TransMRE: Multiple Observation Planes Representation Encoding With Fully Sparse Voxel Transformers for 3-D Object Detection

Ziming Zhu, Yu Zhu, *Member, IEEE*, Kezhi Zhang, Hangyu Li, and Xiaofeng Ling

***Abstract*— The effective representation and feature extraction of 3-D scenes from sparse and unstructured point clouds pose a significant challenge in 3-D object detection. In this article, we propose TransMRE, a network that enables fully sparse multiple observation plane feature fusion using LiDAR point clouds as single-modal input. TransMRE achieves this by sparsely factorizing a 3-D voxel scene into three separate observation planes: XY, XZ, and YZ planes. In addition, we propose Observation Plane Sparse Fusion and Interaction to explore the internal relationship between different observation planes. The Transformer mechanism is employed to realize feature attention within a single observation plane and feature attention across multiple observation planes. This recursive application of attention is done during multiple observation plane projection feature aggregation to effectively model the entire 3-D scene. This approach addresses the limitation of insufficient feature representation ability under a single bird's-eye view (BEV) constructed by extremely sparse point clouds. Furthermore, TransMRE maintains the full sparsity property of the entire network, eliminating the need to convert sparse feature maps into dense feature maps. As a result, it can be effectively applied to LiDAR point cloud data with large scanning ranges, such as Argoverse 2, while ensuring low computational complexity. Extensive experiments were conducted to evaluate the effectiveness of TransMRE, achieving 64.9 mAP and 70.4 NDS on the nuScenes detection benchmark, and 32.3 mAP on the Argoverse 2 detection benchmark. These results demonstrate that our method outperforms state-of-the-art methods.**

***Index Terms*— 3-D object detection, autonomous driving, deep learning, LiDAR, multiple observation planes representation encoding, point cloud, voxel feature factorizing.**

I. Introduction

LiDAR-BASED 3-D object detection plays a crucial role in the field of computer vision, finding applications in autonomous driving and robotics. However, one of the main challenges facing this task is the ability to effectively represent 3-D scenes and extract object attributes efficiently from sparse and unstructured point clouds. To address this challenge, several methods [1], [2], [3] employ PointNet-like networks [4], [5] to extract 3-D object properties. While these approaches have shown significant advancements, they also introduce a substantial computational complexity in terms of point sampling and grouping. Consequently, these methods are not well-suited for handling large-scale autonomous driving scenes.

To achieve efficient 3-D object detection, grid-based methods in the 2-D BEV space have been proposed. These methods utilize a backbone and neck architecture similar to SECOND [6] for extracting spatial voxel features. However, the backbone and neck of SECOND-like methods [7], [8], [9], [10], [11] store spatial representation using a 3-D voxel grid, consuming significant storage and computation resources. To address this, the sparse 3-D voxel grid is compressed into a dense 2-D BEV grid, reducing storage and computation to an area-based proportion. However, this compression method wastes storage space and computation, and crucial geometric information in the height dimension is lost.

In Fig. 1, we visualize point clouds and images for typical cases in the autonomous driving scenario. It is evident that a common situation arises where the foreground objects that need to be detected are occluded by other background objects directly above them. Consequently, previous second-like grid-based methods struggle with objects exhibiting complex and diverse geometry.

Our proposed method aims to enhance object detection by incorporating multiple observation plane geometric information into spatial features. Building upon NeRF-related work [12] such as TensoRF [13] and MERF [14], by decoupling 3-D voxel features into three pairwise orthogonal planes using sparse dimensionality reduction, we generate distinct features for different voxels while reducing storage space and computation requirements. Sparse self-attention within each observation plane enables feature fusion, while sparse cross-attention between pairwise observation planes facilitates feature interactions. Integrating features from multiple observation planes during the extraction process create richer representations for precise object recognition and localization in complex scenes. We list our contributions as follows:

1) We sparsely factorize the 3-D feature space into pairwise orthogonal XY, XZ, and YZ planes. Next, we divide and sectionalize the sparse feature maps based on

Received 26 April 2024; revised 22 July 2024; accepted 30 July 2024. Date of publication 4 November 2024; date of current version 8 November 2024. This work was supported in part by Shanghai Automotive Industry Science and Technology Development Foundation under Grant 2304 and in part by the National Natural Science Foundation of China under Grant 62476088. The Associate Editor coordinating the review process was Dr. Weihua Li. *(Corresponding author: Yu Zhu.)*

Ziming Zhu, Yu Zhu, Kezhi Zhang, and Hangyu Li are with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail: zimingzhu@mail.ecust.edu.cn; zhuyu@ecust.edu.cn; y30220946@mail.ecust.edu.cn; 15008338806@163.com).

Xiaofeng Ling is with the School of Information Science and Engineering and Shanghai Key Laboratory of Intelligent Sensing and Detection Technology, East China University of Science and Technology, Shanghai 200237, China (e-mail: xfling@ecust.edu.cn).

Digital Object Identifier 10.1109/TIM.2024.3480206

the 3-D projection relationship. We then reconstruct the features into voxels using voxel reconstruction querying (VRQ). This approach reduces computational complexity and storage requirements, as they are only proportional to the area. And, it ensures a complete description of the 3-D space.

2) We propose a new module called multiple observation planes representation encoding (MRE) to improve spatial perception. MRE employs the transformers mechanism, enabling effective feature attention within a single observation plane and across multiple observation planes, based on the 3-D projection relationship. This enhances spatial perception at each interaction level.
3) The proposed method was evaluated using the nuScenes dataset and the Argoverse 2 dataset. Competitive results were achieved on both datasets, showcasing the effectiveness of TransMRE. On the nuScenes test server, TransMRE attained a detection mAP of 64.9 and an NDS of 70.4. On the Argoverse 2 validation set, TransMRE achieved a detection mAP of 32.3. These results surpass those of the state-of-the-art methods.

The rest of this article is organized as follows. Section II reviews the relevant research findings closely related to this article. Section III provides a detailed overview of the proposed TransMRE. Section IV presents experimental results to validate the effectiveness of our approach. Finally, Section V concludes this article.

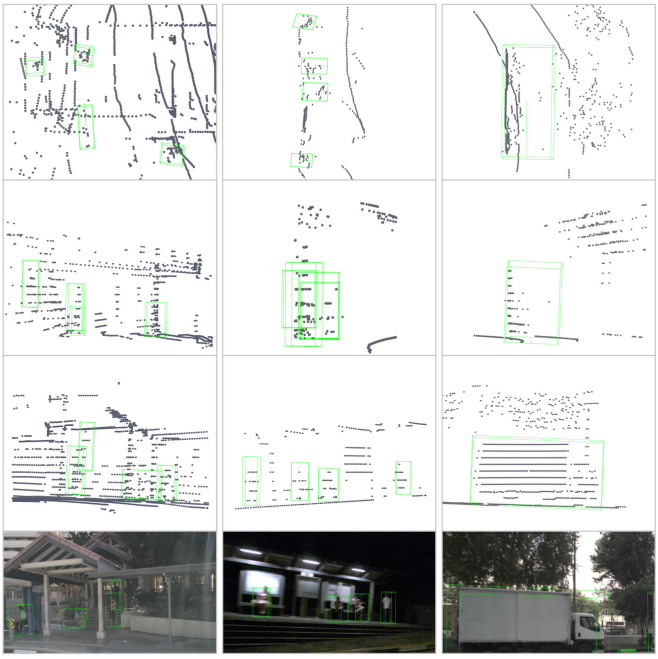

Fig. 1. Visualization of point clouds and images for three cases in the nuScenes dataset. The first three rows show the perspective views of the point clouds from the planes perpendicular to the XY plane [i.e., bird's eye view (BEV)], the XZ plane, and the YZ plane, respectively. The fourth row displays the camera images capturing the cases. For objects with occlusions from above (such as the roof or barrier above pedestrians and bicycles in cases 1 and 2, and the tree leaves obstructing the view above the truck in case 3), these occlusions can interfere with the representation of features in the XY plane, indicating that modeling spatial representations solely from the BEV perspective is incomplete.

II. Related Work

The primary approaches utilized in previous grid-based 3-D detection research involving LiDAR point cloud data is voxel-based, pillar-based, and point-based feature representations.

A. Voxel-Based Spatial Feature

VoxelNet [15] is a pioneering study in end-to-end 3-D detection. SECOND [6] enhances performance and reduces computational overhead by using 3-D sparse convolution. CenterPoint [16] refines SECOND's methodologies by employing a single positive cell for each detected object, streamlining detection pipelines and optimizing computations in crowded scenes. VISTA [17] improves object recognition by fusing 3-D feature maps from BEV and range view (RV). FocalsConv [18] selectively focuses on foreground information to enhance detection accuracy during the learning process. CenterFormer [19] improves bounding box prediction accuracy by aggregating features around the center candidate and introducing Transformers [20]. VoxelNeXt [21] enables the detection of 3-D objects solely through sparse voxel features, eliminating the need for sparse-to-dense conversion. While these methods show promise in object detection, they lack effective utilization of multiview geometric information, potentially compromising accuracy and performance.

B. Pillar-Based Spatial Feature

PointPillars [22] is a pillar-based method that utilizes PointNets [4] to encode point features. These features are then transformed into a pseudo-image in BEV using pooling operations. Infofocus [23] enhances PointPillars [22] by introducing a second-stage attention network for precise proposal refinement. PillarNet [24], based on the CenterPoint-pillar [16] architecture, incorporates a ResNet18 [25] structure with 2-D sparse convolution for efficient BEV feature extraction. Although pillar-based networks achieve accuracy comparable to voxel-based methods in 3-D object detection tasks, they suffer from a loss of critical 3-D geometric information during transformation processes. This loss poses challenges in accurately localizing objects and understanding their spatial relationships within the scene.

C. Point-Based Spatial Feature

PointNet [4] and PointNet++ [5] learn point features from raw point clouds. F-PointNet [26] and F-Conv [27] aggregate features from frustums. PointRCNN [1] generates proposals using PointNet++ [5] and refines bounding boxes with point cloud RoI pooling. STD [28] transfers sparse point features into a dense voxel representation. 3DSSD [2] introduces a sampling strategy for a feature and spatial distance fusion. IA-SSD [29] addresses sampling issues with instance-aware downsampling strategies. FSD [30] employs point clustering and group correction techniques inspired by VoteNet [3]. Point-based methods learn features directly from raw point sets, avoiding sampling information loss during voxelization. However, they have limitations in learning capacity and can be time-consuming and inefficient, affecting overall performance.

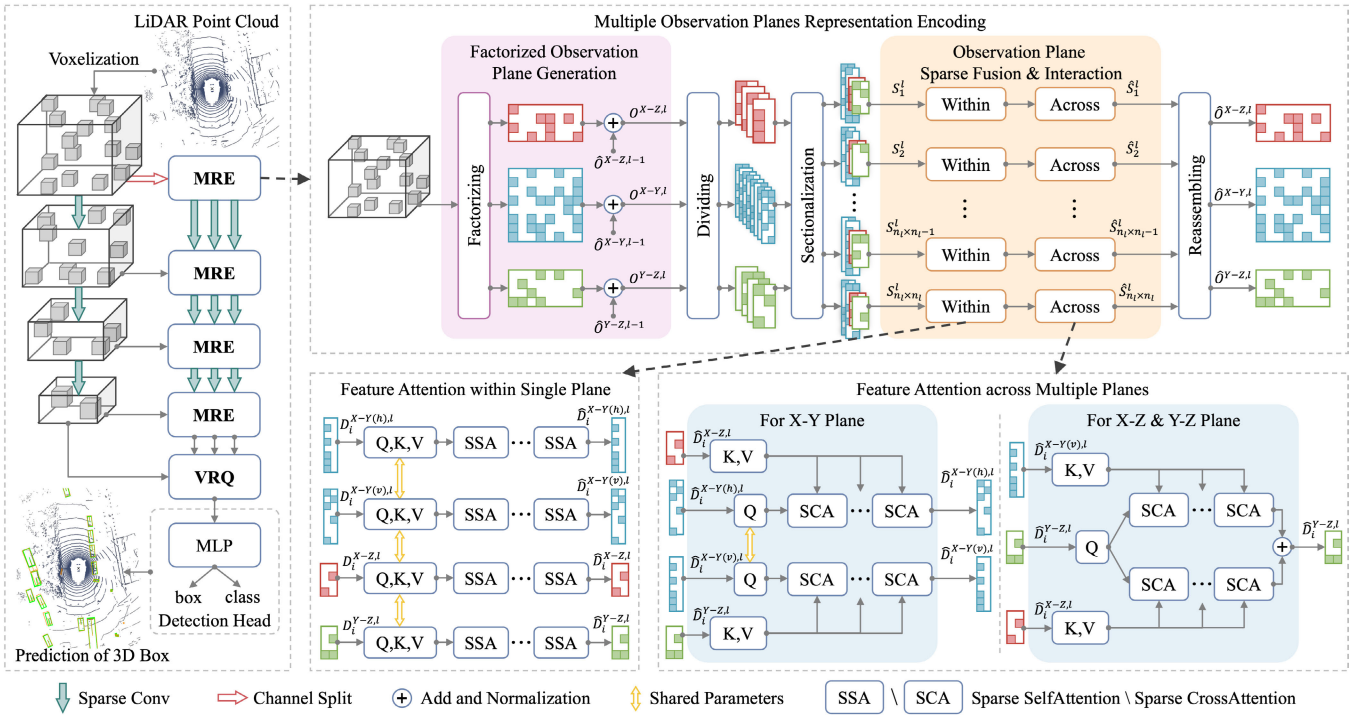

Fig. 2. Overall architecture of the proposed TransMRE. Our approach takes a LiDAR point cloud as input for 3-D object detection, where a serial fusion module with feature attention within a single plane (left below) and feature attention across multiple planes (right below) is proposed for the multiple observation planes feature encoding and interaction. The feature maps are color-coded to represent different spatial perspectives: gray corresponds to the voxel view, blue to the XY plane, red to the XZ plane, and green to the YZ plane.

III. Method

A. Framework Overview

Fig. 2 illustrates the structure of TransMRE, which proposes the incorporation of a Transformer-based multiple observation planes encoder. This encoder utilizes the Transformer mechanism to enhance the point cloud features in the multiple observation planes. In TransMRE, we propose incorporating MRE and VRQ to efficiently extract and reconstruct multiple observation plane features. Each observation plane represents a mesh cell feature that corresponds to one of the three planes, capturing specific observation plane information from the associated pillar area. The feature attention within a single observation plane mechanism focuses on encoding features within the same observation plane, by interacting with observation plane features. Conversely, the feature attention across multiple observation planes enables direct interaction between observation plane features across different observation planes, thereby incorporating richer contextual information from all perspectives.

To minimize the computational demands and parameter count of the entire model, we have implemented a reduction in the number of sparse convolutional layer stacks within the voxel branches. This adjustment effectively controls the computational load and parameter size, resulting in a more efficient and streamlined model overall.

B. Multiple Observation Planes Representation Encoding

1) Factorized Observation Plane Generation: To obtain a comprehensive observation of the scene from different perspectives and overcome limitations, such as object occlusion resulting from solely extracting BEV features, we employ a method of factorizing the voxel features into separate XY, XZ, and YZ planes. This factorization allows us to preserve the voxel features while constructing the three observation plane features, ensuring a more accurate representation of the scene.

The implementation of factorized observation plane generation is completely based on voxels. In 3-D voxel feature maps, it is often the case that certain channels contain redundant features that provide little benefit for prediction. To address this, we selectively choose the features F^1_{MRE} from a subset of channels, with a proportion of m, in each nonempty voxel feature vector before the first downsampling stage for the plane factorization process

$$F^l_{\text{MRE}}, F^l_{\text{vox}} = \begin{cases} F^l_{0:m\times C^l_{\text{vox}}}, F^l_{m\times C^l_{\text{vox}}:}, & \text{if } l = 1 \\ F^l, F^l, & \text{otherwise} \end{cases} \tag{1}$$

where $l \in [1, 2, 3, 4]$ represents the feature scale stage, and there are three downsampling stages in total corresponding to four feature scales. F^l is the feature of each original nonempty voxel and C^l_{vox} is the feature dimension of each original nonempty voxel.

The 3-D object detectors commonly employed in various methods (such as papers [6], [16], and [22]) utilize compression techniques to convert sparse 3-D voxel features into dense 2-D maps. This conversion is accomplished by transforming sparse features into dense ones and integrating altitude information (along the z-axis) into the channel dimension. However, these operations necessitate additional memory and

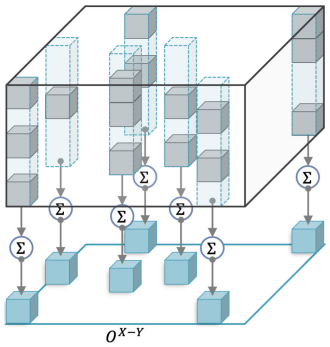

Fig. 3. Factorized observation plane generation. Given a 3-D sparse tensor, we filter out all nonempty voxels' unique 2-D coordinates relative to the 2-D plane. We then sum up the feature vectors within the pillar region of each unique 2-D coordinate in the 3-D sparse tensor as the feature vector of the new 2-D sparse tensor.

computational resources. To achieve accurate predictions while minimizing memory and computational requirements, we employ sparse dimension compression to factorize the features of each observation plane. This process entails individually placing voxels on each observation plane and aggregating the features at corresponding positions to create a 2-D sparse feature tensor ($O^{X-Y,l}$, $O^{X-Z,l}$, and $O^{Y-Z,l}$)

$$O^{\text{pln},l} = \text{SPT}\left(F^{\text{pln},l}, P^{\text{pln},l}\right) \tag{2}$$

where $\text{pln} \in [X-Y, X-Z, Y-Z]$, $\text{SPT}(\cdot,\cdot)$ represents sparse tensor construction function. The construction of $O^{X-Y,l}$, $O^{X-Z,l}$, and $O^{Y-Z,l}$ relies on the values of sparse features and their coordinate indices.

Fig. 3 takes the XY observation plane as an example to sparsely compress the original nonempty voxels onto a 2-D sparse feature tensor. Following the sparse dimension compression, the calculation of sparse features ($F^{X-Y,l}$, $F^{X-Z,l}$, and $F^{Y-Z,l}$) and their corresponding coordinate positions ($P^{X-Y,l}$, $P^{X-Z,l}$, and $P^{Y-Z,l}$) is performed as follows:

$$\begin{aligned} P^{X-Y,l} &= \left\{\left(x_p^l, y_p^l, 0\right) \mid p \in P^{\text{vox},l}\right\} \\ P^{X-Z,l} &= \left\{\left(x_p^l, 0, z_p^l\right) \mid p \in P^{\text{vox},l}\right\} \\ P^{Y-Z,l} &= \left\{\left(0, y_p^l, z_p^l\right) \mid p \in P^{\text{vox},l}\right\} \end{aligned} \tag{3}$$

$$F^{\text{pln},l} = \left\{\sum_{c \in A_{\hat{c}}} f_c, \mid \hat{c} \in P^{\text{pln},l}\right\}, \quad f \in F^l_{\text{MRE}} \tag{4}$$

$$A_{\hat{c}} = \left\{c \mid x_c^l = x_{\hat{c}}^l, y_c^l = y_{\hat{c}}^l, z_c^l = z_{\hat{c}}^l, c \in P^{\text{vox},l}\right\} \tag{5}$$

where P^{vox} is the 3-D coordinate positions of each original nonempty voxel.

2) Dividing and Sectionalization: The factorized observation plane generation structure mentioned above is designed to compress the sparse 3-D feature voxels into three observation planes. This is achieved by utilizing the feature map as the representation of each perspective, allowing for multi-perspective feature interaction. However, when it comes to performing feature fusion and interaction for each observation plane, there is a challenge. The high-resolution nature of each observation plane feature, which consists of approximately 10^4 queries in the Argoverse 2 dataset, makes it inefficient and redundant to compute full-scale attention directly within the same observation plane. This is mainly due to the significant computational costs and GPU memory requirements involved.

To overcome this issue, we propose a twofold solution. First, we propose a dividing scheme that allows us to temporally divide the full-size feature map into nonoverlapping blocks. Each patch covers a small block of the feature map. Second, we implement a sectionalization scheme to group these blocks based on their projection relationship.

Specifically, in Fig. 4, on the left side, the 2-D sparse feature maps $O^{X-Y,l}$ and $O^{Y-Z,l}$, with a spatial dimension of $W^l_{X-Z/Y-Z} \times H^l_{X-Z/Y-Z}$, are uniformly divided into $g = n \times 1$ blocks along the horizontal direction, respectively. Each block has a spatial dimension of $\hat{W}^l_{X-Z/Y-Z} \times H^l_{X-Z/Y-Z}$ (assuming $W^l_{X-Z/Y-Z} = \hat{W}^l_{X-Z/Y-Z} \times n$). As for the X–Y plane, the 2-D sparse feature map $O^{X-Y,l}$ has a size of $W^l_{X-Y} \times H^l_{X-Y}$, where both spatial dimensions are of equal length. We divide it horizontally and vertically into $g = n \times 1$ blocks and $g = 1 \times n$ blocks, with each block having a spatial dimension of $\hat{W}^l_{X-Y} \times W^l_{X-Y}$ (assuming $W^l_{X-Y} = \hat{W}^l_{X-Y} \times n$). In our approach, we maintain a consistent value of n across all downsampling stages. This ensures that each divided block can effectively cover the same real-size receptive field in the sparse feature maps, regardless of their scale. Consequently, the block spatial dimension varies for each downsampling stage.

In Fig. 4, on the right side, the entire 3-D voxel space can be defined as a large cube. Within this space, multiple cube pillar regions can be projected onto the three observation planes, covering their corresponding 2-D rectangular regions. To efficiently organize these blocks, we employ a sectionalization method based on this projection approach. Let us consider an example with a XZ plane. One of its blocks, $D_1^{X-Z,l}$, belongs to the projection of the same cube as XY plane's $D_1^{X-Y(v),l}$. Similarly, for the YZ plane, one of its blocks, $D_1^{Y-Z,l}$, belongs to the projection of the same cube as the XY plane's $D_1^{X-Y(h),l}$. When these cube projections intersect, they form a cube pillar block called $\text{PL}^l_{(1,1)}$. Consequently, $D_1^{X-Z,l}$, $D_1^{Y-Z,l}$, $D_1^{X-Y(v),l}$, and $D_1^{X-Y(h),l}$ are grouped together within the same section S_1^l. By applying the same projection method, we can obtain $n \times n$ sections. This scheme ensures that blocks with similar projection relationships are sectionalized together, highlighting their strong spatial correlation. As a result, the focus of attention calculation can be concentrated within each section. Blocks that do not belong to the same projection relation do not require additional attention calculation, as their spatial correlation is low.

By applying these schemes, we can effectively address the computational and memory challenges associated with computing full-scale attention within the same observation plane. This approach allows for efficient and optimized feature fusion and interaction, enhancing the overall performance of the system.

3) Feature Attention Within Single Observation Plane: In TransMRE, we leverage feature attention within a single observation plane to enhance feature fusion within that plane. To achieve this, we gather sparse features for each plane's feature map block within section S_i^l. These sparse features are

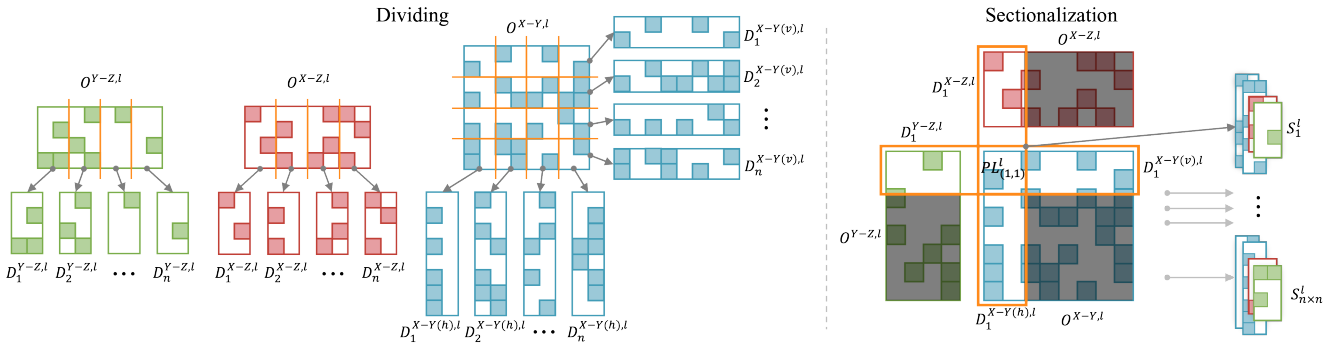

Fig. 4. Dividing and sectionalization approach. To minimize redundant calculations while maintaining accuracy, we implement a process of dividing and sectionalization on the feature maps of each observation plane. This step aims to streamline the subsequent attention calculation process.

denoted as

$$F_i^{\text{dvi},l} = \left\{ f_{1,i}^{\text{dvi},l}, \ldots, f_{N_i^{\text{dvi},l},i}^{\text{dvi},l} \right\} \tag{6}$$

where dvi $\in [X-Y(h), X-Y(v), X-Z, Y-Z]$. Each block's sparse feature in section S_i^l can be represented as a 2-D matrix, with the shape varying depending on the type of observation plane (XY plane or XZ/YZ planes) and each block. This variation arises because each block is a sparse tensor, and the number of nonempty pixels within them is not the same. For instance, in the XY plane representation, the shape is $N_i^{X-Y(h),l} \times C^l$ and $N_i^{X-Y(v),l} \times C^l$, while it is $N_i^{X-Z,l} \times C^l$ and $N_i^{Y-Z,l} \times C^l$ for the XZ and YZ planes, respectively. Here, C represents the feature dimensionality of nonempty pixels. It is noteworthy that these 2-D feature tensors are sparse and satisfy the condition $(W_i \times H_i) \geq N_i$. Then, we gather the position of each nonempty pixel within the block from the sparse tensor

$$P_i^{\text{dvi},l} = \left\{ p_{1,i}^{\text{dvi},l}, \ldots, p_{N_i^{\text{dvi},l},i}^{\text{dvi},l} \right\} \in \mathbb{R}^{N_i^{\text{dvi},l} \times 2}. \tag{7}$$

To further facilitate interaction among all sparse voxels within each observation plane, we calculate feature multihead self-attention with N_{head} heads within the observation plane on a per-plane basis

$$\begin{aligned} \text{PE}_i^{\text{dvi},l} &= \text{MLP}\left(P_i^{\text{dvi},l}\right), \quad \text{for } i = 1, \ldots, n^2 \\ \hat{F}_i^{\text{dvi},l} &= \text{MA}\Big(Q\left(F_i^{\text{dvi},l} + PE_i^{\text{dvi},l}\right) \\ &\qquad K\left(F_i^{\text{dvi},l} + \text{PE}_i^{\text{dvi},l}\right), V\left(F_i^{\text{dvi},l}\right)\Big) \\ \hat{D}_i^{\text{dvi},l} &= \text{SPT}\left(\hat{F}_i^{\text{dvi},l}, P_i^{\text{dvi},l}\right) \end{aligned} \tag{8}$$

$$\begin{aligned} \hat{F}_i^{X-Z,l} &= \text{MA}\Big(Q\left(F_i^{X-Z,l} + \text{PE}_i^{X-Z,l}\right) \\ &\qquad K\left(F_i^{Y-Z,l} + \text{PE}_i^{Y-Z,l}\right), V(F_i^{Y-Z,l})\Big) \\ &\quad + \text{MA}\Big(Q\left(F_i^{X-Z,l} + \text{PE}_i^{X-Z,l}\right) \\ &\qquad K\left(F_i^{X-Y(h),l} + \text{PE}_i^{X-Y(h),l}\right), V\left(F_i^{X-Y(h),l}\right)\Big) \\ \hat{D}_i^{X-Z,l} &= \text{SPT}\left(\hat{F}_i^{X-Z,l}, P_i^{X-Z,l}\right). \end{aligned} \tag{9}$$

the multihead self-attention mechanism utilizes linear projection layers, $Q(\cdot)$, $K(\cdot)$, and $V(\cdot)$, to generate query, key, and value features. To incorporate spatial information into the attention process, we propose a learnable index-based 2-D position embedding called PE for each nonempty pixel. This embedding is obtained by mapping the position of the nonempty pixels through a multilayer perceptron (MLP).

4) Feature Attention Across Multiple Observation Planes: Our objective is to propagate sparse pixel information at the plane-level across different observation planes, following the application of self-attention within each observation plane. This enables us to capture more comprehensive 3-D spatial information. To further enhance the fusion of features between observation planes, we calculate the sparse cross-attention across multiple planes.

Specifically, in the XY plane, there are divided blocks $D_i^{X-Y(h),l}$ in the horizontal direction and divided blocks $D_i^{X-Y(v),l}$ in the vertical direction. $D_i^{X-Y(h),l}$ and $D_i^{X-Z,l}$ are projected from the same 3-D space cube. We calculate the sparse cross-attention between $D_i^{X-Y(h),l}$ and $D_i^{X-Z,l}$ to obtain $\hat{D}_i^{X-Y(h),l}$ after the feature update. Similarly, we calculate the sparse cross-attention between $D_i^{X-Y(v),l}$ and $D_i^{Y-Z,l}$ to obtain $\hat{D}_i^{X-Y(v),l}$ after the feature update

$$\begin{aligned} \hat{F}_i^{X-Y(h),l} &= \text{MA}\Big(Q\left(F_i^{X-Y(h),l} + \text{PE}_i^{X-Y(h),l}\right) \\ &\qquad K\left(F_i^{X-Z,l} + \text{PE}_i^{X-Z,l}\right), V\left(F_i^{X-Z,l}\right)\Big) \\ \hat{D}_i^{X-Y(h),l} &= \text{SPT}\left(\hat{F}_i^{X-Y(h),l}, P_i^{X-Y(h),l}\right). \end{aligned} \tag{10}$$

To update the feature $D_i^{X-Z,l}$, we begin by considering the divided block $D_i^{X-Z,l}$ in the XZ plane. This block is projected from the same 3-D space cube as $D_i^{X-Y(h),l}$ and $D_i^{Y-Z,l}$. We calculate the sparse cross-attention values for $D_i^{X-Y(h),l}$ and $D_i^{Y-Z,l}$ separately. Next, these values are summed to obtain the updated feature, $\hat{D}_i^{X-Z,l}$. Similarly, we calculate the sparse cross-attention values for $D_i^{X-Y(v),l}$ and $D_i^{X-Z,l}$ individually. These values are then summed to update the feature, resulting in $\hat{D}_i^{Y-Z,l}$

$$\begin{aligned} \hat{F}_i^{X-Z,l} &= \text{MA}\Big(Q\left(F_i^{X-Z,l} + \text{PE}_i^{X-Z,l}\right) \\ &\qquad K\left(F_i^{Y-Z,l} + \text{PE}_i^{Y-Z,l}\right), V(F_i^{Y-Z,l})\Big) \\ &\quad + \text{MA}\Big(Q\left(F_i^{X-Z,l} + \text{PE}_i^{X-Z,l}\right) \\ &\qquad K\left(F_i^{X-Y(h),l} + \text{PE}_i^{X-Y(h),l}\right), V\left(F_i^{X-Y(h),l}\right)\Big) \\ \hat{D}_i^{X-Z,l} &= \text{SPT}\left(\hat{F}_i^{X-Z,l}, P_i^{X-Z,l}\right). \end{aligned} \tag{11}$$

Finally, the divided blocks in each section are reassembled to form the complete sparse feature maps of each plane. This is achieved through the reverse operation described in Section III-B2.

C. Voxel Reconstruction Querying

The primary objective of the factorized multiple observation plane representation is to deliver a comprehensive description

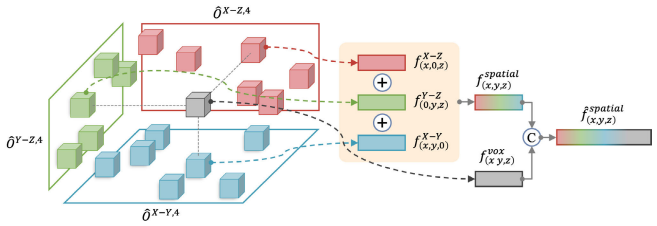

Fig. 5. VRQ module. When provided with a specific voxel location, we begin by projecting its 3-D coordinates onto each of the three axially aligned planes, XY, XZ, and YZ planes. Then, we extract the features of each projection from a sparse 2-D feature map. These extracted features are then combined and consolidated into a single feature vector.

of the query voxels situated within the 3-D space (x, y, and z). This is accomplished through the integration of voxel feature projections onto the XY, XZ, and YZ planes, following MRE. We start it after the last downsampling stage. This process is visually presented in Fig. 5.

To begin the process, we project the voxels onto the XY, XZ, and YZ planes, resulting in the coordinates [(x, y, 0), (x, 0, z), and (0, y, z)]. Following this, we sample the XY, XZ, and YZ planes at these coordinates to obtain the corresponding features [$f^{X-Y}_{(x,y,0)}$, $f^{X-Z}_{(x,0,z)}$, $f^{Y-Z}_{(0,y,z)}$]:

$$\begin{aligned} f^{X-Y}_{(x,y,0)} &= \mathrm{ID}\left(O^{X-Y,4}, \mathrm{CT}_{X-Y}(x,y,0)\right) \\ f^{X-Z}_{(x,0,z)} &= \mathrm{ID}\left(O^{X-Z,4}, \mathrm{CT}_{X-Z}(x,0,z)\right) \\ f^{Y-Z}_{(0,y,z)} &= \mathrm{ID}\left(O^{Y-Z,4}, \mathrm{CT}_{Y-Z}(0,y,z)\right). \end{aligned} \tag{12}$$

By combining these three features with the voxel's intrinsic features, we can generate the final feature representation, denoted as $\hat{f}^{\text{spatial}}_{(x,y,z)}$

$$\begin{aligned} f^{\text{spatial}}_{(x,y,z)} &= \mathrm{SUM}\left(f^{X-Y}_{(x,y,0)}, f^{X-Z}_{(x,0,z)}, f^{Y-Z}_{(0,y,z)}\right) \\ \hat{f}^{\text{spatial}}_{(x,y,z)} &= \mathrm{CONCAT}\left(f^{\text{spatial}}_{(x,y,z)}, f^{\text{vox}}_{(x,y,z)}\right). \end{aligned} \tag{13}$$

The index function $\mathrm{ID}(\cdot,\cdot)$ is responsible for sampling the feature vectors from the sparse feature maps of XY, XZ, and YZ based on the provided coordinates. To simplify the coordinate transformation process, we define a function denoted as $\mathrm{CT}(\cdot,\cdot,\cdot)$. Considering that the XY, XZ, and YZ planes align with the axes of 3-D space, each projection function $\mathrm{CT}(\cdot,\cdot,\cdot)$ only needs to perform a straightforward index dimensionality reduction on the two associated coordinate systems it encompasses.

D. Detection Head

During training, we assign the nonempty voxel closest to the center of each labeled the bounding box as a positive sample, and supervised it with focal loss [31], denoted as L_{Pos}. The overall loss function is defined by weighting the classification, regression, and IoU costs

$$L_{\text{Total}} = \lambda_1 L_{\text{Pos}} + \lambda_2 L_{\text{Cls}} + \lambda_3 (L_{\text{Reg}} + L_{\text{IoU}}). \tag{14}$$

Specifically, we directly regress T class scores $S^{\text{spatial}} \in \mathbb{R}^{N\times T}$ and the bounding box from the nonempty sparse voxel feature $\hat{F}^{\text{spatial}} \in \mathbb{R}^{N\times C}$. The classification loss is represented by L_{Cls}, which is the cross-entropy loss. For each bounding box, we predict the position offset $(\delta x, \delta y) \in \mathbb{R}^{1\times 2}$, the height $z \in \mathbb{R}^{1\times 1}$, the 3-D sizes $(l, w, h) \in \mathbb{R}^{1\times 3}$, and the rotation angle $(\sin(\theta), \cos(\theta)) \in \mathbb{R}^{1\times 2}$. In addition, for the nuScenes dataset, we also have a regression vector velocity $(v_x, v_y) \in \mathbb{R}^{1\times 2}$. The regression loss is denoted as L_{Reg}, which is the L1 loss. To further improve performance, we calculate the IoU loss (L_{IoU}) between the predicted box and the ground-truth box [32]. The weighting coefficients of the individual loss terms are λ_1, λ_2, and λ_3. The entire network is trained under the supervision of L_{Total}. Our predictions are made using a simple MLP layer without the need for complex designs.

IV. Experiments

A. Datasets and Evaluation Metrics

In this study, we utilize two datasets, namely nuScenes [39] and Argoverse 2, to evaluate the performance of the proposed method. These datasets were collected specifically to validate the performance of LiDAR-based 3-D object detection. They encompass multimodal data from various sensors, including LiDAR point cloud and multicamera images. In addition, these datasets provide corresponding labels for each object to be detected, which include its category and 3-D bounding box.

The nuScenes dataset [33] consists of 1000 scenes, each lasting for 20 s, with approximately 1.4 million object bounding boxes across 40 000 keyframes. It is divided into 700 scenes for training, 150 scenes for validation, and 150 scenes for testing. The dataset is fully annotated with 3-D bounding boxes of 23 classes and eight attributes.

The Argoverse 2 dataset [34] contains 1000 multimodal data sequences, each containing about 150 frames, totaling approximately 150 000 frames available for training, validation, and testing. It is divided into 700 sequences for training, 150 sequences for validation, and 150 sequences for testing. This dataset provides 3-D cuboid annotations for 26 target categories. Notably, the perception range of this dataset is 200 m in radius, covering an area of 400 × 400 m.

Mean average precision (mAP) is the most commonly used performance evaluation metric in object detection tasks, and it is closely related to precision (prec) and recall (rec). Precision refers to the ratio of the actual number of positive samples to the total number of detected positive samples, while recall refers to the ratio of the detected positive samples to the total number of actual positive samples. The trends of these two metrics are usually inversely proportional, as when the recall is high, false alarms may lower the precision, and when the precision is high, missed detections may lower the recall. The calculation formulas for precision (prec), recall (rec), and mAP are as follows:

$$\begin{aligned} \text{prec} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{rec} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ \text{AP} &= \int_0^1 \text{prec}(\text{rec})\text{drec} \\ \text{mAP} &= \frac{1}{T}\sum_{i=1}^{T} \text{AP}_i \end{aligned} \tag{15}$$

where TP represents the number of correctly detected positive samples, FP represents the number of false positive samples misclassified as positive, and FN represents the number of missed positive samples. T represents the number of classes.

The nuScenes detection score (NDS) is a metric designed to evaluate models experimented on the nuScenes dataset. It builds upon mAP by additionally defining five TP metrics: average translation error (ATE) represents the Euclidean center distance in 2-D (measured in meters). The average scale error (ASE) is calculated as the 3-D intersection over union (IoU) after aligning orientation and translation ($1 - \text{IoU}$). The average orientation error (AOE) is defined as the smallest yaw angle difference between the prediction and the ground truth (measured in radians). All angles are measured within a full 360° period, except for the barriers class where they are measured within a 180° period. The average velocity error (AVE) is determined as the absolute velocity error, represented by the L2 norm of the velocity differences in 2-D (measured in m/s). The average attribute error (AAE) is defined as 1 minus the attribute classification accuracy ($1 - \text{acc}$). For each TP metric, calculate the mean TP metric (mTP) across all classes, and the calculation formulas for NDS are as follows:

$$\begin{aligned} \text{mTP} &= \frac{1}{T}\sum_{i=1}^{T}\text{TP}_i \\ \text{NDS} &= \frac{1}{10}\left[5\text{mAP} + \sum_{\text{mTP}\in\mathbb{TP}}(1 - \min(1, \text{mTP}))\right] \end{aligned} \tag{16}$$

where $\mathbb{TP}$ represents the set of the five mean true positive metrics.

According to the dataset official settings, we use mAP and NDS to evaluate the performance of the model on the nuScenes dataset and mAP to evaluate the performance of the model on the Argoverse 2 dataset.

B. Implementation Details

The model implementation was trained using four NVIDIA RTX A6000 GPUs and optimized with the Adam algorithm [40]. The initial learning rate was set to 10^{-3} and decayed to 10^{-4} using a cosine annealing schedule. The weight decay value was set to 10^{-2}, and gradients were clipped by a norm of 35.

For the nuScenes dataset, a batch size of 16 was used, and the models were trained for 20 epochs. The point cloud range (PCR) was limited to [−54, 54 m] for the X- and Y-axes, and [−5, 3 m] for the Z-axis in the LiDAR coordinate system. The voxel size used in the voxelization process for the point clouds was set to (0.075, 0.075, 0.2 m). Data augmentation techniques such as random flipping, global scaling, global rotation, GT sampling [6], and translation augmentations were applied. Flipping was randomly conducted along the X- and Y-axes. The rotation angle was randomly selected between −45° and 45°. Global scaling was performed by sampling a factor between 0.9 and 1.1. The translation noise factors were sampled between 0 and 0.5. It is important to note that GT sampling was removed in the last five training epochs [41] for the test submission models.

For the Argoverse2 dataset, the models were trained for six epochs with a batch size of 16. The PCR was constrained to [−200, 200 m] for the X- and Y-axes, and [−20, 20 m] for the Z-axis in the LiDAR coordinate system. During the voxelization process for the point clouds, a voxel size of (0.1, 0.1, 0.2 m) was used. Similar data augmentation techniques were employed as in the nuScenes dataset, with the exception that ground-truth sampling was not utilized.

C. Key Findings

In this study, we conducted a comparison between TransMRE and other state-of-the-art models for 3-D detection based on point clouds. The results obtained from the nuScenes test server and the nuScenes validation set are presented in Tables I and II, respectively. Our TransMRE outperformed several other detection methods. Notably, when compared to VoxelNeXt [21], which also utilizes a fully sparse representation, our proposed method with a three observation planes fusion encoder demonstrated significant improvements. Specifically, we observed an increase of 0.4 mAP and 0.4 NDS on the nuScenes test server.

We observed that TransMRE exhibited notably higher accuracy in detecting pedestrians and traffic cones compared to other solutions. This can be attributed to the fact that pedestrian and traffic cone bounding boxes are more susceptible to interference and occlusion issues due to their small size. By introducing additional observation plane capabilities, TransMRE enhances the point cloud features associated with small objects, thereby improving pedestrian and traffic cone detection performance.

In Fig. 6, we showcase the visualization of TransMRE's results on the nuScenes validation set. For each sample, we project the 3-D object detection bounding box onto both the LiDAR top view and the six surround camera images. The results demonstrate that TransMRE effectively detects the target object in the majority of cases, indicating the accurate localization of the object through the fusion of multiple observation planes. However, we did observe a small number of undetected objects. Further analysis revealed that these undetected objects had limited or no point clouds within their ground-truth bounding boxes. Consequently, empty voxels were generated, hindering the model's ability to extract meaningful features.

D. Long-Range Detection

To fully explore the capabilities of TransMRE, we conducted long-range detection experiments on the Argoverse 2 dataset [34]. Unlike other widely adopted 3-D detection benchmarks such as WOD [43], nuScenes [33], and KITTI [44], the Argoverse 2 dataset offers a significantly longer perception range of 200 m. This allowed us to assess the performance of TransMRE in detecting objects within a larger distance. Moreover, the Argoverse 2 dataset presents another challenge by including objects from 26 different classes, which poses a long-tail issue. This means that there is an imbalanced distribution of objects across these classes, making the detection task more challenging.

TABLE I

PERFORMANCE OF 3-D OBJECT DETECTION METHODS ON THE NUSCENES TEST SERVER. IN ALL MODELS, ONLY A SINGLE-FRAME POINT CLOUD IS USED AS INPUT DURING TESTING, WITHOUT ANY ADDITIONAL AUGMENTATIONS OR MODEL ENSEMBLES INCORPORATED. THE OPTIMAL VALUE FOR EACH METRIC IS INDICATED IN BOLD, WHILE THE SECOND BEST VALUE IS INDICATED WITH AN UNDERLINE

Methods	Publication	NDS↑	mAP↑	FLOPs (G)↓	Car↑	Truck↑	Bus↑	Trailer↑	Construction Vehicle↑	Pedestrian↑	Motorcycle↑	Bicycle↑	Traffic Cone↑	Barrier↑
PointPillars [22]	CVPR'19	45.3	30.5	48.3	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
3DSSD [2]	CVPR'20	56.4	42.6	/	81.2	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
CBGS [35]	arXiv	63.3	52.8	152.3	81.1	48.5	54.9	42.9	10.5	80.1	51.5	22.3	70.9	65.7
CenterPoint [16]	CVPR'21	65.5	58.0	184.2	84.6	51.0	60.2	53.2	17.5	83.4	53.7	28.7	76.7	70.9
CVCNET [36]	NIPS'20	66.6	58.2	238.3	82.6	49.5	59.4	51.1	16.2	83.0	61.8	38.8	69.7	69.7
HotSpotNet [37]	ECCV'20	66.0	59.3	/	83.1	50.9	56.4	53.3	23.0	81.3	63.5	36.6	73.0	71.6
AFDetV2 [38]	AAAI'22	68.5	62.4	/	86.3	54.2	62.5	58.9	26.7	85.8	63.8	34.3	80.1	71.0
Focals Conv [18]	CVPR'22	70.0	63.8	259.7	**86.7**	56.3	67.7	**59.5**	23.8	87.5	64.5	36.3	81.4	74.1
VISTA [17]	CVPR'22	70.4	63.7	/	84.7	54.2	64.0	55.0	**29.1**	83.6	71.0	45.2	78.6	71.8
TransFusion-L [39]	CVPR'22	70.2	**65.5**	159.1	86.2	**56.7**	66.3	58.8	28.2	86.1	68.3	44.2	82.0	**78.2**
VoxelNeXt* [21]	CVPR'23	70.0	64.5	37.9	84.6	53.0	64.7	55.8	28.7	85.8	**73.2**	**45.7**	79.0	74.6
TransMRE*	Ours	**70.4**	64.9	**36.4**	86.3	56.5	**69.9**	59.3	24.1	**87.7**	64.4	41.0	**82.7**	77.2

* indicates the fully sparse method.

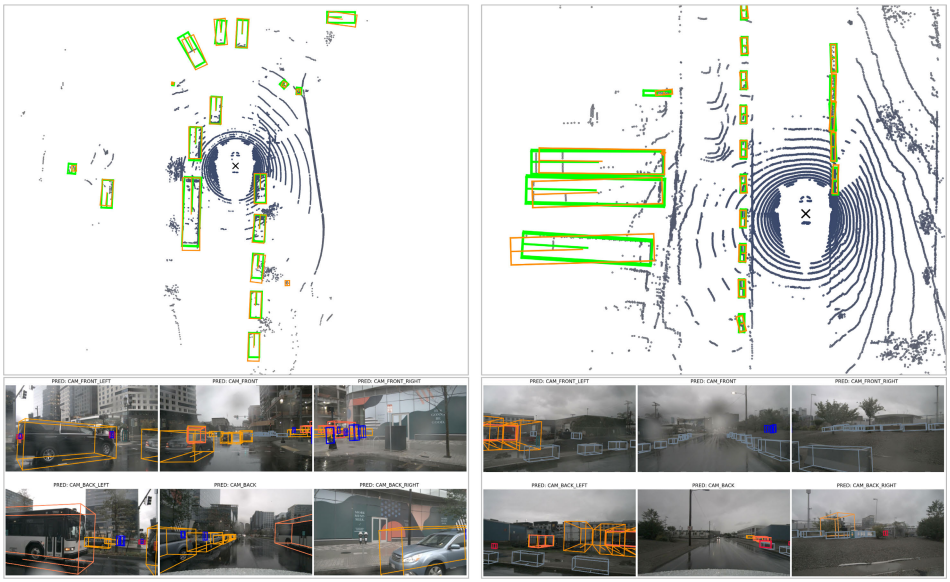

Fig. 6. Visualization of TransMRE on the nuScenes Dataset val set. In the LiDAR top view, the green bounding box corresponds to the ground truth, while the orange bounding box represents the prediction. On the camera images, all bounding boxes are predictions. Orange boxes indicate car and bus, blue boxes indicate pedestrian, dark orange boxes indicate truck, red boxes indicate motorcycle and bicycle, and light blue boxes indicate barrier.

TABLE II

PERFORMANCE OF 3-D OBJECT DETECTION METHODS ON THE NUSCENES VALIDATION SET

Methods	Publication	NDS↑	mAP↑
PointPillars [22]	CVPR'19	58.2	44.6
3DSSD [2]	CVPR'20	59.6	46.1
CBGS [35]	arXiv	62.2	50.6
CenterPoint [16]	CVPR'21	66.0	58.6
CenterFormer [19]	ECCV'22	65.2	55.4
PillarNet [24]	ECCV'22	67.4	59.9
TransFusion-L [39]	CVPR'22	68.7	62.7
Pillar3D-Former [42]	IEEE TIM'23	69.3	62.9
VoxelNeXt* [21]	CVPR'23	70.5	63.1
TransMRE*	Ours	**71.6**	**63.3**

* indicates the fully sparse method.

In Table III, we present a comparison between TransMRE and other 3-D object detectors on the Argoverse 2 dataset. It is evident that TransMRE outperforms VoxelNeXt [21], FSD [30], and CenterPoint [16] in the average metric, showcasing its superior performance. Notably, TransMRE demonstrates significant improvements over VoxelNeXt and CenterPoint in detecting both tiny objects, such as Pedestrians and Construction Cones, as well as objects with extremely large sizes, such as Articulated Buses and School Buses. This remarkable performance can be attributed to the virtue of plane-level comprehensive feature extraction in MRE, which allows TransMRE to effectively capture and leverage the necessary information for accurate detection.

To showcase the efficiency of TransMRE in long-range detection, we analyzed by observing the testing latency and memory usage of VoxelNeXt [21], TransFusion-L [39], and CenterPoint [16]. We achieved this by restricting the scan ranges of the input point cloud during model inference,

TABLE III

PERFORMANCE OF 3-D OBJECT DETECTION METHODS ON THE ARGOVERSE 2 DATASET. IN ALL MODELS, ONLY A SINGLE-FRAME POINT CLOUD IS USED AS INPUT DURING TESTING, WITHOUT ANY ADDITIONAL AUGMENTATIONS OR MODEL ENSEMBLES INCORPORATED

Methods	Publication	mAP↑	Regular Vehicle ↑	Bus ↑	Pedestrian ↑	Stop Sign ↑	Box Truck ↑	Bollard ↑	Construction Barrel ↑	Motorcyclist ↑	MPC-Sign ↑	Motorcycle ↑	Bicycle ↑	Articulated Bus ↑	School Bus ↑	Truck Cab ↑	Construction Cone ↑	Vehicular Trailer ↑	Sign ↑	Large Vehicle ↑	Stroller ↑	Bicyclist ↑
CenterPoint [16]	CVPR'21	22.0	67.6	38.9	46.5	16.9	37.4	40.1	32.2	28.6	27.4	33.4	24.5	8.7	25.8	22.6	29.5	22.4	6.3	3.9	0.5	20.1
TransFusion-L [39]	CVPR'22	23.4	69.1	40.2	47.7	18.5	38.7	41.6	33.4	30.3	28.1	35.5	25.8	10.2	27.0	**24.2**	30.5	24.2	7.2	5.7	1.6	21.8
FSD* [30]	NIPS'22	28.2	68.1	40.9	59.0	29.0	38.5	41.8	42.6	39.7	26.2	**49.0**	38.6	20.4	30.5	14.8	41.2	**26.9**	11.9	5.9	13.8	33.4
VoxelNeXt* [21]	CVPR'23	30.7	72.7	38.8	63.2	40.2	40.1	**53.9**	64.9	44.7	39.4	42.4	40.6	20.1	25.2	19.9	44.9	20.9	14.9	6.8	**15.7**	32.4
TransMRE*	Ours	**32.3**	**74.2**	**41.8**	**66.0**	**41.6**	**40.9**	52.3	**65.8**	**46.4**	**42.4**	48.4	**42.9**	**23.7**	**31.3**	19.2	**46.6**	24.7	**18.5**	**6.8**	15.1	**34.5**

* indicates the fully sparse method.

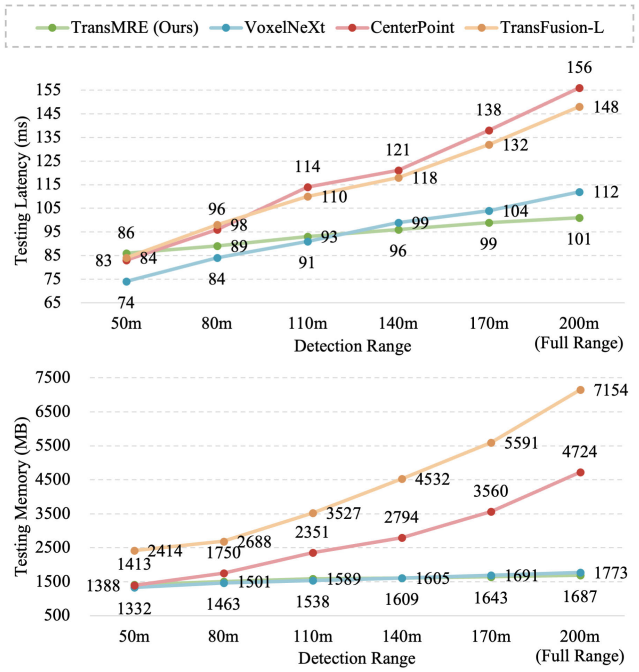

Fig. 7. Latency and memory usage on the Argoverse 2 validation set across different detection ranges. These statistics were obtained using a single NVIDIA RTX A6000 GPU with a batch size of 1. Testing latency was evaluated using the standard parameter in OpenPCDet.

TABLE IV

COMPARED TO OTHER METHODS IN TERMS OF THE NUMBER OF PARAMETERS AND COMPUTATIONAL COMPLEXITY, TRANSMRE DEMONSTRATES BETTER PERFORMANCE AND HIGHER EFFICIENCY. IT IS WORTH NOTING THAT THE MODEL'S PARAMETER COUNT REMAINS UNCHANGED AS THE DETECTION RANGE INCREASES, BUT A GREATER DETECTION RANGE LEADING TO AN INCREASE IN VOXEL FEATURE MAP SIZE RESULTS IN INCREASED COMPUTATIONAL COMPLEXITY FOR THE MODEL. TRANSMRE EXHIBITS THE SMALLEST INCREASE IN COMPUTATIONAL COMPLEXITY

Methods	Params (M)↓	FLOPs (G)↓ 50m	FLOPs (G)↓ 200m
CenterPoint [16]	8.26	139.1	956.8
TransFusion-L [39]	7.41	122.2	790.4
VoxelNeXt* [21]	3.55	28.8	35.4
TransMRE*	**3.28**	**27.9**	**34.1**

* indicates the fully sparse method.

as shown in Fig. 7. The figure demonstrates a significant increase in latency and memory requirements when applying dense detectors such as CenterPoint and TransFusion-L to larger detection ranges. In contrast, TransMRE, being designed to be fully sparse, exhibits resource needs that are roughly linear to the number of nonempty voxels. Consequently, its latency and memory usage only experience a slight increase as the detection range expands. Table IV demonstrates the high inference efficiency of TransMRE at detection ranges of 50 and 200 m. This is attributed to our multiple observation plane factorization and reconstruction strategy, as well as the fully sparse architecture that avoids densifying feature maps by making predictions only at the positions of nonempty voxels.

E. Ablation Study

We performed ablation experiments on our individual modules and parameters on the nuScenes validation set. More visualizations of the proposed TransMRE for 3-D object detection is shown in Fig. 8.

1) Factorized Observation Plane Generation: The number of factorized planes has a significant impact on the performance of object detection. This is because each factorized plane represents a distinct set of spatial information. For instance, the XY plane signifies BEV information, the XZ plane represents the vertical perspective space relative to the ego vehicle's travel direction, and the YZ plane captures the spatial representation observed from the side of the ego vehicle. In our ablation experiments, as presented in Table V, we conducted thorough analyses on the number and combination of factorized planes. Remarkably, we observed that the highest performance was achieved when TransMRE factorized all planes and integrated their features while allowing them to interact. Therefore, we adopted this configuration as the default setting in our experiments.

TransMRE splits nonempty voxel feature channels based on the selection ratio m. We examine the impact of this setting in Table VI, where we vary the selection ratio to 1/8, 1/4, 1/2, 3/4, and 7/8. We observe that the tradeoff between performance and latency achieves at its best when the ratio is equal to 1/2. As a result, we adopt a default selection ratio of 1/2 in our experiments.

2) Dividing and Sectionalization: TransMRE divides the 2-D sparse feature maps based on the number of blocks, denoted as n. To assess the impact of n on TransMRE, we conducted an ablation study. The results are summarized in Table VII, indicating that the model's mAP and NDS remain relatively stable when n is equal to or less than 15. It should be noted, however, that a slight decrease in the performance

TABLE V
ABLATION STUDY ON THE NUMBER AND COMBINATION OF FACTORIZED PLANES

X-Y	X-Z	Y-Z	mAP↑	NDS↑	Car↑	Pedestrian↑	Bicycle↑	Latency (ms)↓
✓	✓	✓	**63.3**	**71.6**	**86.7**	**87.9**	**54.5**	89
✓	✓	×	60.9 (-0.4)	71.0 (-0.6)	86.3 (-0.4)	87.5 (-0.4)	54.0 (-0.5)	80
✓	×	✓	63.1 (-0.2)	71.2 (-0.4)	86.4 (-0.3)	87.7 (-0.2)	54.2 (-0.3)	81
×	✓	✓	61.5 (-1.8)	69.5 (-2.1)	85.5 (-1.2)	86.2 (-1.7)	52.2 (-2.3)	77
✓	×	×	62.4 (-0.9)	70.4 (-1.2)	85.9 (-0.8)	86.6 (-1.3)	53.3 (-1.2)	70
×	✓	×	60.6 (-2.7)	68.5 (-3.1)	84.5 (-2.2)	84.4 (-3.5)	51.7 (-2.8)	66
×	×	✓	60.8 (-2.5)	68.6 (-3.0)	84.6 (-2.1)	85.1 (-2.8)	51.8 (-2.7)	67

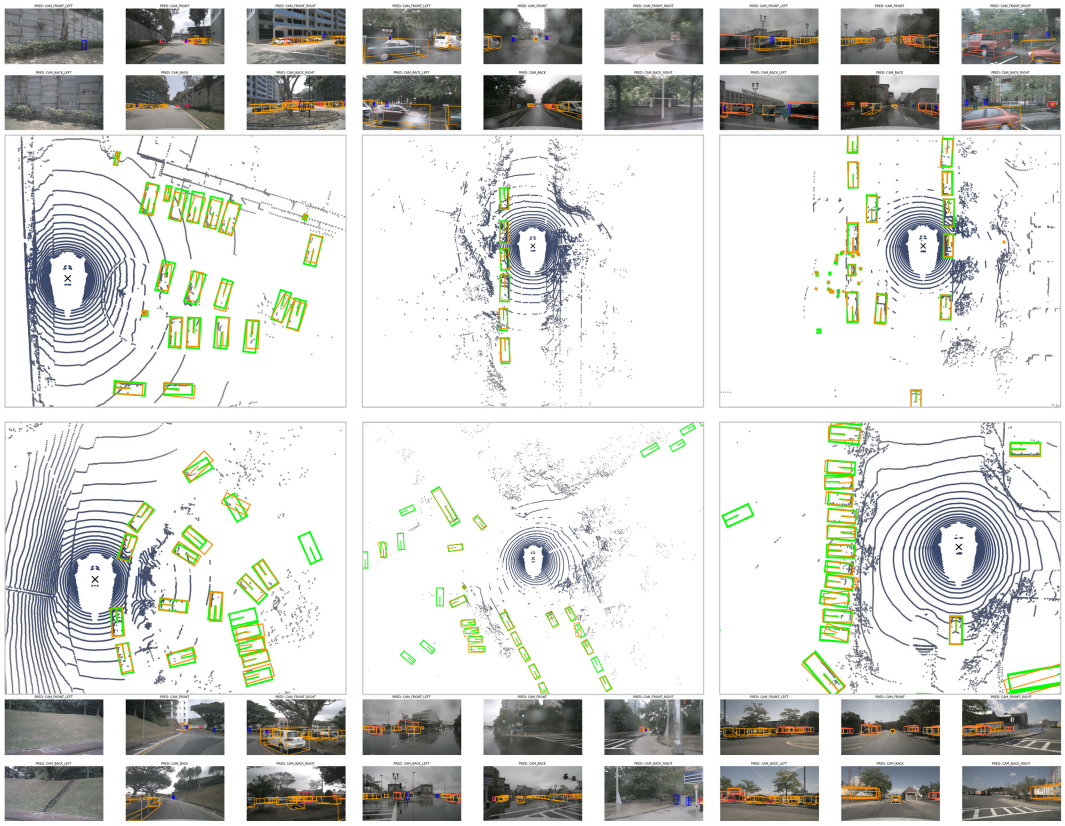

Fig. 8. More visualizations of the proposed TransMRE for 3-D object detection.

TABLE VI
EFFECTS OF SELECTION RATIO IN FACTORIZED OBSERVATION PLANE GENERATION IN OUR PROPOSED FRAMEWORK. THE SELECTION RATIO REFERS TO THE PROPORTION OF FEATURE CHANNELS THAT ARE FED INTO THE MRE MODULE. A SMALLER SELECTION RATIO MEANS THAT FEWER FEATURE CHANNELS ARE FACTORIZED INTO THREE OBSERVATION PLANES AND MORE FEATURE CHANNELS ARE RETAINED IN THE UNDERLYING 3-D TENSOR

Selection Ratio m	1/8	1/4	1/2	3/4	7/8
mAP↑	52.1	60.8	63.3	63.5	63.6
NDS↑	63.4	66.2	71.6	71.8	71.8
Car↑	77.5	83.9	86.7	86.8	86.9
Pedestrian↑	78.3	85.1	87.9	87.9	88.0
Bicycle↑	45.6	52.2	54.5	54.6	54.8
Latency (ms)↓	75	81	89	103	118

TABLE VII
EFFECTS OF THE NUMBER OF DIVIDED BLOCKS IN OUR PROPOSED FRAMEWORK. BY DEFAULT, WE SET THE NUMBER OF DIVIDED BLOCKS (n) TO 15

Block Number n	1 (undivided)	3	5	9	15	45
mAP↑	63.5	63.5	63.4	63.4	63.3	59.4
NDS↑	71.8	71.8	71.8	71.7	71.6	68.0
Car↑	88.4	87.9	87.2	87.8	86.7	83.1
Pedestrian↑	85.3	86.2	87.8	89.0	87.9	84.5
Bicycle↑	52.8	53.3	54.6	54.5	54.5	51.7
Latency (ms)↓	163	121	108	97	89	80

of pedestrian and bicycle detection is observed when n is reduced to 3. This can be attributed to the relatively small bounding boxes of pedestrians and bicycles. When the block spatial dimensions are larger, more noise information tends to be collected, which can adversely affect the accuracy of detecting small targets. Consequently, we have set the default value of n to 15.

TABLE VIII

EFFECTS OF FEATURE ATTENTION WITHIN SINGLE OBSERVATION PLANE AND FEATURE ATTENTION ACROSS MULTIPLE OBSERVATION PLANES IN OUR PROPOSED FRAMEWORK. THE FIRST ROW REPRESENTS OUR TRANSMRE WITH DEFAULT SETTINGS, WHICH SERVES AS A REFERENCE POINT FOR COMPARISON

Within	Across	mAP↑	NDS↑	Latency (ms)↓
Sparse Self-Attn	Sparse Cross-Attn	**63.3**	**71.6**	89
×	Sparse Cross-Attn	61.2 (-2.1)	69.1 (-2.5)	71
Sparse Self-Attn w/o PE	Sparse Cross-Attn	63.1 (-0.2)	71.1 (-0.5)	87
Dense MLP Mixer	Sparse Cross-Attn	63.4 (+0.1)	71.6 (-0.0)	146
Sparse Self-Attn	×	61.0 (-1.3)	70.3 (-1.2)	73
Sparse Self-Attn	Sparse Cross-Attn w/o PE	63.2 (-0.1)	71.4 (-0.2)	88

TABLE IX

ABLATION STUDY ON THE LOSS COEFFICIENT OF THE DETECTION HEAD

λ_1	λ_2	λ_3	mAP↑	NDS↑	Car↑	Pedestrian↑	Bicycle↑
0.8	1.0	0.25	**63.3**	**71.6**	**86.7**	**87.9**	**54.5**
0.8	0.5	0.25	63.0 (-0.3)	71.1 (-0.5)	86.6 (-0.1)	87.6 (-0.3)	54.3 (-0.2)
0.8	1.8	0.25	63.2 (-0.1)	71.2 (-0.4)	86.7 (-0.0)	87.7 (-0.2)	54.3 (-0.2)
0.8	2.5	0.25	/	/	/	/	/
0.1	1.0	0.25	61.5 (-0.8)	70.3 (-1.3)	86.3 (-0.4)	85.8 (-1.1)	53.6 (-0.9)
0.4	1.0	0.25	63.0 (-0.3)	71.0 (-0.6)	86.5 (-0.2)	87.5 (-0.4)	54.4 (-0.1)
1.2	1.0	0.25	62.7 (-0.5)	70.7 (-0.9)	86.4 (-0.3)	87.2 (-0.7)	54.1 (-0.4)

3) Feature Attention Within Single Observation Plane: To evaluate the effectiveness of our proposed feature attention within a single observation plane, we conducted various design experiments. In the second row of Table VIII, we removed the sparse self-attention module, which led to a decrease of 2.1 mAP and 2.5 NDS compared to the first row. By removing the index-based position embedding in the sparse self-attention module (as observed in the third row of Table VIII), we found that the proposed index-based position embedding contributed to a gain of 0.2 mAP and 0.5 NDS when compared to the first row. In the fourth row of Table VIII, we replaced the sparse self-attention module with a dense MLP mixer, resulting in a slight improvement of 0.1 mAP compared to the first row. However, we decided not to utilize this setting due to the requirement of converting the sparse feature maps to dense feature maps, which would significantly increase subsequent computation.

4) Feature Attention Across Multiple Observation Planes: By comparing the results of the first and fifth rows in Table VIII, it is evident that there is a significant decrease in both mAP (−1.3) and NDS (−1.3) when the sparse cross-attention module is removed. This highlights the importance of interplane interactions in achieving accurate detection. The model's ability to integrate information from multiple observation planes in the entire 3-D space is crucial for optimal performance. We removed the index-based position embedding in the sparse cross-attention module. Upon comparing the results of the first and sixth rows in Table VIII, we observed that the proposed index-based position embedding contributed to a gain of 0.1 mAP and 0.2 NDS. These findings further emphasize the significance of both the sparse cross-attention module and the index-based position embedding in improving the model's performance by facilitating effective interactions and information integration across different observation planes.

5) Each Loss Coefficient of the Detection Head: Table IX presents the results of ablation experiments on the weighting coefficient of a single loss in TransMRE's detection performance. We observed that when the coefficient of the classification loss is excessively large, it leads to convergence issues in the network. However, within a reasonable range, the detection performance is not significantly affected by the coefficient.

V. CONCLUSION

In this study, we propose TransMRE, a novel 3-D object detection network that is based on LiDAR technology. Unlike traditional 3-D object detection networks that convert 3-D scene features into BEV features, TransMRE takes into consideration that the features from a single perspective are often sparse and may not fully represent the entire 3-D scene. To address this, TransMRE incorporates the MRE module, which effectively factorizes the features along the XY, XZ, and YZ planes. This spatial representation greatly enhances the capability of the network. The VRQ module proposed by TransMRE reconstructs features from multiple observation planes into voxels, further improving the network's performance.

To overcome the computational cost associated with Transformer mechanisms in high-resolution feature maps, we propose a dividing and sectionalization calculation strategy. Sparse self-attention is performed within each block to extract relevant features, while sparse cross-attention is applied between blocks only when they are projected onto the same voxel.

Future research will focus on further improving the network's feature extraction capabilities from sparse point clouds

and exploring the integration of camera images to enhance the representation of LiDAR point clouds across multiple observation planes. These advancements are expected to significantly improve the accuracy and efficiency of 3-D object detection based on LiDAR point clouds.

REFERENCES

[1] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[2] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11040–11048.

[3] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9277–9286.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon et al., Eds., Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[6] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[7] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2020.

[8] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10529–10538.

[9] S. Shi et al., "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection," *Int. J. Comput. Vis.*, vol. 131, no. 2, pp. 531–551, Feb. 2023.

[10] J. Deng et al., "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1201–1209.

[11] P. An et al., "SP-det: Leveraging saliency prediction for voxel-based 3D object detection in sparse point cloud," *IEEE Trans. Multimedia*, vol. 26, pp. 2795–2808, 2024.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Dec. 2021.

[13] A. Chen et al., "TensoRF: Tensorial radiance fields," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 333–350.

[14] C. Reiser et al., "MERF: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–12, Aug. 2023.

[15] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[16] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11784–11793.

[17] S. Deng, Z. Liang, L. Sun, and K. Jia, "VISTA: Boosting 3D object detection via dual cross-view spatial attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 8448–8457.

[18] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 5428–5437.

[19] Z. Zhou, X. Zhao, Y. Wang, P. Wang, and H. Foroosh, "CenterFormer: Center-based transformer for 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 496–513.

[20] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon et al., Eds., Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[21] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "VoxelNeXt: Fully sparse VoxelNet for 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21674–21683.

[22] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[23] J. Wang, S. Lan, M. Gao, and L. S. Davis, "InfoFocus: 3D object detection for autonomous driving with dynamic information modeling," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 405–420.

[24] G. Shi, R. Li, and C. Ma, "PillarNet: Real-time and high-performance pillar-based 3D object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 35–52.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[26] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[27] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1742–1749.

[28] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1951–1960.

[29] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, "Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 18953–18962.

[30] L. Fan, F. Wang, N. Wang, and Z.-X. Zhang, "Fully sparse 3D object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 351–363.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[32] D. Zhou et al., "IoU loss for 2D/3D object detection," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 85–94.

[33] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[34] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2023, *arXiv:2301.00493*.

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

[36] Q. Chen, L. Sun, E. Cheung, and A. L. Yuille, "Every view counts: Cross-view consistency in 3D object detection with hybrid-cylindrical-spherical voxelization," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 21224–21235.

[37] Q. Chen, L. Sun, Z. Wang, and A. Yuille, "Object as hotspots: An anchor-free 3D object detection approach via firing of hotspots," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 68–84.

[38] Y. Hu et al., "AFDetV2: Rethinking the necessity of the second stage for object detection from point clouds," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, 2022, pp. 969–979.

[39] X. Bai et al., "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 1090–1099.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[41] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3D object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 18442–18455.

[42] L. Tao, H. Wang, L. Chen, Y. Li, and Y. Cai, "Pillar3D-former: A pillar-based 3-D object detection and tracking method for autonomous driving scenes," *IEEE Trans. Instrum. Meas.*, early access, Dec. 1, 2024, doi: 10.1109/TIM.2023.3338662.

[43] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 2446–2454.

[44] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

Ziming Zhu is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China.

His current research interests include artificial intelligence, digital image and point cloud data processing, deep learning algorithms and applications, and computer vision. Specific research contents include camera-based, 4-D imaging radar, LiDAR single-modal and multimodal 3-D object detection, and semantic occupancy prediction.

Yu Zhu (Member, IEEE) is a Professor, Doctoral Supervisor, and Head of the Department of Electronic and Communication Engineering, School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. She serves as the Executive Director of Shanghai Image and Graphics Society. She has long been committed to artificial intelligence, image and video information processing, and deep learning algorithms and applications.

Kezhi Zhang was born in Ningbo, Zhejiang, China, in 2000. He received the B.S. degree in information engineering from the East China University of Science and Technology, Shanghai, China, in 2022, where he is currently pursuing the M.S. degree.

His research interests include computer vision based on deep learning.

Hangyu Li is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China.

His research interests primarily revolve around generative models, neural radiation fields, deep learning algorithms and applications, and computer vision. Within these areas, he focuses on topics such as AIGC, medical image reconstruction using generative models, and novel perspective synthesis of medical images based on neural radiation fields.

Xiaofeng Ling received the B.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2012, respectively.

From 2013 to 2015, he served as the Director of research and development in a start-up company. He is currently an Associate Professor with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai. His research interests include wideband array signal processing, wireless communications, and MIMO technique.