldsymbol{h_c}'$.

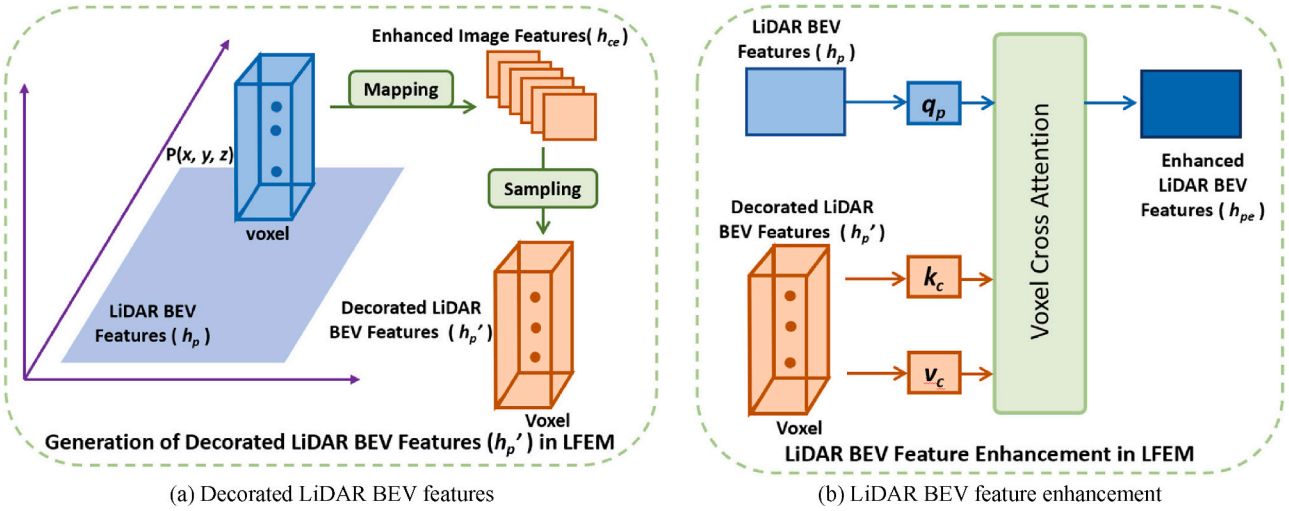


Fig. 4. LiDAR Feature Enhancement Module (LFEM). (a) The process of constructing decorated point cloud BEV features $\boldsymbol{h_p}'$ through mapping and sampling. (b) The process of voxel cross-attention interaction between point cloud BEV features $\boldsymbol{h_p}$ and decorated point cloud BEV features $\boldsymbol{h_p}'$.

to perform cross-modal feature interaction and enhancement.

3.1. Cross-modal correspondence mapping and sampling

To facilitate the interaction between modalities while maintaining the independence of each modality, it is essential to establish a mapping relationship between image features $\boldsymbol{h_c}$ and point cloud BEV features $\boldsymbol{h_p}$. We refer to the modality alignment method applied in DeepInteraction [15], which forms the decorated image features $\boldsymbol{h_c}'$ and decorated point cloud BEV features $\boldsymbol{h_p}'$ through mapping and sampling, thus establishing a correspondence between the two modalities.

The mapping and sampling process in the Image Feature Enhancement Module (IFEM) is as follows, as illustrated in Fig. 3(a). First, each 3D point $P(x, y, z)$ in space is projected onto multiple views to form a sparse depth map. Then, a dense depth map is generated through bilinear interpolation, which is subsequently back-projected into 3D space to form a pseudo point cloud. By using the horizontal coordinates of this pseudo point cloud, the corresponding point cloud BEV features $Pb(x_p, y_p)$ can be sampled, thereby generating the decorated image features $\boldsymbol{h_c}'$. At this stage, the decorated image features $\boldsymbol{h_c}'$ have the same dimensions as the original image features $\boldsymbol{h_c}$, and both the original image features $\boldsymbol{h_c}$ and the decorated image features $\boldsymbol{h_c}'$ carry respective information from the image and the point cloud.

The mapping and sampling process in the LiDAR Feature Enhancement Module (LFEM) is as follows, as depicted in Fig. 4(a). Similar to $\boldsymbol{h_c}'$, each 3D point $P(x, y, z)$ in the point cloud is projected onto multiple views. Valid points that can be projected onto these views are selected, and the features at the projection positions in the multi-view images are sampled to form sampling results in voxel units. Non-empty voxels are retained as the decorated point cloud BEV features $\boldsymbol{h_p}'$. At this stage, the point cloud BEV features $\boldsymbol{h_p}$ and the decorated point cloud BEV features $\boldsymbol{h_p}'$ are matched in voxel dimensions, with $\boldsymbol{h_p}$ and $\boldsymbol{h_p}'$ carrying respective feature information from the point cloud and the image.

The transformation formula for converting the point cloud from LiDAR coordinates to camera coordinates, mapping to image coordinates, and then discretizing to pixel coordinates is as follows:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} \tag{1}$$

$$K = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

$$H = \begin{bmatrix} R & t \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3}$$

where $\boldsymbol{K}$ represents the camera's intrinsic parameter matrix, and $\boldsymbol{H}$ is the

transformation matrix from LiDAR to camera.

The sampling value of the LiDAR point $\boldsymbol{P}$ on the image feature map $\boldsymbol{F}(\boldsymbol{P})$ can be obtained by bilinear interpolation of the values of the four surrounding pixel points. Similarly, the sampling value on the BEV feature map for the image pixel point (u, v) can also be obtained by bilinear interpolation of the values of the four surrounding BEV grids.

The sampling process is as follows. Given the input feature map $\boldsymbol{F}$, the corresponding sampling value $\boldsymbol{F}(\boldsymbol{P})$ for the sampling point $\boldsymbol{P}$ on the feature map $\boldsymbol{F}$ can be calculated using the following bilinear interpolation formula:

$$F(P) = \alpha\beta F(Q_{22}) + \alpha(1-\beta)F(Q_{21}) + (1-\alpha)\beta F(Q_{12}) + (1-\alpha)(1-\beta)F(Q_{11}) \tag{4}$$

where $\boldsymbol{Q}_{11}$, $\boldsymbol{Q}_{12}$, $\boldsymbol{Q}_{21}$, $\boldsymbol{Q}_{22}$ represent the four nearest neighbor pixel/BEV grid points around the sampling point $\boldsymbol{P}$ on the feature map $\boldsymbol{F}$. Besides, α and β are the ratios of the distance of point $\boldsymbol{P}$ to its left (or above) to the distance of the adjacent grid in the x and y directions, respectively. $\boldsymbol{F}(\boldsymbol{Q}_{22})$, $\boldsymbol{F}(\boldsymbol{Q}_{21})$, $\boldsymbol{F}(\boldsymbol{Q}_{12})$, and $\boldsymbol{F}(\boldsymbol{Q}_{11})$ separately represent the values of these four adjacent pixels on the feature map.

3.2. Attention-based feature interaction

Based on the cross-modal feature sampling and mapping relationship between the image features $\boldsymbol{h_c}$ and point cloud BEV features $\boldsymbol{h_p}$, we design an attention-based feature interaction module to fully utilize image information. This approach addresses the drawbacks of unilateral fusion and reduces the issue of module stacking.

First, we introduce an image feature enhancement module, as illustrated in Fig. 3(b), which uses the decorated image features $\boldsymbol{h_c}'$ to enhance the image features $\boldsymbol{h_c}$. The original features $\boldsymbol{h_c}$ interact with the sampled features $\boldsymbol{h_c}'$ through local cross-attention, treating $\boldsymbol{h_c}$ as the query and $\boldsymbol{h_c}'$ as both the key and value. Each feature point of $\boldsymbol{h_c}$ queries the corresponding feature point within the local window of $\boldsymbol{h_c}'$, and after this interaction, the image features $\boldsymbol{h_c}$ incorporate point cloud information, resulting in the enhanced image features $\boldsymbol{h_{ce}}$. The formula is as follows:

$$f(h_c, h_c') = softmax\left(\frac{q_c k_p}{\sqrt{d}}\right)v_p \tag{5}$$

where $\boldsymbol{q_c}$ represents the query from the image features $\boldsymbol{h_c}$, and $\boldsymbol{k_p}$, $\boldsymbol{v_p}$ represent key and value from the sampled point cloud BEV features from $\boldsymbol{h_c}'$.

Then, we present a point cloud BEV feature enhancement module, as illustrated in Fig. 4(b): a module that uses the enhanced image features $\boldsymbol{h_{ce}}$ to inversely enhance the point cloud BEV features $\boldsymbol{h_p}$. We perform sampling on the enhanced multi-view image features $\boldsymbol{h_{ce}}$ to obtain the augmented point cloud BEV features $\boldsymbol{h_p}'$, where $\boldsymbol{h_p}$ represents the point cloud BEV features, and $\boldsymbol{h_p}'$ represents the feature sampling points of point cloud points within each voxel across different views. Using $\boldsymbol{h_p}$ as the query and $\boldsymbol{h_p}'$ as both the key and value, local attention interactions are conducted within the corresponding BEV. After the interaction, the point cloud BEV features $\boldsymbol{h_p}$ incorporate the enhanced image information, resulting in the enhanced point cloud BEV features $\boldsymbol{h_{pe}}$, which further improve the feature expressiveness. The formula is as follows:

$$f(h_p, h_p') = softmax\left(\frac{q_p k_c}{\sqrt{d}}\right)v_c \tag{6}$$

where $\boldsymbol{q_p}$ represents the query from point cloud BEV features, and $\boldsymbol{k_c}$, $\boldsymbol{v_c}$ represent key and value from the sampled enhanced image features $\boldsymbol{h_p}'$.

These two modules are concatenated in a cascading manner to form an alternating interaction method. The input to a single module is the original image features $\boldsymbol{h_c}$ and the original point cloud BEV features $\boldsymbol{h_p}$, with the output being the enhanced image features $\boldsymbol{h_c}$ and point cloud BEV features $\boldsymbol{h_p}$. Compared with the parallel interaction encoders of DeepInteraction [15], the alternating interaction method can simplify half of the transformer modules under the same stack depth, making the feature fusion network structure more concise and lightweight.

3.3. Decoder

We use a DETR-type decoder. First, we initialize a series of bounding boxes, then generate object queries through an embedding layer. These queries pass through multiple Transformer layers composed of self-attention, cross-attention, and feed-forward neural network layers. Finally, we detect the targets using a series of detection heads.

The decoder's self-attention, cross-attention, and feed-forward layers are described by formulas. The calculation for self-attention layer is as follows:

$$Q_S = K_S = V_S = Q + Q_{pos} \tag{7}$$

$$Q' = LayerNorm(Q + Dropout\left(softmax\left(\frac{Q_S K_S^T}{\sqrt{d}}\right)V_S\right) \tag{8}$$

where Q and Q_{pos} are derived from the initialized bounding boxes. The formula for cross-attention layer is as follows:

$$Q_C = Q' + Q_{pos} \tag{9}$$

$$K_C = V_C = K + K_{pos} \tag{10}$$

$$Q'' = LayerNorm\left(Q' + Dropout\left(softmax\left(\frac{Q_C K_C^T}{\sqrt{d}}\right)V_C\right)\right) \tag{11}$$

where K and K_{pos} come from the point cloud and image enhancement features output by the encoder. The formula for feed-forward layer is as follows:

$$FFN(Q'') = Linear_2(Dropout(Activation(Linear_1(Q'')))) \tag{12}$$

$$Q_{final} = LayerNorm(Q'' + Dropout(FFN(Q''))) \tag{13}$$

3.4. Loss function

We adopt multi-task loss function. Overall, the total loss can be formulated as:

$$L = L_{reg} + L_{cls} + L_{iou} + L_{heatmap} \tag{14}$$

Specifically, we firstly parameterize the 3D ground-truth box as $(x_g, y_g, z_g, l_g, w_g, h_g, \theta_g, v_g)$, where (x_g, y_g, z_g) denotes the center coordinate of bounding box in 3D space, (l_g, w_g, h_g) defines the size of bounding box, θ_g is the yaw rotation along the z-axis, and v_g is the velocity of bounding box. Correspondingly, the 3D prior box can be described as $(x_a, y_a, z_a, l_a, w_a, h_a, \theta_a, v_a)$. We utilize L1 function to calculate regression loss L_{reg} for positive predictions N_{pos}, and the expression is described as:

$$L_{reg} = \frac{1}{N_{pos}}\sum_i Smooth_{L1}(\Delta r) \tag{15}$$

where $\Delta \boldsymbol{r}$ is the absolute value of the difference between $(x_g, y_g, z_g, l_g, w_g, h_g, \theta_g, v_g)$ and $(x_a, y_a, z_a, l_a, w_a, h_a, \theta_a, v_a)$.

For classification loss L_{cls}, we adopt focal loss to alleviate the foreground-background imbalance problem. The formula is as follows:

$$L_{cls} = \frac{1}{N_{pos}}\sum_i -\alpha(1-p_i)^{\gamma} log(p_i) \tag{16}$$

where p_i denotes the confidence score for the i-th box, $\alpha = 0.25$ and $\gamma = 2$ are hyperparameters.

L_{iou} is calculated as follows:

$$L_{iou} = \frac{1}{N_{pos}} \sum_{i} -I_{gt} \tag{17}$$

where I_{gt} denotes the target IoU that is computed between a positive prediction and the ground-truth box.

As for $L_{heatmap}$, GaussianFocalLoss, borrowed from our baseline method DeepInteraction [15], is used to measure the difference between the predicted heatmap ($\boldsymbol{pred}_{\boldsymbol{heatmap}}$) and the ground truth heatmap ($\boldsymbol{gt}_{\boldsymbol{heatmap}}$) generated based on a Gaussian distribution. The formula is as follows:

$$L_{heatmap} = GaussianFocalLoss(pred_{heatmap}, gt_{heatmap}) \tag{18}$$

4. Experiments and results

4.1. Experimental setup

1) Dataset.

The nuScenes dataset [44] is a large dataset designed for autonomous driving research and development, supporting tasks such as 3D object detection, tracking, and prediction. This dataset consists of three parts: the training set, validation set, and test set, containing data from 700, 150, and 150 different scenes, respectively. Each frame includes a point cloud and six calibrated images covering a 360-degree horizontal field of view, along with annotations for vehicles, pedestrians, and other dynamic objects. The data originates from a 20 Hz 32-beam LiDAR and six 12 Hz cameras with a 360-degree field of view. In terms of image data, the images from the 6 surround-view cameras are resized to 800*448 before being input into the network.

2) Evaluation metrics.

When evaluating model performance, the nuScenes dataset employs multiple metrics, including mean Average Precision (mAP) and the nuScenes detection score (NDS). The mAP is calculated based on the distance of the BEV center at various distance thresholds, using the average results under 0.5 m, 1 m, 2 m, and 4 m thresholds across ten categories. Moreover, the NDS comprehensively considers mAP along with other attribute metrics such as translation, scale, orientation and velocity. These evaluation metrics aim to thoroughly assess the model's performance in 3D object detection and tracking tasks.

4.2. Implementation details

Our implementation is based on the open-source framework mmdetection3d. For the backbone network of the image branch, we employed ResNet-50 [45]. The parameters of ResNet-50 [45] were initialized from the instance segmentation model Cascade Mask R-CNN [46] and nuImage [44], both pretrained on the COCO dataset [47]. To reduce computational costs, we halved the input image size from its original dimensions and froze the weights of the image branch during training, consistent with the method described in Deepinteraction [15]. For the point cloud branch's backbone network, we used Second [27] to further process and extract point cloud features. The voxel size was set to (0.075 m, 0.075 m, 0.2 m), with the detection range for the X and Y axes set to [-54 m, 54 m] and for the Z axis set to [-5m, 3 m].

In terms of data augmentation, we performed translations across all three axes with a standard deviation of [0.5, 0.5, 0.5], scaled the point clouds horizontally and vertically with a ratio of [0.9, 1.1], and applied random rotations within the range of $[-\pi/4, \pi/4]$. The settings of the above hyperparameters refer to DeepInteraction [15]. The data also underwent random 3D flips in horizontal and vertical directions with a probability of 0.5. We adopted the CBGS [48] class-balanced sampling strategy to generate more balanced training data. For the optimizer configuration, we used the AdamW optimizer with a learning rate of 0.0001 and a weight decay coefficient of 0.01 (lr = 0.0001, weight_decay = 0.01). The gradient clipping strategy was set with max_norm = 0.1, norm_type = 2. The learning rate adjustment strategy followed a single cycle policy with a change ratio of (10, 0.0001), and the momentum adjustment strategy also followed a single cycle policy with a change ratio of (0.895, 1). The model was trained on two NVIDIA 3090 GPUs for 6 epochs with a batch size of 2.

Table 1
Comparison with state-of-the-art methods on the nuScenes val set. 'L' and 'C' represent LiDAR and camera, respectively.

Method	Modality	Backbones		Validation	
		Image	LiDAR	mAP (%)	NDS (%)
BEVDet4D [49]	C	Swin-Base	–	42.1	54.5
BEVFormer [60]	C	V99	–	48.1	56.9
PolarFormer [50]	C	V99	–	50.0	56.2
Ego3RT [51]	C	V99	–	47.8	53.4
PointPillar [29]	L	–	–	40.1	55.0
SECOND [27]	L	–	–	50.85	61.96
CBGS [48]	L	–	–	52.8	63.3
CenterPoint [52]	L	–	VoxelNet	59.6	66.8
Transfusion-L [13]	L	–	VoxelNet	65.1	70.1
Focals Conv [53]	L	–	VoxelNet-FocalsConv	61.2	68.1
LargeKernel3D [54]	L	–	VoxelNet-LargeKernel3D	63.3	69.1
PointPainting [8]	L + C	–	–	46.4	58.1
3D-CVF [55]	L + C	–	–	52.7	62.3
FUTR3D [25]	L + C	R101	VoxelNet	64.5	68.3
MVP [56]	L + C	DLA34	VoxelNet	67.1	70.8
PointAugmenting [37]	L + C	DLASeg	VoxelNet	66.8	71.0
AutoAlignV2 [57]	L + C	CSPNet	VoxelNet	67.1	71.2
FusionPainting [58]	L + C	–	–	68.1	71.6
BEVFusion [59]	L + C	Swin-Tiny	VoxelNet	67.9	71.0
BEVFusion [10]	L + C	Swin-Tiny	VoxelNet	68.5	71.4
SparseFusion [61]	L + C	Swin-Tiny	VoxelNet	68.7 70.6	70.6
Transfusion [13]	L + C	R50	VoxelNet	67.5	71.3
DeepInteraction [15]	L + C	R50	Second	67.0	71.1
Ours	L + C	R50	Second	68.9	72.2

4.3. Comparison to the state of the art methods

As shown in Table 1, our model achieved highly competitive results on the nuScenes detection benchmark. Without utilizing any Test Time Augmentation (TTA) or model ensemble strategies, our alternating interaction method, leveraging just a simple ResNet-50 [45] image backbone, surpassed many existing advanced algorithms. Compared with TransFusion [13], where both models used ResNet-50 [45], our approach achieved a performance improvement of (+1.43 % mAP, +0.89 % NDS). We attribute this performance gain to our fusion stage, which, by alternating interactions, not only preserved the independence of image features but also enhanced the features of both modalities and fully utilized image information, thereby improving feature expressiveness. In contrast, TransFusion [13], with its LiDAR-centric and image-auxiliary detection model, could not fully exploit image information, thus capping the model's performance potential. Compared with PointAugmenting [37], another multimodal model that primarily focuses on point cloud information with image information as auxiliary, our method also achieved significant performance gains (+2.13 % mAP, +1.19 % NDS). Compared with DeepInteraction [15], which optimizes feature representation through parallel interaction and employs the same image backbone ResNet-50 [45] and point cloud backbone Second [27], our method also saw notable improvements (+1.91 % mAP, +1.07 % NDS). We attribute this enhancement to our alternating interaction method eliminating the redundant parts of parallel interaction modules, removing hard-to-train distracting features, and making the retained

Table 2
Run time comparison. Absent the encoder module, the inference speed attained 6.71 FPS. Conversely, the parallel fusion encoder method only achieved an inference speed of 2.99 FPS. Our proposed method successfully optimized the inference speed to 4.12 FPS.

Method	mAP	mATE	mASE	mAOE	mAVE	mAAE	NDS	FPS(3090)
without using encoder	–	–	–	–	–	–	–	6.71
TransFusion-L	65.1	–	–	–	–	–	70.1	8.70
TransFusion	67.5	–	–	–	–	–	71.3	3.76
DeepInteraction [15]	67.03	0.2712	0.2487	0.2764	0.2535	0.1891	71.12	2.99
Ours	68.93	0.2704	0.2495	0.2794	0.2443	0.1838	72.19	4.12

modules more capable of feature expression. Furthermore, the practice of using optimized image features to subsequently refine point cloud features, updating subsequent iterations with the latest features, also enables the modules to update more rapidly as a whole, thus exhibiting stronger feature performance.

4.4. Time consumption

We calculated the parameter count of each model. In terms of parameter count, all three models use the same backbone and neck modules, totaling approximately 33 M parameters, and the same decoder and head modules, totaling approximately 22 M parameters. The model using the parallel fusion encoder [15] has a total parameter count of 58 M, with the encoder accounting for 2.7 M. The model without any encoder has a total parameter count of 55.3 M. Our method optimizes the sequence of modality interactions, reducing the number of encoder interaction modules by half compared with the parallel fusion encoder method [15], resulting in an encoder parameter count of 1.35 M and a total model parameter count of 56.65 M.

Then, we tested the inference speed of each model on a 3090 GPU, along with analyzing the computational complexity of the encoder. The comparison results of inference speed are shown in Table 2, where the comparison methods include the parallel fusion encoder method [15] and the method without any encoder. Besides, we also considered the inference speed of the unilateral fusion method [13]. Without the encoder module, the inference speed reached 6.71 FPS, while the parallel fusion encoder method only achieved an inference speed of 2.99 FPS. Our method optimized the inference speed to 4.12 FPS. Considering that the computational power applied in our experiments is analogous to that of in-vehicle computing hardware, the diminished computational cost realized by our research is indeed applicable to practical deployment scenarios.

As mentioned in [27], encoders require extensive processing and optimization of features from surround view images and point clouds, which occupy a significant portion of the overall latency in multimodal 3D detectors. Based on our analysis of the encoder structure and model parameter count, encoders that compute cross-modal attention based on mapping and sampling have complex structures and computational operations, involving numerous attention mechanisms and cross-modal fusion operations. The computational complexity of attention mechanisms grows quadratically with input sequence length, making them especially time-consuming for long sequences or large feature maps. Although the encoder accounts for a low proportion of parameters, it consumes a substantial amount of inference time, which explains the reason why the proposed method, despite reducing a small number of encoder parameters, achieves a significant improvement in inference speed.

4.5. Ablation experiments

To accurately validate the significance of our alternating interaction method in multimodal feature fusion, we conducted a comparison between models with and without the alternating interaction encoder, keeping other configurations unchanged. Experimental results, as shown in Table 3, indicate that using the alternating interaction encoder leads to a noticeable improvement (+1.37 % mAP, +0.61 % NDS), demonstrating the effectiveness of the alternating interaction method.

Furthermore, we performed a validation experiment with the Lidar2Image encoder to assess the impact of the interaction sequence on fusion performance. In our proposed AIM, image features are optimized first, followed by point cloud features, while the validation experiment reverses the order, optimizing point cloud features first and then image features. The results in Table 3 show that exchanging the feature optimization order in the Lidar2Image encoder leads to a significant degradation in model performance. Compared with the method using the alternating interaction encoder, it not only underperforms but also fares worse than the method without any alternating interaction encoder, indicating that using the Lidar2Image encoder constitutes negative optimization, resulting in decreased performance (−1.18 % mAP, −0.98 % NDS). The primary factor underlying this outcome is that, in multimodal fusion between images and LiDAR point clouds, the point cloud modality plays a crucial role in determining detection accuracy. Therefore, we choose to first perform image feature fusion, update the image information, and then incorporate it into the point cloud processing. This ensures that the point cloud receives the most updated fused state, maximizing the use of rich semantic information captured from the image to improve final detection performance. The comparison with the parallel interaction encoder is also listed in Table 3, facilitating the comparison of various methods.

The bounding boxes detected by various encoder methods are projected onto the images generated by the front-view camera, as shown in Fig. 5. Our method performs the best overall performance among the four methods. In the yellow box, other methods show significant missed detections, while our method successfully detects most targets. The method “Using parallel interaction encoder” shows a noticeable height discrepancy between the ground truth and predicted boxes, as indicated by the yellow arrow. The first two methods within the green box show false detections. For detecting small distant targets, point cloud information is less effective due to its sparsity, so image information is of greater importance.

Such findings also confirm that in the fusion of multimodal features between images and point clouds, prioritizing point cloud features as the

Table 3
Ablations on the encoder. A comparison between models with and without the alternating interaction encoder is conducted, keeping other configurations consistent. Applying the alternating interaction encoder leads to a noticeable improvement (+1.37 % mAP, +0.61 % NDS).

Method	mAP	mATE	mASE	mAOE	mAVE	mAAE	NDS
Without alternating interaction encoder	67.56	0.2706	0.2496	0.2772	0.2373	0.1850	71.58
Using Lidar2Image encoder	66.38	0.2775	0.2521	0.2884	0.2494	0.1916	70.60
Using parallel interaction encoder	67.03	0.2712	0.2487	0.2764	0.2535	0.1891	71.12
Using alternating interaction encoder (Ours)	68.93	0.2704	0.2495	0.2794	0.2443	0.1838	72.19

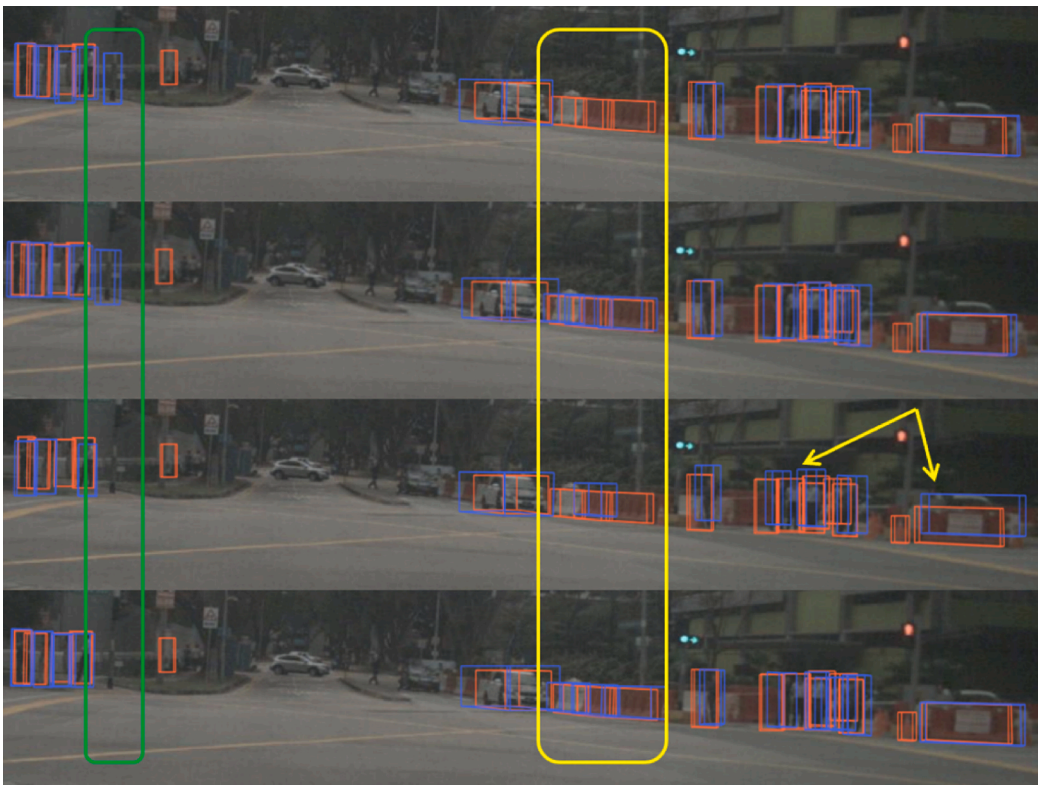


Fig. 5. These four images show the 3D bounding boxes projected onto the images generated by the front-view camera. The red boxes are the ground truth, and the blue boxes are the predictions. From top to bottom, the four images represent the methods in Table 3. The methods are "Without alternating interaction encoder", "Using Lidar2Image encoder", "Using parallel interaction encoder", and "Using alternating interaction encoder (Ours)".

main source of information and image features as auxiliary yields better results. However, previous approaches such as TransFusion [13] substantially neglected image information, leading to suboptimal fusion outcomes. In contrast, the alternating interaction module effectively balances the significance of both image and point cloud information, enabling the model to better leverage image features for improved multimodal feature fusion.

In addition, the proposed method encounters limitations in the preceding feature mapping and sampling stages due to the trade-offs inherent in traditional and attention-based alignment methods. While local attention reduces computational load, it sacrifices accuracy, and global attention, though precise, is computationally intensive. Advanced methods, such as deformable attention and polar coordinate-based alignment, present promising approaches to balancing cost and accuracy. Furthermore, the method is susceptible to adverse weather and sensor noise, highlighting the necessity for enhancing robustness strategies. To address these issues, multi-sensor integration, data augmentation and denoising techniques can enhance performance in challenging conditions. Future research will concentrate on refining these aspects to optimize multi-modal feature-level fusion.

4.6. Visualization

To understand the actual effect of the feature fusion in the alternating interaction fusion module, we first visualize the heatmaps on the BEV plane. The ground truth heatmap is generated by drawing a Gaussian distribution for each ground truth bounding box. The predicted heatmap is obtained from the point cloud BEV features through the heatmap detection head. The similarity between the ground truth and predicted heatmaps shows how close the initialized proposal boxes are to the ground truth boxes. This also reflects the expressive ability of the point cloud BEV features. In the two scenarios shown in Fig. 6, the areas circled in yellow do not contain any ground truth targets. When

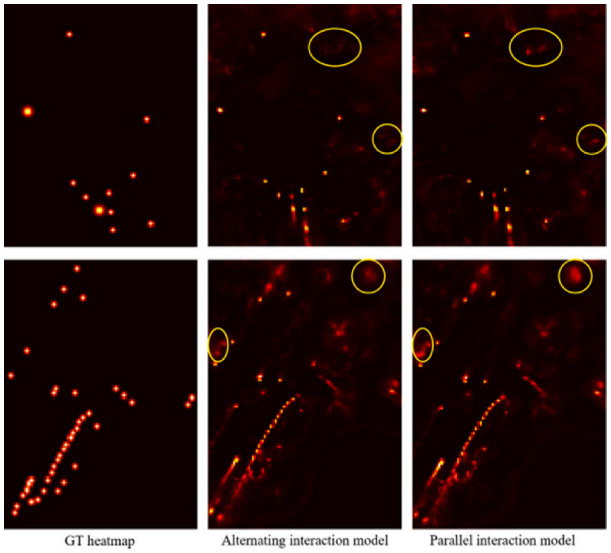


Fig. 6. These are the heatmaps for the initialized proposed bounding boxes. The first column shows the Gaussian heatmaps of the ground truth boxes, the second column shows the predicted heatmaps after feature extraction by the alternating interaction method, and the third column shows the predicted heatmaps after feature extraction by the parallel interaction model. The first and second rows represent two different scenarios.

comparing the alternating interaction module with the parallel interaction module, we see clear differences. The parallel interaction module makes significant prediction errors in these areas. It assigns a high probability of targets in regions without ground truth targets. In contrast, the alternating interaction module gives more accurate

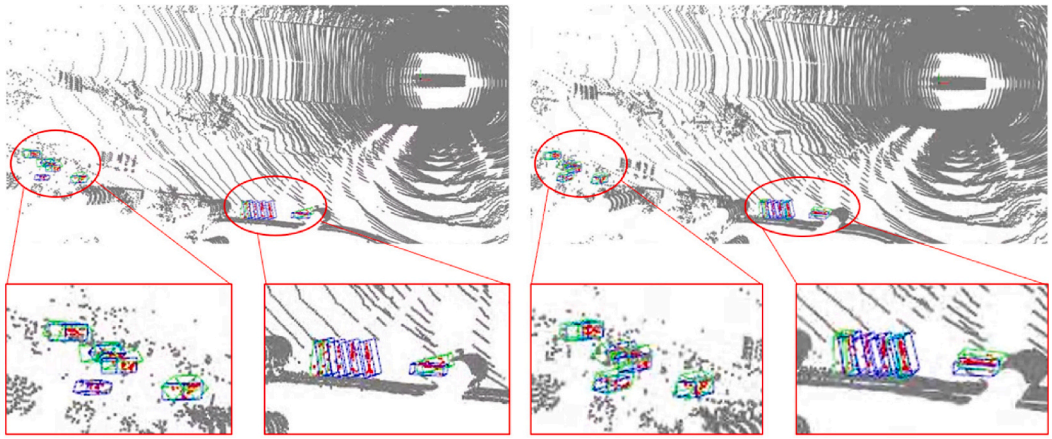


Fig. 7. Detection results of the parallel interaction method (left) and the alternating interaction method (right) on densely occluded objects. The green bounding boxes represent predicted targets, and the purple bounding boxes represent ground truth targets.

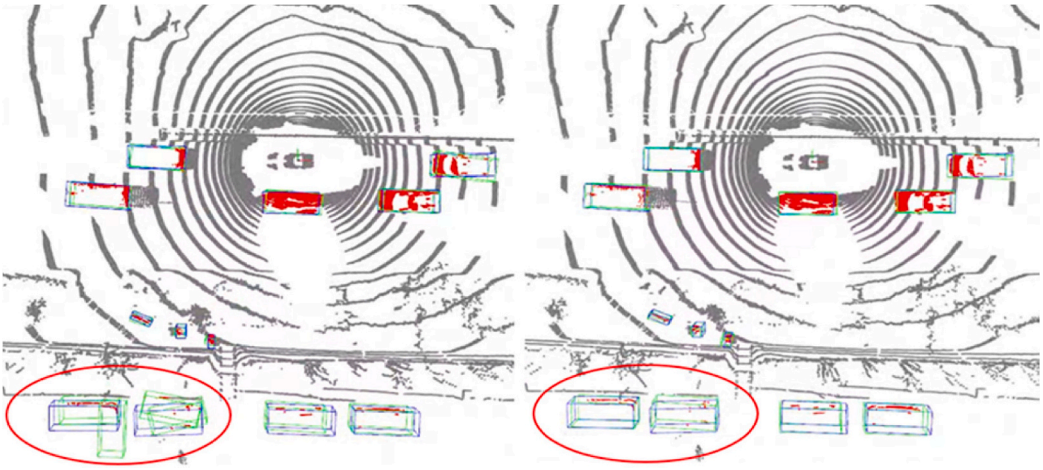


Fig. 8. Detection results of the parallel interaction method (left) and the alternating interaction method (right) on large target objects. The green bounding boxes represent predicted targets, and the purple bounding boxes represent ground truth targets.

heatmap predictions for these areas. It assigns low probability where there are no ground truth targets. This demonstrates the effective feature fusion and representation capabilities of the alternating interaction module.

We also visualize the predicted targets and ground truth targets on point cloud maps for both our alternating interaction model and the parallel interaction model [15]. Fig. 7 depicts a group of objects at a distance, dense and occluded, and another closely clustered group of objects adjacent to each other. It can be observed that the parallel interaction method fails to detect targets at a distance, especially the occluded ones. In such complex scenarios, our method is capable of identifying the occluded objects. We attribute this to the more efficient utilization of image information by the alternating interaction module. At longer ranges, the sparsity of the point cloud reduces the model's detection capabilities, which can be compensated by a full integration of image information. Even in situations where adjacent obstacles are closely connected, the alternating interaction module achieves better detection results. With the aid of more comprehensive visual information, the model's instance-level understanding is enhanced. The main issue with the detection results of the parallel interaction model presented in Fig. 8 is overlapping detection and inaccurate positioning of bounding box. We believe this misidentification is primarily due to the ineffective use of visual information, whereas our model can effectively optimize this situation.

5. Conclusion

This paper explores the sensor fusion problem between LiDAR and cameras for 3D detection in autonomous driving. Point cloud-based unilateral fusion methods do not fully use the rich contextual information in images. Parallel interaction fusion strategies enhance feature expressiveness but cause significant computational burden due to module stacking. To solve these issues, we propose an alternating interaction fusion approach. Firstly, we enhance image feature expression through local cross-attention interactions between image and point cloud features. Then, we use the enhanced image features to strengthen point cloud BEV features. Our approach avoids the limitations of point cloud-centric fusion and the complexity of module stacking in parallel fusion schemes. Experiments on the nuScenes dataset validate our method's effectiveness. Results show that our approach combines the strengths of LiDAR and cameras, improving 3D detection performance and maintaining computational efficiency. This advancement provides a new perspective for sensor fusion research in autonomous driving and sets the stage for future studies. Future work will focus on optimizing our fusion module for real-time perception processing while ensuring accurate detection. We also plan to explore the application of this fusion strategy in more autonomous driving scenarios and its robustness under different environmental conditions.

CRediT authorship contribution statement

Guofa Li: Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Haifeng Lu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jie Li:** Writing – review & editing. **Zhenning Li:** Writing – review & editing, Visualization, Methodology. **Qingkun Li:** Writing – review & editing, Visualization, Methodology. **Xiangyun Ren:** Writing – review & editing, Resources. **Ling Zheng:** Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (Grant No. 52272421), and the State Key Laboratory of Intelligent Green Vehicle and Mobility under Project No. KFZ2409.

Data availability

The authors do not have permission to share data.

References

[1] K. Wang, Z. Zhang, X. Wu and L. Zhang, "Multi-class object detection in tunnels from 3D point clouds: An auto-optimized lazy learning approach," *Advanced Engineering Informatics*, vol. 52, article ID: 101543, 2022.
[2] Y. Wang, B. Xiao, A. Bouferguene, M. Al-Hussein and H. Li, "Vision-based method for semantic information extraction in construction by integrating deep learning object detection and image captioning," *Advanced Engineering Informatics*, vol. 53, article ID: 101699, 2022.
[3] X. Qu, D. Pi, L. Zhang and C. Lv, "Advancements on unmanned vehicles in the transportation system," *Green Energy and Intelligent Transportation*, vol. 2, no. 3, 2023.
[4] G. Chen, Preface for Robust and Certifiable Perception System for Intelligent Vehicle, *Automotive Innovation* 5 (2022) 221–222.
[5] J. Lahoud, J. Cao et al., "3D Vision with Transformers: A Survey," *arXiv preprint arXiv*: 2208.04309.
[6] J. Ruan, H. Cui, Y. Huang, T. Li, C. Wu and K. Zhang, "A review of occluded objects detection in real complex scenarios for autonomous driving," *Green Energy and Intelligent Transportation*, vol. 2, no. 3, 2023.
[7] X. Liu, J. Li, J. Ma, H. Sun, Z. Xu, T. Zhang and H. Yu, "Deep transfer learning for intelligent vehicle perception: A survey," *Green Energy and Intelligent Transportation*, vol. 2, no. 5, pp. 2023.
[8] S. Vora, A.H. Lang, B. Helou, O. Beijbom, PointPainting: Sequential Fusion for 3D Object Detection, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2020 (2020) 4603–4611.
[9] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3D Object Detection Network for Autonomous Driving, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2017 (2017) 6526–6534.
[10] Z. Liu, et al., BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation, *IEEE International Conference on Robotics and Automation (ICRA)* 2023 (2023) 2774–2781.
[11] W. Chen, W. Tian, X. Xie, et al., RGB Image- and Lidar-Based 3D Object Detection Under Multiple Lighting Scenarios, *Automotive Innovation* 5 (2022) 251–259.
[12] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers," *arXiv preprint arXiv*: 2203.17270.
[13] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, C.-L. Tai, TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2022 (2022) 1080–1089.
[14] Y. Li, A.W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, B. Wu, Y. Lu, D. Zhou, et al., DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2022 (2022) 17161–17170.
[15] Z. Yang, J. Chen, Z. Miao et al., "Deepinteraction: 3d object detection via modality interaction," *arXiv preprint arXiv*: 2208.11112.
[16] Y. Zhao, W. Tian, H. Cheng, Pyramid Bayesian Method for Model Uncertainty Evaluation of Semantic Segmentation in Autonomous Driving, *Automotive Innovation* 5 (2022) 70–78.
[17] Y. Li, Z. Ge, G. Yu, et al., BEVDepth: Acquisition of Reliable Depth for Multi-View 3D Object Detection, Proceedings of the AAAI Conference on Artificial Intelligence 37 (2) (2023) 1477–1485.
[18] G. Brazil, X. Liu, M3D-RPN: Monocular 3D Region Proposal Network for Object Detection, *IEEE/CVF International Conference on Computer Vision (ICCV)* 2019 (2019) 9286–9295.
[19] Z. Liu, Z. Wu, R. Tóth, SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation, *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2020 (2020) 4289–4298.
[20] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, K.Q. Weinberger, Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019 (2019) 8437–8445.
[21] Y. You, Y. Wang et al., "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," *arXiv preprint arXiv*: 1906.06310.
[22] S. Li, J. Chen, W. Peng, et al., A vehicle detection method based on disparity segmentation, *Multimed. Tools Appl.* 82 (13) (2023) 19643–19655.
[23] J. Philion, S. Fidler, Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d, *Computer Vision - ECCV* 2020 (2020) 194–210.
[24] P. Sun, Y. Wang, P. He, et al., GCD-L: A Novel Method for Geometric Change Detection in HD Maps Using Low-Cost Sensors, *Automotive Innovation* 5 (2022) 324–332.
[25] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "FUTR3D: A Unified Sensor Fusion Framework for 3D Detection," *arXiv preprint arXiv*: 2203.10642.
[26] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. M. Solomon, "DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries," *Proceedings of the 5th Conference on Robot Learning*, PMLR 164:180-191, 2022.
[27] Y. Yan, Y. Mao, B. Li, Second: Sparsely embedded convolutional detection, *Sensors* 18 (2018) 3337.
[28] Y. Zhou, O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.
[29] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, PointPillars: Fast Encoders for Object Detection from Point Clouds, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12697–12705.
[30] L. Fan, X. Xiong, F. Wang, N. Wang, Z. Zhang, RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2918–2927.
[31] J. Mao, Y. Xue, M. Niu, et al., Voxel transformer for 3d object detection, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3164–3173.
[32] X. Pan, Z. Xia, S. Song, et al., 3d object detection with pointformer, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7463–7472.
[33] Z. Liu, X. Yang, H. Tang, S. Yang, S. Han, FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2023 (2023) 1200–1211.
[34] P. Wang, "Octformer: Octree-based transformers for 3d point clouds," *arXiv preprint arXiv*: 2305.03045.
[35] N. Carion et al., "End-to-end object detection with transformers," *arXiv preprint arXiv*: 2005.12872.
[36] I. Misra, R. Girdhar, A. Joulin. "An end-to-end transformer model for 3d object detection," *arXiv preprint arXiv*: 2109.08141.
[37] C. Wang, C. Ma, M. Zhu, X. Yang, PointAugmenting: Cross-Modal Augmentation for 3D Object Detection, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2021 (2021) 11789–11798.
[38] X. Chen, J. Cui, Y. Liu, et al., Joint scene flow estimation and moving object segmentation on rotational LiDAR data, IEEE Trans. Intell. Transp. Syst. 25 (11) (2024) 17733–17743.
[39] Z. Li, Y. Wang, R. Zhang, et al., A LiDAR-OpenStreetMap matching method for vehicle global position initialization based on boundary directional feature extraction, IEEE Trans. Intell. Veh. (2024).
[40] J. Ku, M. Mozifian, J. Lee, A. Harakeh, S.L. Waslander, Joint 3D Proposal Generation and Object Detection from View Aggregation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2018 (2018) 1–8.
[41] B. Liao, S. Chen, Y. Zhang et al., "Maptrv2: An end-to-end framework for online vectorized HD map construction," *arXiv preprint arXiv*:2308.05736, 2023.
[42] J. Duan, W. Cao, Y. Zheng, L. Zhao, On the optimization landscape of dynamic output feedback linear quadratic control, *IEEE Trans. Autom. Control* 69 (2) (2024) 920–935.
[43] J. Duan, Y. Ren, F. Zhang, J. Li, S.E. Li, Y. Guan, K. Li, Encoding distributional soft actor-critic for autonomous driving in multi-lane scenarios, *IEEE Comput. Intell. Mag.* 19 (2) (2024) 96–112.
[44] H. Caesar, et al., nuScenes: A Multimodal Dataset for Autonomous Driving, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2020 (2020) 11618–11628.
[45] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2016 (2016) 770–778.
[46] Z. Cai, N. Vasconcelos, Cascade R-CNN: High Quality Object Detection and Instance Segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (5) (2021) 1483–1498.

[47] T. Lin, M. Maire, S. Belongie et al., "Microsoft coco: Common objects in context," *Computer Vision - ECCV 2014 (ECCV)*, 2014, pp. 740-755.
[48] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-Balanced Grouping and Sampling for Point Cloud 3D Object Detection," *arXiv preprint arXiv*: 1908.09492.
[49] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View," *arXiv preprint arXiv*: 2112.11790.
[50] Y. Jiang, L. Zhang, Z. Miao, and X. Zhu et al., "Polarformer: Multi-camera 3d object detection with polar transformers," *arXiv preprint arXiv*: 2206.15398.
[51] J. Lu, Z. Zhou, X. Zhu, H. Xu, and L. Zhang, "Learning ego 3d representation as ray tracing," *Computer Vision - ECCV 2022 (ECCV)*, 2022, pp. 129-144.
[52] T. Yin, X. Zhou, P. Krahenbuhl, Center-based 3d object detection and Tracking, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11784–11793.
[53] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," *arXiv preprint arXiv*: 2204.12463.
[54] Y. Chen, J. Liu, X. Qi, X. Zhang, J. Sun, and J. Jia, "Scaling up kernels in 3d cnns," *arXiv preprint arXiv*: 2206.10555.
[55] J. Yoo, Y. Kim, J. Kim, and J. Choi, "3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," *Computer Vision - ECCV 2020 (ECCV)*, 2020, pp. 720-736.
[56] T. Yin, X. Zhou, P. Krähenbühl, Multimodal virtual point 3d detection, *Adv. Neural Inf. Proces. Syst.* 34 (2021) 16494–16507.
[57] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, F. Zhao, B. Zhou, and H. Zhao, "Autoalign: Pixel-instance feature aggregation for multi-modal 3d object detection," *arXiv preprint arXiv*: 2201.06493.
[58] S. Xu, D. Zhou, J. Fang, J. Yin, B. Zhou, L. Zhang, FusionPainting: multimodal fusion with adaptive attention for 3d object detection, *IEEE International Intelligent Transportation Systems Conference (ITSC)* 2021 (2021) 3047–3054.
[59] T. Liang, H. Xie et al., "BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework," *arXiv preprint arXiv*: 2205.13790.
[60] Z. Li, W. Wang et al., "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers," *arXiv preprint arXiv*: 2203.17270.
[61] Y. Xie, C. Xu et al., "SparseFusion: Fusing Multi-Modal Sparse Representations for Multi-Sensor 3D Object Detection," *arXiv preprint arXiv*: 2304.14340.