

M2CNet: LiDAR 3D Semantic Segmentation Based on Multi-Level Multi-View Cross-Attention Fusion for Autonomous Vehicles

Mu Zhou¹, Senior Member, IEEE, Rui Xiang¹, Nan Du¹, Wei He¹, and Zhao Jiang²

Abstract—Light Detection and Ranging (LiDAR) point cloud semantic segmentation is an essential task for autonomous vehicles, enabling precise environment perception for safe navigation. Existing point-based and voxel-based methods face limitations related to computational costs and quantization errors, especially dealing with large-scale point clouds. In contrast, 2D projection methods, including Range View and Bird's Eye View, have gained popularity for their balance of real-time performance and accuracy. However, most projection approaches overlook critical issues arising from projection, including 1) inter-class occlusion, 2) boundary-blurring, and 3) inadequate multi-scale feature extraction. To address these issues, we propose a Multi-level Multi-view Cross-attention Fusion Network (M2CNet), which fully leverages complementary information across different views. First, we employ parallel encoder-decoder networks to extract semantic information within a multi-task segmentation framework. Given the high correlation between views, we incorporate a Multi-view Feature Cross Fusion Module (MFCFM) to facilitate cross-view feature association, capturing potential complementarities to mitigate information loss from single-view projections. Then, we place a Multi-cascaded Enhanced Semantic Context Atrous Spatial Pyramid Pooling (MESCAPPP) module between the encoder and decoder to capture multi-scale contextual information, enhancing pixel relationships effectively. Finally, the Kernel Point Convolution (KPConv) is employed to refine point-wise features at the output level. The proposed method is evaluated on urban datasets SemanticKITTI and nuScenes (achieving mIoU scores of 65.7% and 78.9% respectively), the off-road dataset RELLIS-3D (44.6%), and

a real-world test set. Experimental results demonstrate that our method surpasses representative methods in both effectiveness and scene adaptability.

Index Terms—Autonomous driving, deep learning, multi-view fusion, point cloud, semantic segmentation.

I. INTRODUCTION

WITH the continuous advancement of autonomous driving technology, an efficient and accurate understanding of road traffic scenes is essential for precise navigation, informed decision-making, and safe control in complex dynamic environments [1], [2]. As a core component of scene understanding, the semantic segmentation provides fine-grained object labels, enabling the extraction of various information, such as object categories and shapes. This task has been widely applied in autonomous driving [3]. In terms of implementation, autonomous driving systems typically utilize camera sensors to capture 2D images or Light Detection and Ranging (LiDAR) sensors to acquire 3D point clouds. However, camera sensors are susceptible to changes in lighting, obstacles, and adverse weather conditions, which can lead to insufficient color and texture features in the collected images [4]. In contrast, LiDAR as an active sensor are mainly unaffected by environmental light variations and provide highly accurate distance measurements and 3D geometric information, demonstrating strong resistance to interference. Therefore, LiDAR has been widely deployed in autonomous driving navigation.

LiDAR point clouds are 3D data captured by laser scanners, comprising XYZ coordinates. However, location information alone is insufficient to distinguish between different objects. To address this, the point cloud semantic segmentation task assigns semantic labels (e.g., cars, persons, roads, buildings) to each point, enabling accurate segmentation and integration. In recent years, deep learning has brought revolutionary advancements to 3D computer vision tasks, significantly enhancing its applications in autonomous driving, including object detection [5], object classification [6], and semantic segmentation [7]. Numerous deep learning-based 3D point cloud semantic segmentation models have been proposed [8], [9]. Based on the encoding methods for point clouds, these deep learning methods can be categorized into three main types: point-based methods, voxel-based methods, and projection-based methods. Point-based methods [10], [11] perform well on small-scale

Received 13 January 2025; revised 15 July 2025; accepted 9 August 2025. Date of publication 18 August 2025; date of current version 13 February 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62101085, in part by Chongqing Natural Science Foundation Project under Grant CSTB2023NSCQ-LZX0014 and Grant CSTB2024NSCQ-LZX0123, in part by the Special Project of Chongqing Technology Innovation and Application Development (Major Project) under Grant CSTB2024TIAD-STX0035, and in part by the Postgraduate Scientific Research and Innovation Project of Chongqing under Grant CYS25444. The review of this article was coordinated by Dr. Weisong Wen. (Corresponding authors: Wei He; Zhao Jiang.)

Mu Zhou and Rui Xiang are with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: zhoumu@cqupt.edu.cn; s230101165@stu.cqupt.edu.cn).

Nan Du is with the Department of Computer Science and Technology, Tangshan Normal University, Tangshan 063000, China (e-mail: d220101005@stu.cqupt.edu.cn).

Wei He is with the School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: hew@cqupt.edu.cn).

Zhao Jiang is with the College of Civil Engineering, Zhejiang University of Technology, Hangzhou 310014, China (e-mail: jiangchao-0755@163.com).

The source code is publicly available at <https://github.com/Terminal-lidar/M2CNet>.

Digital Object Identifier 10.1109/TVT.2025.3599413

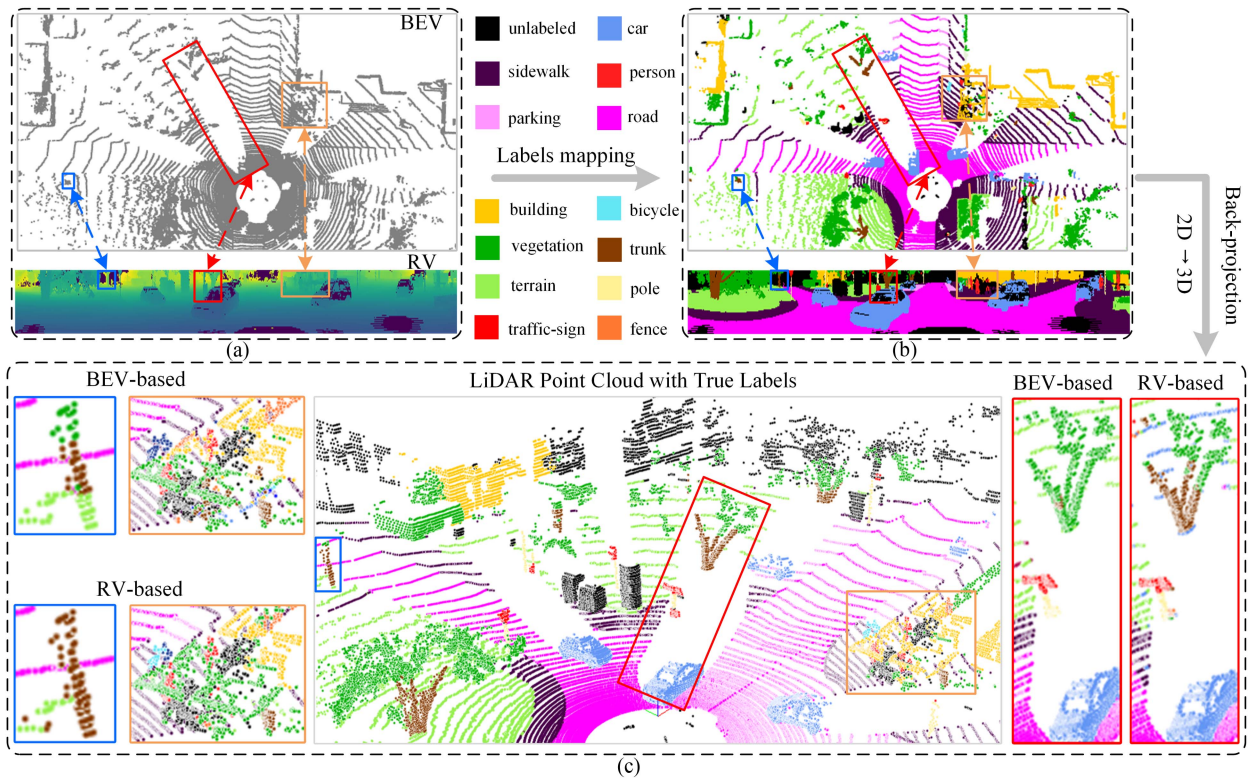


Fig. 1. Distribution of different target information captured by BEV and RV. (a) shows the original projected image, and (b) displays the image obtained after true semantic labels mapping. The sides of (c) show the regional 3D space prediction comparison of PolarNet [14] based on BEV and SalsaNext [15] based on RV. The blue boxes highlight vertically distributed objects that are more easily captured in RV. The red boxes indicate boundary-blurring between horizontally distributed classes in RV, which is avoided in BEV. The orange boxes show the limitations of a single view in multi-scale information representation, as it cannot simultaneously capture both the global structure and detailed features.

datasets but struggle with large-scale point clouds and complex road traffic, resulting in slow speeds, low accuracy, and poor robustness [12]. Voxel-based methods [13] convert point clouds into structured voxels, but suffer from redundancy and high computational complexity, leading to quantization errors and difficulty meeting real-time demands for autonomous driving. In contrast, projection-based methods map 3D point clouds to Bird's Eye View (BEV) or Range View (RV) via top-down or spherical projections, achieving good segmentation while maintaining real-time performance.

Due to differences in projection directions, RV and BEV capture distinct point cloud information, resulting in varying degrees of information loss across different categories. To highlight the differences, we compare the point cloud distributions in RV and BEV representations, focusing on areas with significant information loss for certain categories. As shown in the red boxes of Fig. 1(a) and (b), the compactness of the RV projection can cause horizontal occlusion and sensitivity to distance variations, leading to the loss of horizontal boundary information. In Fig. 1(c), the red box illustrates how this can result in boundary-blurring during the reprojection process, especially for small objects at greater distances. In contrast, BEV projects the point cloud from a top-down view, reducing object overlap and better distinguishing items in different regions, which helps mitigate occlusion and boundary-blurring. However, as shown in the blue boxes, vertical structural information may be lost due to perspective effects when mapping to the BEV grid, particularly

for sparse objects. Therefore, RV is preferred for small, vertically oriented objects (e.g., motorcycles, persons, and poles), while BEV is more effective for larger objects with significant spatial distribution in the XY plane (e.g., cars, buildings, and roads). Additionally, road scenes often contain entities with varying scales. Recognizing small-scale objects requires detailed information, while segmenting large-scale objects demands a network with a sufficiently large receptive field. As indicated in the orange box, the projection methods inevitably compress 3D spatial information and the inherently limited receptive field of the network causes fine details to be lost. Both methods thus face significant challenges in handling multi-scale context information.

Therefore, single-view projections suffer from information loss by single viewpoint and inadequate multi-scale context feature extraction. Due to the potential relevance of the information present in the different views, integrating data from multi-view projections is crucial to addressing the information loss caused by individual RV and BEV projections. To address these issues, we propose a novel Multi-level Multi-view Cross-attention Fusion Network (M2CNet), which improves the accuracy and adaptability of semantic segmentation by combining features from different views. The main contributions of this paper are as follows.

- 1) To improve point cloud target semantic segmentation, this paper proposes a novel dual-stream neural network with multi-level multi-view cross-attention fusion. By

integrating multi-level features, the model enhances the use of inter-class distribution information and overcomes the limitations of single-view projections.

- 2) To fully integrate information from different views, this paper introduces a Multi-view Feature Cross Fusion Module (MFCFM). This module improves feature representation and relationship by effectively sharing view-specific features through cross-attention, thereby enabling the learning of both local and long-range dependencies for a more comprehensive scene understanding.
- 3) To preserve more multi-scale detail in segmented entities, we propose a Multi-cascaded Enhanced Semantic Context Atrous Spatial Pyramid Pooling (MESCASPP) module. This module includes parallel convolutional branches with varying dilation rates and an Enhanced Semantic Context Module (ESCM), enabling dense pixel aggregation across multiple receptive fields and cascading context modules, thereby capturing richer contextual semantic information.

The remaining part of this paper is organized as follows. Section II summarizes the related works. Section III introduces the proposed methodology. Experimental results are presented in Section IV. Finally, Section V concludes this paper.

II. RELATED WORKS

A. Point-Based Methods

Point-based methods directly process raw point clouds by leveraging their inherent spatial topology. These methods can be broadly categorized into Multi-Layer Perceptron (MLP)-based and point-convolution-based approaches. Early MLP-based methods, such as PointNet [10], employ shared MLPs to extract per-point features and use global max pooling for scene-level abstraction. Although efficient, they ignore local geometric relationships. PointNet++ [11] addresses this limitation via hierarchical grouping and feature learning from local neighborhoods, but still suffers from limited efficiency and scalability. To further improve speed and local detail preservation, RandLA-Net [16] presents an efficient network structure that combines random point sampling with local feature aggregation strategies, improving point cloud processing efficiency and effectively retaining more geometric details. However, using K-Nearest Neighbors (KNN) to maintain local relationships between points can slow inference due to the overhead of k-dimensional tree construction. Point-convolution-based approaches have since emerged to enhance geometric representation. KPConv [17] introduces a kernel point convolution technique, where convolution weights are based on the Euclidean distance to kernel points, enhancing local feature extraction with spatially adaptive weights. SWCF-Net [18] introduces a similarity-weighted convolution to enhance local feature extraction and reduce overfitting, along with a lightweight attention module for global context modeling. Meanwhile, SIESEF-FusionNet [19] focuses on boundary refinement, enhancing spatial inter-correlation using inverse distance weighting and angular compensation, and introduces spatial-adaptive pooling for semantic embedding. While point-based methods offer precise local feature learning,

they face challenges in modeling long-range dependencies and maintaining efficiency on large-scale scenes.

B. Voxel-Based Methods

Voxel-based methods discretize point clouds into structured 3D grids, allowing the use of convolutional neural networks to exploit spatial continuity. These approaches can be categorized into regular voxel and sparse voxel strategies. Regular voxel methods such as VoxelNet [13] introduce voxel feature encoding to transform point clouds into feature representation vectors. However, due to the sparsity of point clouds, the algorithm's accuracy relies heavily on voxel density, leading to computational redundancy. LSK3DNet [20] tackles the high cost of large 3D kernels by employing spatial-wise dynamic sparsity and channel-wise pruning, enabling efficient large-receptive-field modeling with fewer parameters. Sparse voxel approaches aim to reduce redundancy by focusing computations only on non-empty voxels. SPVNAS [21] presents a sparse point-voxel convolution module designed specifically for small object recognition. It addresses the challenges of maintaining high-resolution representations with sparse convolutions and scaling to large-scale scenes. Cylinder3D [22] transforms point clouds into specialized cylindrical grids, incorporating asymmetric residual blocks and context modeling modules to match relevant features of scene objects and capture higher-order contextual information. PCSCNet [23] fuses point convolutions with sparse 3D convolutions to support real-time segmentation, but coarse voxelization can still induce spatial quantization errors. SVASeg [24] introduces sparse voxel-based multi-head attention, using hash-based neighbor lookup and region grouping to enlarge receptive fields efficiently. However, voxel-level locality may still hinder fine-geometry preservation in complex scenes. In summary, voxel-based methods preserve geometric and topological information in 3D point clouds, but sparsity, loss of detail, and discretization artifacts limit their performance. Additionally, the high computational cost of 3D convolutions makes it challenging to balance real-time performance with accuracy.

C. Single-View Projection-Based Methods

Projection-based methods transform 3D point clouds into 2D images, leveraging effective 2D Convolutional Neural Network (CNN) to recognize pixel-level semantic information. This approach significantly reduces computational complexity while maintaining spatial information, and is typically divided into RV and BEV. In RV projection, RangeNet++ [12] employs spherical range-based projection, proposing a KNN post-processing method to refine point cloud labels. SalsaNext [15] introduces a context module based on residual dilated convolutions for a larger receptive field. Additionally, it incorporates a Bayesian neural network to estimate uncertainty, enhancing both network performance and reliability. LENet [25] presents a multi-scale convolution attention module that extracts features at various scales, improving segmentation accuracy through an attention mechanism. More recent approaches like SwinURNet [26] adopt

range images and introduces a hybrid CNN-Transformer architecture for unstructured road segmentation. While it improves boundary prediction in challenging scenarios, its reliance on a single RV projection limits geometric completeness and robustness to occlusion. RangeViT [27] further explores the application of pre-trained vision transformers to range images, showing that such models can achieve strong performance with minimal adaptation.

In BEV projection, SalsaNet [28] is commonly used for segmenting roads and vehicles in road scenes. PolarNet [14] quantifies point clouds via polar coordinate BEV, making them better suited for LiDAR scan representations and addressing the imbalanced spatial distribution in point clouds. It designs circular convolutions to align the network's attention with the long-tail distribution of points along the radial axis. DGPolarNet [29] employs dynamic graph convolution networks to capture long-range dependencies and local context. It dynamically builds a graph from the input point cloud, leveraging multi-scale features and graph convolutions to extract spatial relationships and discriminative features. PC-BEV [30] proposes an efficient BEV-only fusion strategy by combining polar and Cartesian grids. This approach reduces the need for computationally expensive point-level fusion, but may suffer from reduced adaptability in occluded or sparsely scanned regions due to the absence of cross-view cues. In general, RV-based methods may encounter boundary-blurring issues during reprojection and are sensitive to changes in range. Conversely, BEV-based methods are sensitive to height variations, often leading to compound errors and suboptimal performance in detecting vertical structures, thereby limiting the efficacy of these methods.

D. Multi-View Projection-Based Methods

To address the limitations of single-view projections, recent studies have explored multi-view fusion frameworks to leverage their complementary spatial and contextual information. TORNADO-Net [31] uses BEV features, extracted through a pillar projection learning module, and projects them onto RV. It incorporates a global context module to process the fused images and extract rich semantic features. AMVNet [32] employs a post-sampling strategy, guided by assertions, to select uncertain points, extract additional point features, and perform fusion between the two views. MPF [33] analyzes RV and BEV projections using two parallel lightweight 2D CNNs, merging the segmentation results from both views through a soft voting mechanism at the output stage. CPGNet [34] builds a cascaded fusion framework that integrates point-wise, RV, and BEV representations, enabling hierarchical refinement of semantic predictions. GFNet [35] designs a geometric flow module applied during the decoding phase across multiple feature maps of different scales, facilitating bidirectional alignment and propagation of geometric information between RV and BEV projections. BALViT [36] introduces a multi-view fusion framework that integrates frozen 2D vision transformers with range-view LiDAR features and enhances them via cross-attention from a BEV branch. During inference, an output-level fusion strategy based on confidence thresholding is adopted to selectively merge

predictions. However, most existing approaches concentrate on the input or output stages, neglecting mid-level feature interactions and deep geometric alignment. Moreover, in dynamic road environments with diverse object scales, these methods often lack the ability to extract robust multi-scale features, which restricts their effectiveness in fine-grained segmentation tasks.

III. METHODOLOGY

In this section, we design a point cloud semantic segmentation network named M2CNet. It builds upon the baseline networks of SalsaNet and ResNet34 [37] for RV stream and BEV stream respectively, enabling parallel extraction of semantic information. As shown in Fig. 2, we extract view-specific features in parallel and use the MFCFM to enhance the feature representations in the RV and BEV streams. However, directly using encoder-decoder multi-level feature maps often captures weak contextual dependencies due to limited receptive fields, which are insufficient for effective multi-scale feature aggregation. To address this, we introduce the MESCASPP module, which expands the receptive field and captures multi-scale contextual information. Finally, to mitigate reprojection errors resulting from the loss of original point cloud data during projection, we incorporate a point-based post-processing method into the end-to-end training process for 3D predictions.

A. Semantic Segmentation Backbone

1) *RV Stream*: Given a LiDAR point cloud frame with N points, each point is represented by (x, y, z, r) , where (x, y, z) denotes the Cartesian coordinates relative to the LiDAR sensor and r is the reflectance intensity. We project the 3D data into a dense RV image I_{RV} using spherical coordinate transformation. This process maps each 3D point to image coordinates (u, v) as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2} (1 - \arctan(y, x)\pi^{-1}) W_{rv} \\ (1 - (\arcsin(z, d) - f_{down}) f^{-1}) H_{rv} \end{pmatrix} \quad (1)$$

Here, $f = f_{up} - f_{down}$ denotes the vertical field of view, where $f_{up} (> 0)$ and $f_{down} (< 0)$ are the maximum upward and downward pitch angles of the LiDAR beams. The distance to the LiDAR center is given by $d = \sqrt{x^2 + y^2 + z^2}$. W_{rv} and H_{rv} represent the width and height of the RV image, respectively. We encode the values (x, y, z, r, d) into five channels, forming an input feature map $I_{RV} \in \mathbb{R}^{H_{rv} \times W_{rv} \times 5}$. As shown in Fig. 1, RV-based segmentation exposes vertically continuous but horizontally varying object boundaries, often leading to the loss of channel-specific details during direct encoding. To address this, we introduce a Residual Coordinate Channel Spatial Attention (RCCSA) module, built upon Coordinate Attention (CA) [38], to enhance semantic representation by capturing directional position and long-range spatial dependencies. As illustrated in Fig. 3, the input features are first processed by CA, followed by channel-spatial attention, and then fused via residual addition.

2) *BEV Stream*: To mitigate the long-tail distribution in Cartesian coordinates, especially caused by distant sparse points and near-field clutter, we adopt the polar coordinate projection strategy. The (x, y, z) coordinates are transformed to

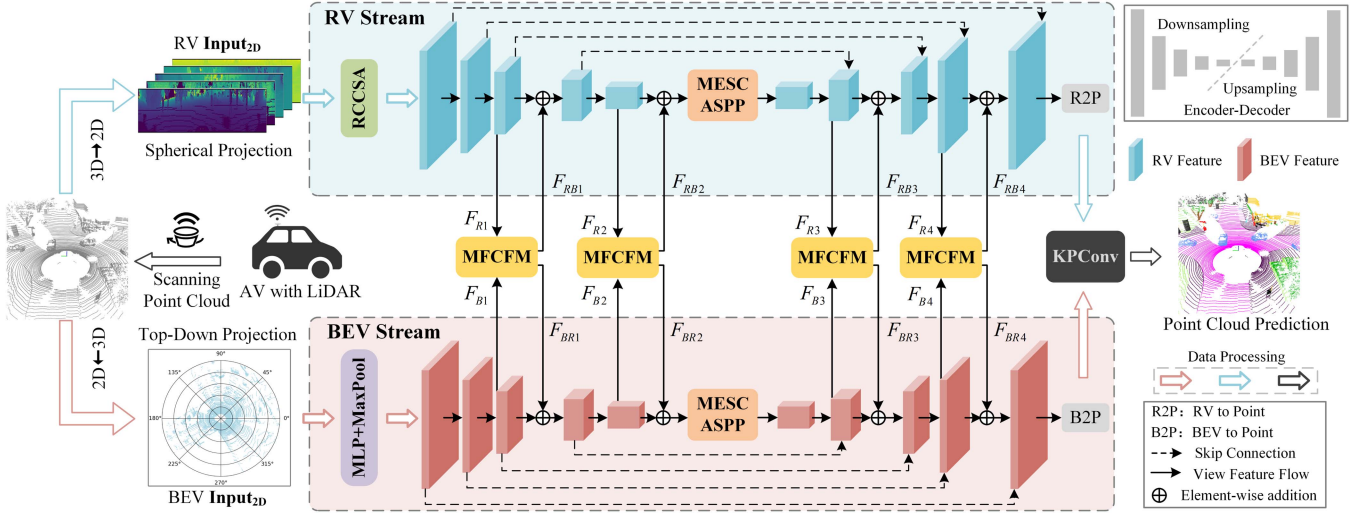


Fig. 2. Overall architecture of the M2CNet. It consists of four main components: 1) the semantic segmentation backbone, combining parallel RV and BEV streams with the encoder-decoder structures. 2) the multi-view feature cross fusion through the MFCFM during the multi-level encoding and decoding phases. ($F_{R1}, F_{R2}, F_{R3}, F_{R4}$) and ($F_{B1}, F_{B2}, F_{B3}, F_{B4}$) represent different levels of RV features and BEV features respectively. ($F_{RB1}, F_{RB2}, F_{RB3}, F_{RB4}$) and ($F_{BR1}, F_{BR2}, F_{BR3}, F_{BR4}$) represent corresponding fusion features. 3) enhanced semantic context perception by using the MESCASPP between the encoder and decoder. 4) KPCConv-based post-processing is employed to further refine the output point cloud features.

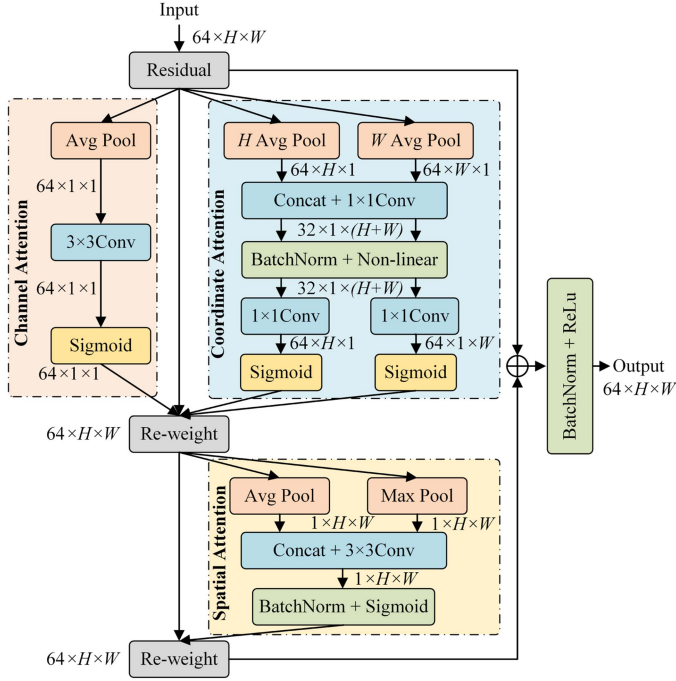


Fig. 3. Architecture of the RCCSA Module.

(ρ, ϕ, z) as:

$$\begin{pmatrix} \rho \\ \phi \\ z \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \arctan(y, x) \\ z \end{pmatrix} \quad (2)$$

where ρ and ϕ represent the polar radius and azimuthal angle, respectively. Points are assigned to a BEV grid I_{BEV} using

normalized polar coordinates:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{\rho - \rho_{\min}}{\rho_{\max} - \rho_{\min}} \times W_{bev} \\ \frac{\phi - \phi_{\min}}{\phi_{\max} - \phi_{\min}} \times H_{bev} \end{pmatrix} \quad (3)$$

Each grid encodes 9 features: [$\rho, \phi, z, \Delta\rho, \Delta\phi, \Delta z, x, y, r$], where the Δ terms denote relative offsets from the grid center. This encoding can enhance positional awareness and improve the capture of local geometric structures. After encoding, the input feature map is obtained with a dimensional size $H_{bev} \times W_{bev} \times 9$. Specifically, a shared MLP captures high-dimensional features, while max pooling aggregates point information within each grid into a fixed-length representation that highlights key features.

B. Multi-View Feature Cross Fusion Module

Given the strong correlation between RV and BEV semantic segmentation tasks, it is essential to exploit the complementary relationship between low-level geometric features and high-level semantic features. Inspired by the excellent real-time performance of the Swin-Transformer [39] in downstream tasks, we propose the MFCFM. This module integrates multi-view feature information within the multi-level intermediate layers of the encoder-decoder architecture, addressing information loss from different projection methods and enhancing the complementary benefits across views. Fig. 4(a) illustrates the design of the proposed MFCFM.

To fuse features from different views, MFCFM utilizes the geometric relationships between views to associate feature maps of varying scales. By utilizing the connection between the view and the original 3D point cloud, the RV and BEV features, denoted as F_R and F_B respectively, are first input into the Cross-view Association Module (CAM). This module uses point clouds $\mathcal{P} \in \mathbb{R}^{N \times 4}$ as an intermediate medium, establishing cross-view

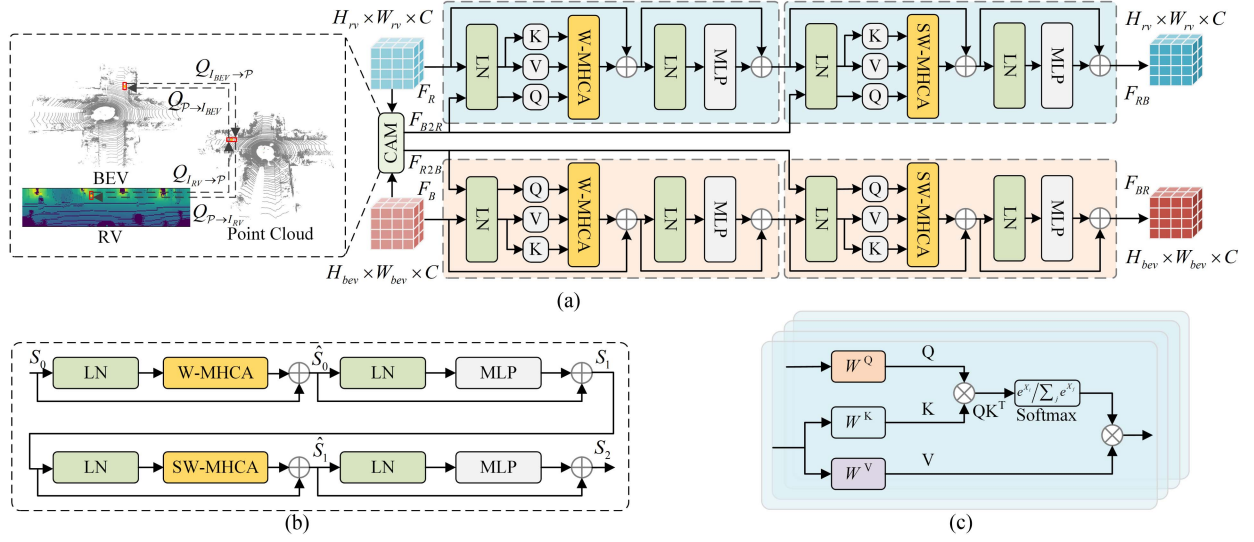


Fig. 4. Overall architecture of the MFCFM. (a) illustrates the main algorithm flowchart of the MFCFM. The CAM associates view features across views, followed by the application of shifted window-based cross-attention for deep fusion of the view features. (b) presents the flowchart of the shifted window multi-head cross-attention mechanism. (c) shows the flowchart for calculating multi-head cross-attention scores by using Q, K, and V.

associations through the point-wise image coordinate matrix $Q_{I \rightarrow P}$ generated during the point cloud projection process, as well as the pixel-wise point cloud index matrix $Q_{P \rightarrow \hat{I}}$. $(I, \hat{I}) \in \{(I_{BEV}, I_{RV}), (I_{RV}, I_{BEV})\}$ represents the associated image and the target image respectively. $Q_{I \rightarrow P} \in \mathbb{Z}^{N \times 2}$ is shown below.

$$Q_{I \rightarrow P} = [m_0 \ \cdots \ m_{N-1}]^T = \begin{bmatrix} u_0 & \cdots & u_{N-1} \\ v_0 & \cdots & v_{N-1} \end{bmatrix}^T \quad (4)$$

where $\{m_i | 0 \leq i \leq N-1\}$ denotes the associated image coordinate value (u_i, v_i) for each point. $Q_{P \rightarrow \hat{I}} \in \mathbb{Z}^{H \times W \times 1}$ is represented as

$$Q_{P \rightarrow \hat{I}} = \begin{bmatrix} n_{(0,0)} & \cdots & n_{(0,W-1)} \\ \vdots & \ddots & \vdots \\ n_{(H-1,0)} & \cdots & n_{(H-1,W-1)} \end{bmatrix} \quad (5)$$

where $\{n_{(i,j)} | 0 \leq i \leq H-1, 0 \leq j \leq W-1\} \in [0, N]$ represents the original point cloud index corresponding to each image grid. H and W denote the height and width of the associated image $\hat{I}$ respectively. By combining $Q_{I \rightarrow P}$ and $Q_{P \rightarrow \hat{I}}$, M2CNet can obtain the cross-view association matrix $Q_{I \rightarrow \hat{I}} \in \mathbb{Z}^{H \times W \times 2}$, such that

$$Q_{I \rightarrow \hat{I}} = \begin{bmatrix} m_{n_{(0,0)}} & \cdots & m_{n_{(0,W-1)}} \\ \vdots & \ddots & \vdots \\ m_{n_{(H-1,0)}} & \cdots & m_{n_{(H-1,W-1)}} \end{bmatrix} \quad (6)$$

Based on this, the features of the associated image $\hat{I}$ are sampled through the cross-view association matrix to align the feature grids between the two views. After the cross-view association, the view features F_R and F_B are transformed into F_{B2R} and F_{R2B} respectively.

To effectively fuse view features and make better use of complementary information, we introduce a Multi-head Cross-attention (MHCA) mechanism. This approach enhances the fusion of feature information by exchanging the Query (Q), Key (K), and Value (V) from different data sources. As shown in Fig. 4(a), given the input feature pair (F_R, F_{B2R}) or (F_B, F_{R2B}) , the K-V and Q for the Windows Multi-head Cross-attention (W-MHCA) layer and the Shifted Windows Multi-head Cross-attention (SW-MHCA) layer are obtained through Layer Normalization (LN). Unlike self-attention mechanisms, the cross-attention mechanism allows direct interaction between features from different views, capturing interdependencies and improving feature fusion quality. Additionally, the shifted windows strategy confines attention computation to non-overlapping local windows, achieving higher efficiency. Through the MFCFM, the feature representations are denoted as F_{RB} and F_{BR} . As illustrated in Fig. 4(b), the attention fusion process is expressed as

$$\begin{cases} \hat{S}_0 = W\text{-MHCA}(LN(S_0)) + S_0 \\ S_1 = MLP(LN(\hat{S}_0)) + \hat{S}_0 \\ \hat{S}_1 = SW\text{-MHCA}(LN(S_1)) + S_1 \\ S_2 = MLP(LN(\hat{S}_1)) + \hat{S}_1 \end{cases} \quad (7)$$

where $\hat{S}_0$ and S_1 represent the residual outputs after the W-MHCA module and the MLP module respectively, while $\hat{S}_1$ and S_2 denote the residual outputs after the SW-MHCA module and the MLP module respectively.

Fig. 4(c) illustrates the MHCA computation of the module, where four independent attention heads perform attention calculations simultaneously. In this context, (W^Q, W^K, W^V) represents the learnable parameter matrices corresponding to (Q, K, V). The computation process is expressed by the following

Algorithm 1: Multi-view Feature Cross Fusion from Associated-view I to Target-view $\hat{I}$:

Input: Point-wise Coordinate matrix $Q_{I \rightarrow \mathcal{P}} \in \mathbb{Z}^{N \times 2}$;
Pixel-wise Index matrix $Q_{\mathcal{P} \rightarrow \hat{I}} \in \mathbb{Z}^{H \times W \times 1}$;
Target-view features $F_{\hat{I}} \in \mathbb{R}^{B \times C \times H \times W}$;
Associated-view features $F_I \in \mathbb{R}^{B \times C \times H_I \times W_I}$
Output: Fused feature $F_{out} \in \mathbb{R}^{B \times C \times H \times W}$

- 1: **Step 1: Cross-view Association Matrix Construction**
- 2: $Q_{I \rightarrow \hat{I}} \leftarrow$ initialize matrix with shape $\mathbb{Z}^{H \times W \times 2}$
// View Space Align
- 3: **foreach** (i, j) *where* $Q_{\mathcal{P} \rightarrow \hat{I}}(i, j) \geq 0$ **do**
- 4: $Q_{I \rightarrow \hat{I}}(i, j, :) \leftarrow Q_{I \rightarrow \mathcal{P}}[Q_{\mathcal{P} \rightarrow \hat{I}}(i, j)]$
- 5: **Step 2: Feature Resampling from I to $\hat{I}$**
// Coordinate Adjust
- 6: $Q_{I \rightarrow \hat{I}} \leftarrow \text{flip}(Q_{I \rightarrow \hat{I}})$
// Resolution Align
- 7: $Q_{I \rightarrow \hat{I}}^{scaled} \leftarrow \text{interpolate}(Q_{I \rightarrow \hat{I}}, (H, W))$
- 8: $F_{warp} \leftarrow \text{grid_sample}(F_I, Q_{I \rightarrow \hat{I}}^{scaled})$
- 9: **Step 3: Feature Fusion**
- 10: $F_{cat} \leftarrow \text{Concat}(F_{\hat{I}}, F_{warp})$
- 11: $F_{fused} \leftarrow \text{Conv3} \times 3(F_{cat})$
- 12: $F_{norm} \leftarrow \text{LayerNorm}(F_{fused})$
// Dual-stage Cross Attention Fusion
- 13: $F_{tr} \leftarrow \text{SW-MHCA}(\text{W-MHCA}(F_{\hat{I}}, F_{norm}), F_{norm})$
- 14: $F_{out} \leftarrow F_{\hat{I}} \oplus F_{tr}$
- 15: **return** F_{out}

equations.

$$\text{head}_i(Q_i, K_i, V_i) = \text{Softmax} \left(\frac{Q_i K_i^T}{\sqrt{d_k}} \right) V_i, \quad (8)$$

$$\text{MHCA}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \text{head}_3, \text{head}_4) \quad (9)$$

where d_k denotes the scaling factor. The outputs of the individual heads are concatenated and then element-wise added to the original features to enrich their semantic information. Furthermore, the introduction of the MHCA mechanism offers the advantage of adaptively and dynamically adjusting feature weights, enabling the model to understand and capture the underlying relationships between the two views. The pseudo-code for the MFCFM is provided in Algorithm 1.

C. Multi-Cascaded Enhanced Semantic Context Atrous Spatial Pyramid Pooling

For road environment perception, segmentation networks need to refine pixel features of projected images and focus on the coherence of contextual semantics. Specifically, the segmentation network first needs a large receptive field due to the wide field of view in road scenes. Second, entities in road environments often exhibit multi-scale characteristics, and there is a similarity among specific categories. To address these issues and further enhance deep feature extraction in point cloud semantic segmentation tasks, we propose the MESCASPP module, as detailed in Fig. 5.

In the MESCASPP module, we utilize the output from the last layer of the dual-branch encoder as the input with the number of input channels 512. For the dilated convolutions on the parallel branches, the receptive field size RF_s of a dilated convolution with an expansion rate R and a kernel size K is expressed as

$$RF_s = (K - 1) \times (R - 1) + K \quad (10)$$

When dilated convolutions with the same expansion rate but kernel sizes K_1 and K_2 are cascaded, the equivalent receptive field corresponding to the combined kernel size is given by

$$K = K_1 + K_2 \quad (11)$$

From (10) and (11), it can be seen that when the kernel size is fixed, the relationship between R and RF_s is directly proportional. However, due to the sparse sampling of dilated convolutions, the correlation between distant pixels diminishes, making it easier to overlook small target objects during information transmission. Therefore, to aggregate object information in a multi-scale context and reduce the loss of multi-scale object information, we use four parallel dilated convolution branches. Each branch sequentially cascades two 3×3 dilated convolutions with expansion rates (3, 6, 12, 18), further enhancing the network's receptive field to capture denser pixel information.

By considering the low correlation of multi-scale and multi-category contextual dependencies, each dilated convolution feature maintains the same spatial resolution but differs in pixel-receptive fields. To address this issue, we embed the ESCM into each parallel branch of the dilated convolutions, as illustrated in Fig. 6. Specifically, by fusing the feature maps from each convolutional branch, this module adaptively weights the multi-scale feature maps by generating attention maps through global and local mechanisms. The output of the ESCM can be expressed as

$$F_{out} = F_{in} \odot F_0 \odot F_1 \quad (12)$$

where F_{in} and F_{out} correspond to the input and output feature maps respectively. $\odot$ denotes the matrix multiplication of feature maps. F_0 and F_1 represent the global and local attention maps, with the generation process expressed as

$$F_0 = \text{Sigmoid}(\text{BN}(\text{Conv}(\text{Pooling}(F_{in})))) \quad (13)$$

$$F_1 = \text{Sigmoid}(\text{BN}(\text{Conv}(F_{in}))) \quad (14)$$

where Pooling refers to global average pooling. Conv denotes the 1×1 convolution. BN represents the batch normalization function. Sigmoid is the activation function.

D. Post-Processing

After feature-level fusion during the multi-view encoding and decoding phase, M2CNet further performs output-level processing of point cloud features to improve segmentation accuracy. We transform the decoder outputs from 2D images to 3D point cloud features using R2P and B2P, as illustrated in Fig. 7. Specifically, we apply Bilinear Interpolation (BI) to obtain feature maps corresponding to the input size, with dimensions $H \times W \times C_{num}$ and C_{num} equal to the number of classes. Then, based on the corresponding projection relationship, the 2D features are restored to 3D point cloud features by Inverse

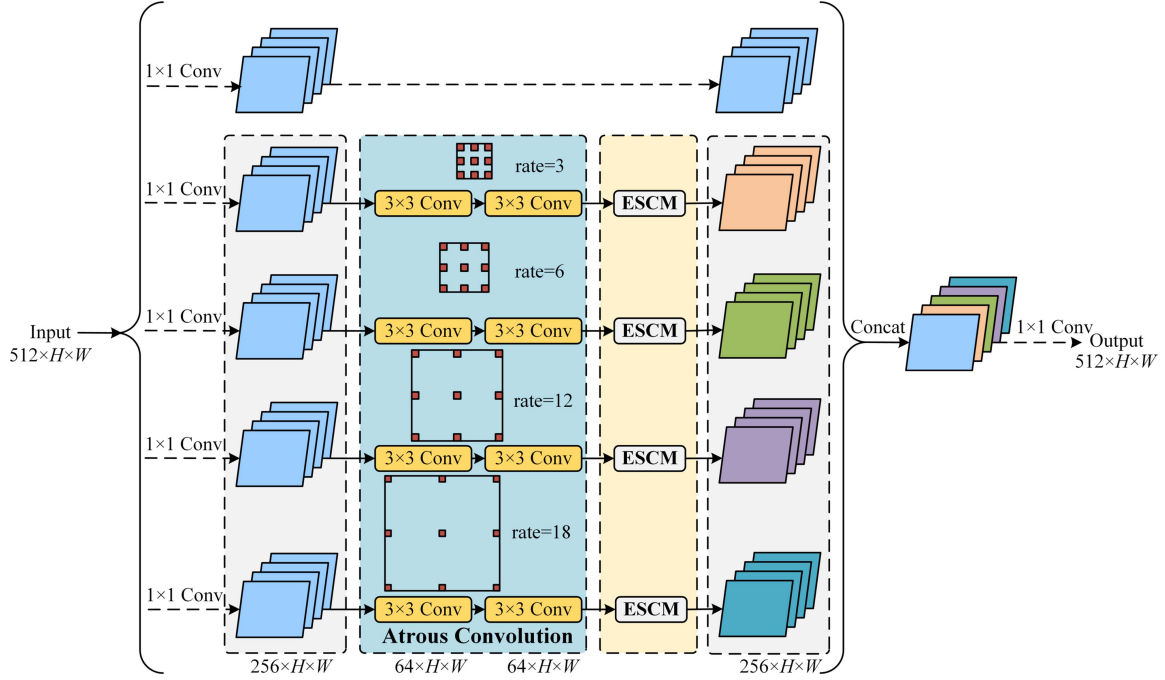


Fig. 5. Architecture of the MESCASPP Module. It consists of parallel branches with cascaded 3×3 dilated convolutions (rates: 3, 6, 12, 18) to expand the receptive field and capture dense multi-scale features. The ESCM aggregates pixel-level semantic information, while 1×1 convolution adjust channel dimensions without altering feature map size.

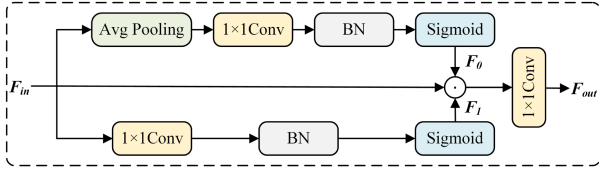


Fig. 6. Architecture of the ESCM.

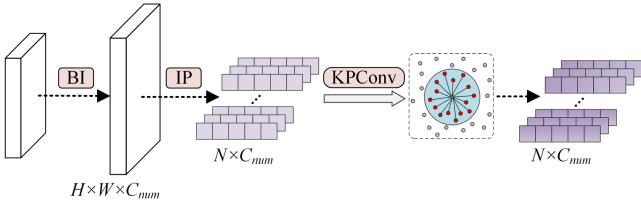


Fig. 7. Architecture of the R2P/B2P.

Projection (IP). Finally, KPConv is utilized to achieve the final 3D predictions for each branch. The specific implementation process is as follows.

KPConv applies convolution directly on point clouds by aggregating features from neighboring points using spatially defined kernel points. Given a center point $x \in \mathcal{P} \subset \mathbb{R}^{N \times 3}$ and its feature $f \in \mathcal{F} \subset \mathbb{R}^{N \times D_{in}}$, the convolution is defined as:

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}} g(x_i - x) f_i \quad (15)$$

where $\mathcal{N} = \{x_i \in \mathcal{P} \mid \|x_i - x\| \leq r\}$ is the set of neighbors within radius r , f_i is the feature of neighbor x_i , and $g(\cdot)$ is the kernel function assigning weights based on spatial offsets.

The input offset is denoted by $e_i = x_i - x \in \mathbb{R}^3$, and $\{e_i\} \subset S_r^3$ represents the relative positions within a spherical domain S_r^3 . A set of K learnable kernel points $\{\hat{x}_k \in \mathbb{R}^3\}_{k=1}^K \subset S_r^3$ defines regions of influence. Each kernel point is associated with a weight matrix $W_k \in \mathbb{R}^{D_{in} \times D_{out}}$ to transform the feature dimension from D_{in} to D_{out} . This structure allows us to preserve local geometric information and adapt weights spatially, making it well-suited for point-based 3D feature learning.

$$g(e_i) = \sum_{k < K} \max \left(0, 1 - \frac{\|e_i - \hat{x}_k\|}{\sigma} \right) W_k \quad (16)$$

where $\max(\cdot)$ represents the linear relationship between e_i and $\hat{x}_k$, with the correlation inversely proportional to the distance between them. σ denotes the association radius of the kernel points. By interacting with neighboring points through the application of the kernel point weights, M2CNet can fully exploit the local geometric features of the point cloud to enhance the expression of point features.

E. Loss Function

In the task of semantic segmentation for vehicle-mounted LiDAR point clouds in autonomous driving scenarios, two main issues arise, i.e., class imbalance and optimization of Intersection over Union (IoU). To mitigate the class imbalance issue, this paper employs the Weighted Cross Entropy (WCE) optimization function, defined as

$$\mathcal{L}_{wce}(y, \hat{y}) = - \sum_c \alpha_c p(y_c) \log(p(\hat{y}_c)) \quad (17)$$

where c represents the class, y_c and $\hat{y}_c$ denote the ground truth and predicted labels respectively, f_c indicates the frequency of the c th class, and $\alpha_c = 1/\sqrt{f_c}$ represents the weight for the c th class. Furthermore, we employ the widely used Lovasz-Softmax loss function [40] to optimize the IoU values. The function is defined as

$$\mathcal{L}_{ls} = \frac{1}{|C|} \sum_{c \in C} \overline{\Delta_{J_c}}(e(c)), \quad e_i(c) = \begin{cases} 1 - h_i(c), & c = y_i(c) \\ h_i(c), & \text{otherwise} \end{cases} \quad (18)$$

where $|C|$ represents the total number of classes, $\overline{\Delta_{J_c}}$ defines the Lovasz extension of the Jaccard index, and $h_i(c) \in [0, 1]$ and $y_i(c) \in \{-1, 1\}$ denote the predicted label and ground truth label for the i th pixel of the c th class respectively.

M2CNet adopts a weighted combination of three loss terms to enable end-to-end training. Let $Loss_F$, $Loss_R$, and $Loss_B$ denote the losses from the fused output, RV branch, and BEV branch, respectively. The corresponding predictions are P_F , P_R , and P_B , with P_T as the ground truth. The loss terms are defined as follows:

$$Loss_F = \mathcal{L}_{wce}(P_F, P_T) + \mathcal{L}_{ls}(P_F, P_T) \quad (19)$$

$$Loss_R = \mathcal{L}_{wce}(P_R, P_T) + \mathcal{L}_{ls}(P_R, P_T) \quad (20)$$

$$Loss_B = \mathcal{L}_{wce}(P_B, P_T) + \mathcal{L}_{ls}(P_B, P_T) \quad (21)$$

To optimize the proposed model, the total loss function is formulated as

$$Loss = \alpha Loss_F + \beta Loss_R + \lambda Loss_B \quad (22)$$

where α , β , and λ are the weights of the corresponding loss functions, which are set as 2, 1, and 1 respectively to balance the model's overall loss.

IV. EXPERIMENTS

A. Dataset and Metrics

SemanticKITTI dataset [41] is a large-scale benchmark for point cloud semantic segmentation, collected using a Velodyne HDL-64E LiDAR in diverse real-world driving scenes. It contains 43,551 annotated frames across 22 sequences, with 19,130 frames for training, 4,071 for validation, and 20,351 for testing, covering 19 semantic classes. **nuScenes** [42] is a large-scale autonomous driving dataset containing 1,000 scenes captured with a Velodyne HDL-32E LiDAR. It provides 40,157 annotated point cloud frames across 16 semantic classes, including 28,130 for training, 6,019 for validation, and 6,008 for testing. **RELLIS-3D** [43] is an unstructured outdoor dataset covering complex off-road environments such as brush, sparse trees, and grasslands. It contains 13,556 LiDAR frames collected with a 64-line Ouster OS1 sensor, split into 7,800 for training, 2,413 for validation, and 3,343 for testing, with annotations for 14 semantic classes.

To further evaluate the proposed method in real-world scenarios, we collect 2,000 frames of LiDAR data in campus road environments using a modular experimental platform equipped with a Velodyne VLP-16, onboard computation, and communication modules (Fig. 8). This setup supports data acquisition and algorithm validation in open environments.

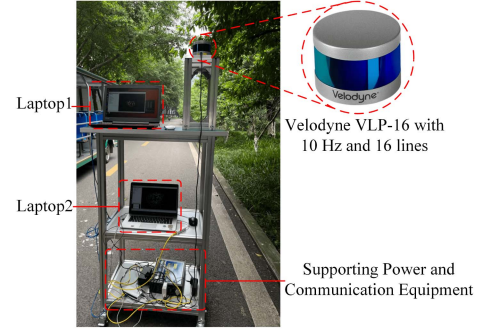


Fig. 8. Experimental platform with a Velodyne VLP-16 LiDAR mounted on a mobile platform, used for outdoor point cloud data collection.

In this paper, we employ the IoU as an evaluation metric to measure the disparity between the predicted segmentation results and the ground truth labels. Additionally, we employ the mean IoU (mIoU) to evaluate overall performance. The calculation formula is expressed as

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \quad (23)$$

$$mIoU = \frac{1}{C} \sum_{c=1}^C IoU_c \quad (24)$$

where C represents the total number of classes, and TP_c , FP_c , and FN_c denote the true positives, false positives, and false negatives for the category c respectively.

B. Implementation Details

All experiments in this paper are conducted by using the PyTorch deep learning framework on Ubuntu 20.04, equipped with two NVIDIA A40 GPUs, each with 48 GB of memory. The RV size is set as 64×2048 (SemanticKITTI and RELLIS-3D) or 32×1024 (nuScenes), while the BEV size is set as 480×320 . Data augmentation is applied during training with specific settings. Up to 10% of points are randomly dropped ($\in [0, 0.1]$), and the (x, y, z) coordinates are scaled by a factor from $[0.9, 1.1]$. For the RV branch, augmentations include horizontal flipping (along the y -axis), random translation ($x \in [-5, 5]$, $y \in [-3, 3]$, $z \in [-1, 1]$), and spatial rotation. The BEV branch additionally uses polar rotation (0° - 360°) and random flipping along the x , y , or both axes. The network model is trained for 150 epochs with the batch size 4. In the training, to accelerate the convergence of the network model, we employ Stochastic Gradient Descent (SGD) as the optimizer, with weight decay and momentum set as 0.0001 and 0.9 respectively. The initial learning rate is set as 0.001, and we utilize cosine annealing to gradually decrease it.

C. Comparative Experimental Validation

1) *Results on SemanticKITTI*: Table I presents the results of these quantitative comparisons. The results demonstrate that

TABLE I
QUANTITATIVE COMPARISON RESULTS OF FPS (Hz) AND mIoU (%) ON THE SEMANTICKITTI TEST SET

Methods		car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mIoU (%)	FPS (Hz)
Point	PointNet [10]	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6	2
	PointNet++ [11]	53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1	0.1
	RandLA-Net [16]	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7	53.9	1.3
	KPConv [17]	96.0	30.2	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	90.5	64.2	84.8	69.2	69.1	56.4	47.4	58.8	0.8
	SWCF-Net [18]	94.6	51.8	46.9	34.3	37.3	59.4	61.0	29.9	91.6	58.5	75.3	31.2	89.2	60.5	82.2	62.7	67.8	54.0	61.6	60.5	-
Voxel	SPVNAS [21]	97.2	50.6	50.4	56.6	58.0	67.4	67.1	50.3	90.2	67.6	75.4	21.8	91.6	66.9	86.1	73.4	71.0	64.3	67.3	67.0	4
	Cylinder3D [22]	97.1	67.6	64.0	59.0	58.6	73.9	67.9	36.0	91.4	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6	67.8	7
	SVASeg [23]	96.7	56.4	57.0	49.1	56.3	70.6	67.0	15.4	92.3	65.9	76.5	23.6	91.4	66.1	85.2	72.9	67.8	63.9	65.2	65.2	9
	PCSCNet [24]	95.7	48.8	46.2	36.4	40.6	55.5	68.4	55.9	89.1	60.2	72.4	23.7	89.3	64.3	84.2	68.2	68.1	60.5	63.9	62.7	-
Single-view	PolarNet [14]	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5	54.3	16
	PC-BEV [30]	96.5	66.5	57.7	37.6	48.2	62.5	68.0	49.1	93.0	73.5	79.8	32.1	92.1	69.2	85.4	70.0	70.0	62.4	63.0	67.2	-
	SalsaNext [15]	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1	59.5	28
	LENet [25]	93.9	57.0	51.3	44.3	44.4	66.6	64.9	36.0	91.8	68.3	76.9	30.5	91.2	66.0	83.7	68.3	67.8	58.6	63.2	64.5	26
	RangeNet++ [12]	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	20.2	26.3	52.2	16
	SwinURNet [26]	93.0	45.9	45.1	45.2	42.3	63.7	63.0	27.1	90.8	66.4	74.7	28.6	90.5	64.1	82.2	65.0	66.4	56.3	61.2	61.6	25
	RangeViT [27]	95.4	55.8	43.5	29.8	42.1	63.9	58.2	38.1	93.1	70.2	80.0	32.5	92.0	69.0	85.3	70.6	71.2	60.8	64.7	64.0	-
Multi-view	MPF [33]	93.4	30.2	38.3	26.1	28.5	48.1	46.1	18.1	90.6	62.3	74.5	30.6	88.5	59.7	83.5	59.7	69.2	49.7	58.1	55.5	21
	GFNet [35]	96.0	53.2	48.3	31.7	47.3	62.8	57.3	44.7	93.6	72.5	80.8	31.2	94.0	73.9	85.2	71.1	69.3	61.8	68.0	65.4	10
	AMVNet [32]	96.2	59.9	54.2	48.8	45.7	71.0	65.7	11.0	90.1	71.0	75.8	32.4	92.4	69.1	85.6	71.7	69.6	62.7	67.2	65.3	-
M2CNet (Ours)		96.5	55.3	52.9	32.1	48.4	60.8	59.5	43.2	93.9	72.8	76.0	32.9	94.7	71.2	86.7	72.6	71.5	62.8	65.4	65.7	11

FPS is the inference time per frame (including data preprocessing time). The best score is in bold.

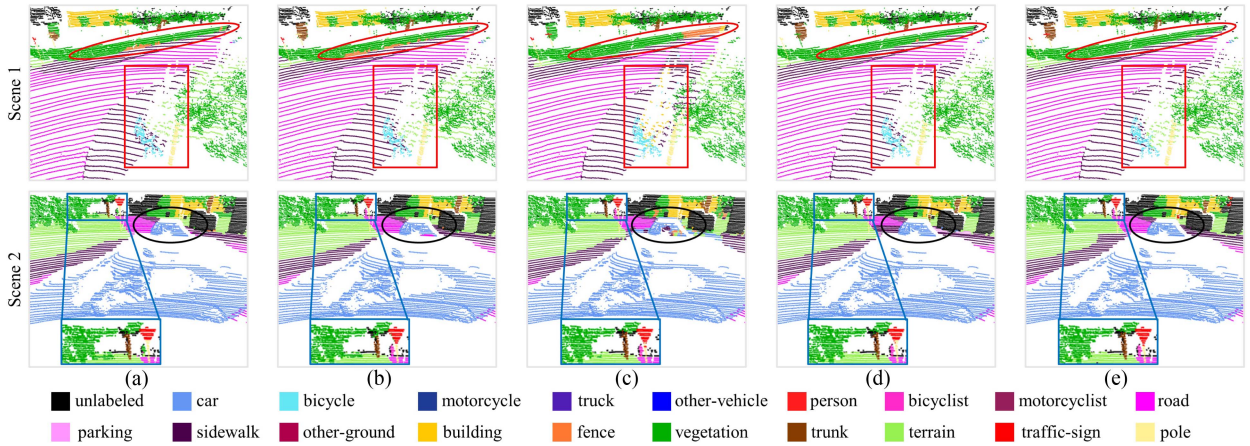


Fig. 9. Qualitative comparison results with Cylinder3D, SalsaNext, and GFNet on the SemanticKITTI dataset. To highlight the differences, the blue boxes denote regions exhibiting multi-scale characteristics, the red boxes highlight areas affected by boundary ambiguity, and the closed boxes indicate occlusions induced by view-based projection transformations. (a) Cylinder3D. (b) GBNet. (c) SalsaNext. (d) M2CNet(Ours). (e) Ground Truth.

the proposed method significantly outperforms most point-based, voxel-based, and single-view methods regarding segmentation accuracy. Compared to the multi-view projection methods AMVNet and GFNet, the overall segmentation accuracy shows improvements of +0.4% and +0.3% respectively. Due to the inevitable information loss during projection, the proposed method shows a slight 2.1% decrease in mIoU compared to Cylinder3D. However, it effectively propagates complementary information between the RV and BEV, enabling distinct feature learning from both perspectives. This leads to improvements in segmenting vertically extended objects (e.g., *trunk* and *pole*) and horizontally distributed ones (e.g., *road* and *building*). Intuitively, Fig. 9 presents qualitative comparison results with several representative methods. In scene 1, compared to the clear

boundary misclassifications of other methods, M2CNet more accurately delineates the boundary between *road* and *sidewalk*. In scene 2, the occlusion between two vehicles (closed box) causes spatial distortion in other methods, while M2CNet, with the help of MFCFM, effectively mitigates projection-induced misalignment and restores fine-grained structures. Moreover, distant multi-scale objects such as *bicyclist* and *vegetation* are better preserved, demonstrating the strong multi-scale adaptability of MESCASPP.

Due to the non-uniform density of LiDAR point clouds, with denser points in proximity and sparser points at greater distances, segmentation performance across radial ranges is crucial for model evaluation. As shown in Tables II and III, M2CNet consistently outperforms baseline methods across

TABLE II
COMPARISON OF SEGMENTATION ACCURACY (mIoU, %) FOR PERSON, BICYCLE, AND MOTORCYCLE ACROSS VARIOUS DISTANCE RANGES USING DIFFERENT METHODS

Methods	Person						Bicycle						Motorcycle					
	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}
SalsaNext [15]	73.2	62.8	43.1	23.4	13.3	+0.0	57.5	34.5	20.5	9.9	3.9	+0.0	55.8	35.4	10.2	0.2	0.0	+0.0
Cylinder3D [22]	84.7	71.0	60.8	34.3	20.9	+7.6	40.1	27.4	25.2	11.7	9.2	+5.3	53.1	61.4	54.3	10.3	0.0	+0.0
GFNet [35]	74.7	67.1	54.1	35.9	25.1	+11.8	57.4	31.1	21.2	11.7	4.8	+0.9	63.1	46.0	33.3	6.4	0.0	+0.0
M2CNet (Ours)	79.8	70.1	64.0	48.1	30.2	+16.9	59.9	35.0	31.3	22.7	13.5	+8.7	66.8	52.9	51.3	12.2	0.9	+0.9

Δ_{40-50} indicates the relative performance of these methods compared to SalsaNext at a distance of 40–50 m.

TABLE III
COMPARISON OF SEGMENTATION ACCURACY (mIoU, %) FOR CAR, POLE, AND ROAD ACROSS VARIOUS DISTANCE RANGES USING DIFFERENT METHODS

Methods	Car						Pole						Road					
	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}	0-10	10-20	20-30	30-40	40-50	Δ_{40-50}
SalsaNext [15]	94.8	89.7	76.9	62.3	51.0	+0.0	62.8	54.8	46.2	40.3	35.2	+0.0	95.3	93.3	84.8	78.9	73.2	+0.0
Cylinder3D [22]	96.2	95.0	92.9	83.6	73.8	+22.8	68.0	62.7	56.9	53.1	48.7	+13.5	93.1	94.1	90.5	85.4	77.1	+3.9
GFNet [35]	95.2	95.9	91.3	85.3	75.2	+24.2	68.2	63.2	56.0	53.0	46.7	+11.5	93.1	93.1	90.1	85.1	76.2	+3.0
M2CNet (Ours)	96.5	96.0	93.1	82.1	70.0	+19.0	68.4	63.1	57.2	53.1	49.4	+14.2	96.2	95.7	92.5	88.1	84.3	+11.1

Δ_{40-50} indicates the relative performance of these methods compared to SalsaNext at a distance of 40–50 m.

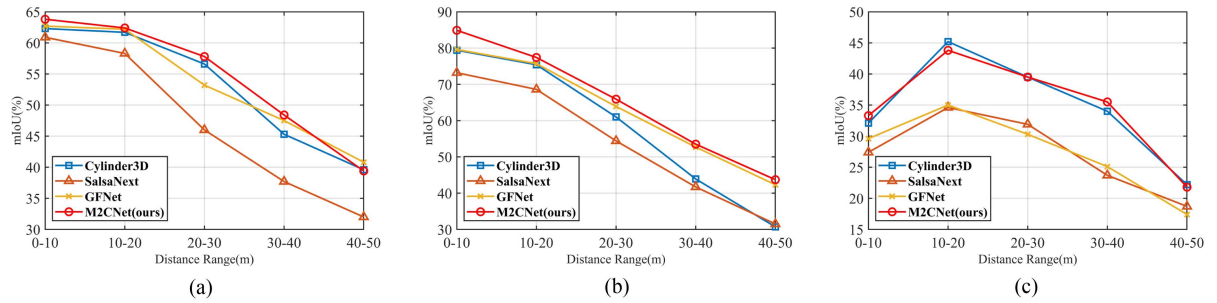


Fig. 10. Distance-based segmentation performance comparison on the SemanticKITTI, nuScenes, and RELLIS-3D validation sets. As distance increases, sparser point clouds lead to performance degradation. (a) Comparisons on SemanticKITTI. (b) Comparisons on nuScenes. (c) Comparisons on RELLIS-3D.

various distance ranges, particularly within the 40–50 m range. For small and distant targets such as *person*, *bicycle*, and *pole*, our method achieves noticeable improvements of +16.9%, +8.7%, and +14.2% over SalsaNext respectively. Additionally, for larger but distant categories like *car* and *road*, M2CNet achieves performance gains of +19.0% and +11.1% respectively. Moreover, Fig. 10(a) illustrates M2CNet consistently achieves the highest mIoU, particularly in the 0–40 m range, indicating stronger long-range perception.

2) *Results on Nuscenes*: Table IV shows that M2CNet achieves the highest score (78.9%) on the nuScenes validation set, surpassing all other compared methods across most object categories. Compared to SphereFormer, M2CNet demonstrates more consistent performance across diverse categories, particularly for small and distant objects such as *bicycle* (+1.5%) and *barrier* (+1.8%). In contrast, M2CNet maintains robust performance across both static and dynamic classes, reflecting its superior cross-view fusion and multi-scale feature modeling. Fig. 10(b) shows that M2CNet maintains the highest mIoU across all distance intervals, demonstrating better robustness in

sparse and distant regions compared to other methods. As shown in Fig. 11, M2CNet exhibits sharper segmentation boundaries between nearby objects such as *road* and *sidewalk* in scene 1, where other methods produce blurred or misclassified results. However, as shown in the red rectangle, *driveable*, *terrain*, and *sidewalk* share high similarity, and occlusion from the pedestrian causes point cloud loss. All methods show some misclassification in this area, revealing that handling similar and discontinuous regions remains challenging. In the future, we plan to introduce more effective inter-class similarity modeling and multi-frame temporal association mechanisms to address this limitation. In scene 2, due to the horizontal occlusion formed by the *manmade* and *barrier* in front of the *truck*, most existing methods suffer from large-scale misclassifications. Benefiting from more effective BEV fusion, M2CNet not only accurately identifies the *truck* but also achieves superior performance on the area of *driveable*.

3) *Results on RELLIS-3D*: To further validate the effectiveness of the proposed model in unstructured scenarios, we compare M2CNet with several representative methods. As shown

TABLE IV
QUANTITATIVE COMPARISON RESULTS OF FPS (Hz) AND mIoU (%) ON THE NUSCENES VALIDATION SET

Methods		barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other-flat	sidewalk	terrain	manmade	vegetation	mIoU (%)	FPS (Hz)
Point	SphereFormer [8]	77.7	43.8	94.5	93.1	52.4	86.9	81.2	65.4	73.4	85.3	97.0	73.4	75.4	75.0	91.0	89.2	78.4	-
Voxel	SVAseg [24]	73.1	44.5	88.4	86.6	48.2	80.5	77.7	65.5	57.5	82.1	96.5	70.5	74.7	74.6	87.3	86.9	74.7	9
	Cylinder3D [22]	76.4	40.3	91.2	93.8	51.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	76.4	75.4	90.5	87.4	76.1	9
	PCSCNet [23]	73.3	42.2	87.8	86.1	44.9	82.2	76.1	62.9	49.3	77.3	95.2	66.9	69.5	72.3	83.7	82.5	72.0	-
Single-view	PolarNet [14]	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7	71.0	26
	PC-BEV [30]	78.2	46.3	92.5	93.4	55.0	87.1	81.0	65.4	69.2	85.7	97.1	76.8	77.0	76.3	90.6	88.5	78.8	32
	RangeNet++ [12]	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8	65.5	44
	SalsaNext [15]	74.2	31.5	86.1	89.7	39.2	69.4	72.3	62.8	65.3	77.3	96.2	71.3	73.0	73.1	86.5	84.3	72.0	41
	RangeViT [27]	75.5	40.7	88.3	90.1	49.3	79.3	77.2	66.3	65.2	80.0	96.4	71.4	73.8	73.8	89.9	87.2	75.2	-
Multi-view	AMVNet [32]	79.8	32.4	82.2	86.4	62.5	81.9	75.3	72.3	83.5	65.1	97.4	67.0	78.8	74.6	90.8	87.9	76.1	-
	GFNet [35]	77.9	37.7	93.0	89.1	49.0	82.8	78.7	65.9	68.7	81.7	97.1	74.6	76.8	76.1	90.4	89.0	76.8	15
M2CNet (Ours)		79.5	45.3	93.9	88.9	66.7	83.8	78.6	65.9	67.9	85.5	96.8	77.8	77.0	77.6	89.1	87.8	78.9	18

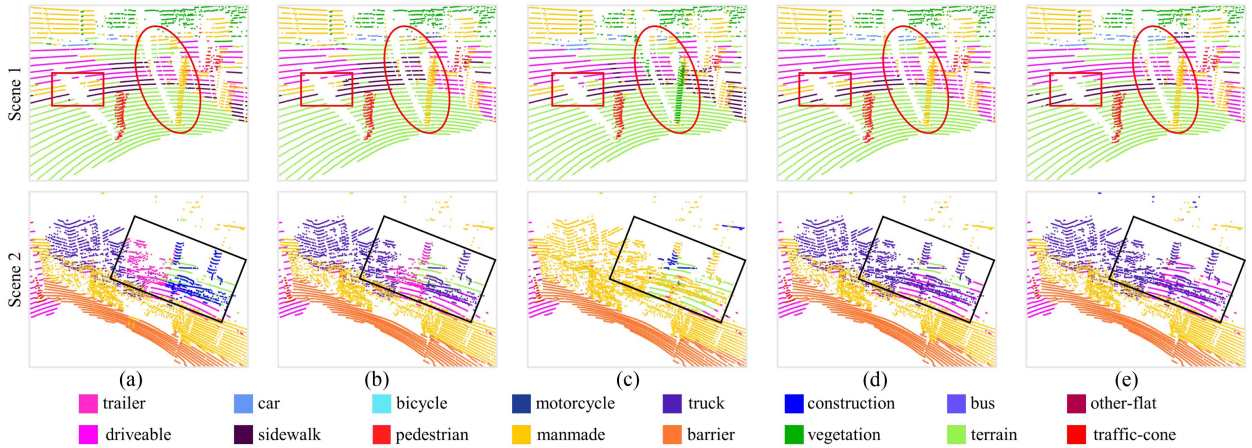


Fig. 11. Qualitative comparison results on the nuScenes dataset. To highlight the differences, the red boxes highlight areas affected by boundary ambiguity and the closed boxes indicate occlusions induced by view-based projection transformations.

in Fig. 12, M2CNet demonstrates clearer class boundaries and fewer misclassifications in complex outdoor scenes. In scene 1 and scene 2, it more accurately distinguishes similar classes such as *bush* and *grass*. In scene 2, For *vehicle* targets of varying scales, other methods exhibit varying degrees of misclassification, especially for the small and sparse *vehicle* on the left side. In contrast, M2CNet distinguishes multi-scale objects more accurately. As shown in Table V, M2CNet achieves a competitive mIoU of 44.6% on the test set, outperforming SphereFormer(+2.4%), GFNet(+1.0%), and SalsaNext(+1.5%). Compared to Cylinder3D, benefiting from multi-view feature extraction and multi-scale attention, M2CNet excels in complex off-road categories such as *concrete* (+0.8%) and *mud* (+2.6%), and shows consistent improvements in *grass*, *tree*, and *bush*, as visualized in Fig. 12. Compared to SphereFormer, M2CNet achieves better balance across small/thin objects like *pole*, *person*, and *puddle*, indicating its robustness in unstructured and fine-grained scenarios. As shown in Fig. 10(c), due to the sparsity of point clouds and cluttered backgrounds, all

methods experience a noticeable performance drop beyond 30 meters. In this range, M2CNet consistently achieves superior performance, remaining competitive with Cylinder3D overall and significantly outperforming GFNet and SalsaNext. These results underscore the robustness of M2CNet under complex long-range perception conditions.

4) *Results on Our Test Set*: To further validate the effectiveness and adaptability of M2CNet in real-world scenes, we utilize real-world data as the test set and conduct qualitative comparative analyses. As shown in Fig. 13, we select two typical driving scenes for the testing. Since the collected data primarily consists of structured point clouds, we test our model based on training conducted on the SemanticKITTI and nuScenes. Additionally, the real-world point clouds are acquired by using a 16-line LiDAR, resulting in sparser and more irregular point clouds. In scene 1, for the stationary inter-class horizontally occluded *car* category on the left side, M2CNet demonstrates superior recognition accuracy. Furthermore, for the moving *car* category on the right side, compared to GFNet and

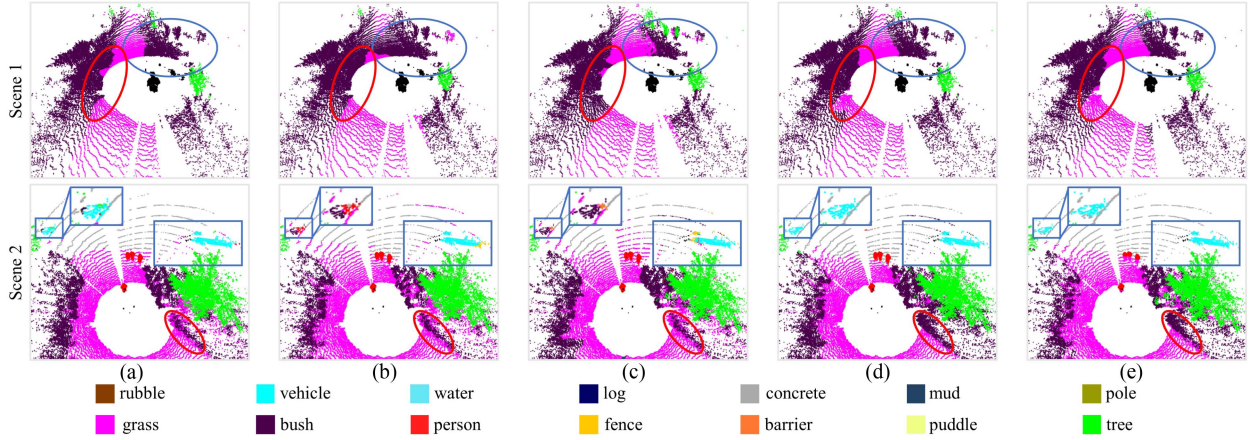


Fig. 12. Qualitative comparison results on the RELLIS-3D dataset. To highlight the differences, the blue boxes denote regions exhibiting multi-scale characteristics, the red boxes highlight areas affected by boundary ambiguity. (a) Cylinder3D. (b) GFNet. (c) SalsaNext. (d) M2CNet(Ours). (e) Ground Truth.

TABLE V
QUANTITATIVE COMPARISON RESULTS OF FPS (Hz) AND mIoU (%) ON THE RELLIS-3D TEST SET

Methods		grass	tree	pole	water	vehicle	log	person	fence	bush	concrete	barrier	puddle	mud	rubble	mIoU (%)	FPS (Hz)
Point	SphereFormer [8]	66.5	79.5	56.4	0.0	22.4	8.1	83.7	9.9	71.7	84.2	76.8	9.2	17.5	4.8	42.2	-
	RandLA-Net [16]	62.1	76.7	42.8	0.0	34.8	6.3	82.9	10.2	69.8	72.8	54.3	17.6	8.5	1.7	38.6	1.6
Voxel	SPVNAS [21]	64.1	76.1	54.4	0.0	48.6	15.1	85.8	11.0	71.5	70.0	64.8	22.4	8.5	5.2	42.7	6
	Cylinder3D [22]	65.6	75.8	61.5	0.0	62.6	4.6	86.6	9.2	72.2	82.1	82.2	22.9	8.0	2.7	45.4	7
Single-view	RangeNet++ [12]	63.0	66.5	23.5	0.0	8.7	14.4	79.0	2.5	68.4	50.8	6.9	26.8	8.6	0.6	30.0	17
	SalsaNext [15]	64.7	79.0	56.3	0.0	23.1	18.8	83.2	16.1	72.9	75.3	75.9	23.2	9.6	5.0	43.1	31
Multi-view	GFNet [35]	64.3	76.4	49.8	0.0	34.4	21.5	83.1	12.6	72.4	73.5	79.3	24.2	10.1	7.2	43.6	12
M2CNet (Ours)		65.7	76.6	65.4	0.0	46.8	2.1	86.9	7.2	72.6	82.9	81.8	25.7	10.6	0.3	44.6	14

SalsaNext, M2CNet exhibits enhanced multi-scale preservation characteristics owing to MESCASPP. In Scene 2, as indicated by the blue regions, M2CNet achieves significantly better segmentation performance for multi-scale objects such as *trunk*, *vegetation*, and *building* compared to other baseline methods. Moreover, regarding the boundary confusion issue among *car*, *road*, and *terrain* in the red regions, the experimental results further validate M2CNet's boundary preservation capability. In contrast, by deeply integrating RV and BEV, M2CNet achieves complementary feature fusion across views, significantly enhancing segmentation accuracy for objects with diverse spatial distributions, thus demonstrating improved robustness and precision.

5) *Running Efficiency Analysis*: To assess the computational efficiency of the network, we compare the Frames Per Second (FPS) of several different methods in Tables I, IV, and V. The result reveals that our method exhibits significantly higher computational efficiency compared to the point-based and voxel-based methods. Specifically, compared to Cylinder3D (7 Hz) with higher accuracy, M2CNet offers significantly higher efficiency (+4 Hz and +7 Hz respectively). Furthermore, M2CNet presents lower efficiency than single-view methods, such as SalsaNext (-17 Hz), but it achieves considerable time

efficiency while maintaining relatively high segmentation accuracy. Notably, LIDAR data is typically acquired at 10 Hz. While multi-view methods such as GFNet achieve comparable speeds (10 Hz and 12 Hz respectively), M2CNet outperforms it in both efficiency and segmentation performance. In summary, M2CNet balances accuracy and efficiency, leveraging multi-view fusion and attention mechanisms.

D. Ablation Study

To further validate the effectiveness of each module in our proposed method, we conduct detailed ablation studies on the tested datasets. All ablation experiments are carried out with identical parameter settings.

1) *Effectiveness of RV and BEV Fusion*: As shown in Table VI, we evaluate the performance of the proposed approach on the SemanticKITTI validation set. It demonstrates that cross-view feature transfer in $RV \leftarrow BEV$ and $BEV \leftarrow RV$ improves performance by +0.8% and +1.6% respectively, confirming the effectiveness of transferring complementary information across views. Furthermore, the output-level fusion in the $RV+BEV$ method improves segmentation performance by +1.7% and +3.7% compared to single-view branches, highlighting the crucial role of output-level fusion. Notably, with

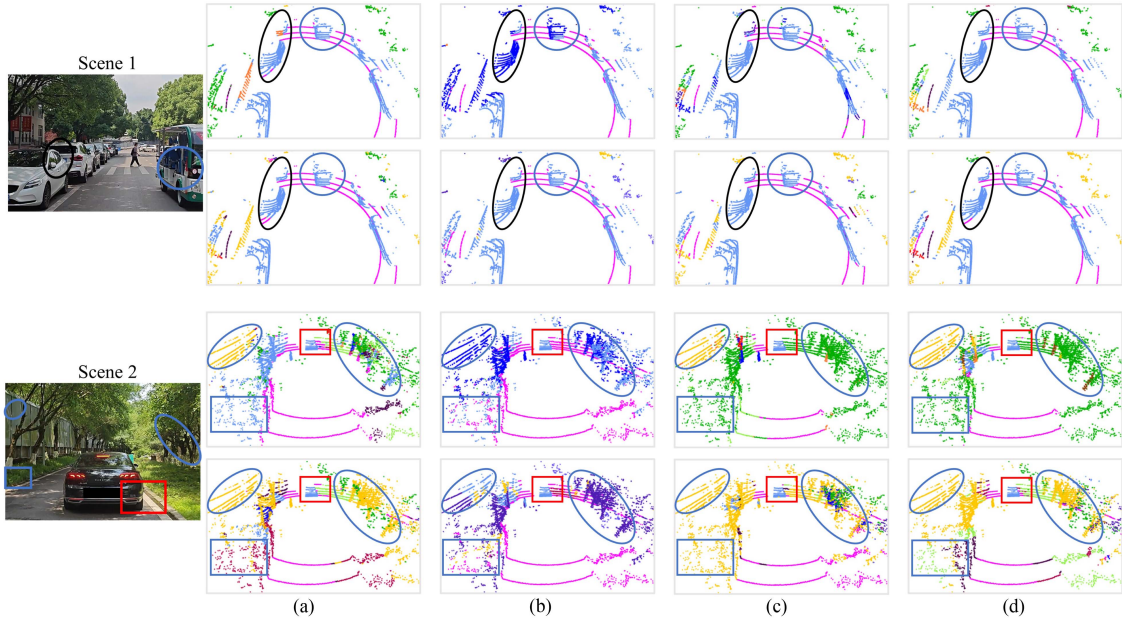


Fig. 13. Qualitative comparison results of different models on our test set. The top row of each scenario represents the predictions using the model trained on the SemanticKITTI, while the bottom row shows the predictions from the model trained on the nuScenes. The blue boxes denote regions exhibiting multi-scale characteristics, the red boxes highlight areas affected by boundary ambiguity, and the closed boxes indicate occlusions induced by view-based projection transformations. (a) Cylinder3D. (b) SalsaNext. (c) GFNet. (d) M2CNet.

TABLE VI
PERFORMANCE COMPARISON OF QUANTITATIVE ABLATION STUDIES OF THE FUSION STRATEGY ON THE SEMANTICKITTI VALIDATION SET

Methods	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mIoU (%)
RV	94.3	40.7	54.2	66.4	37.5	65.2	73.7	0.0	95.7	45.3	83.1	0.0	89.6	59.2	87.7	66.3	75.2	62.7	47.8	60.2
RV $\leftarrow$ BEV	94.0	43.1	51.7	65.2	40.1	65.3	73.7	0.0	95.3	46.6	82.9	2.2	89.7	60.5	87.9	65.6	74.9	63.2	48.7	60.6
BEV	92.7	37.5	52.8	77.9	21.1	59.4	71.6	0.0	93.1	42.9	79.2	0.1	91.4	56.0	86.9	59.6	74.7	61.3	47.2	58.2
BEV $\leftarrow$ RV	94.3	37.1	52.5	70.7	38.2	62.5	76.6	0.0	95.3	44.5	82.3	0.3	90.2	58.4	87.8	66.3	74.9	59.5	44.5	59.8
RV+BEV	94.9	44.4	57.8	70.4	39.5	68.3	79.4	0.0	95.8	45.9	83.3	0.1	90.5	60.0	88.3	68.8	75.8	64.6	48.5	61.9
M2CNet (Ours)	95.9	47.9	62.6	80.9	45.9	73.2	88.3	0.0	92.7	42.7	83.4	1.3	89.5	53.2	86.3	67.3	69.4	64.8	51.9	63.0

In the “methods” column, “RV” and “BEV” represent single-view, “RV $\leftarrow$ BEV” indicates RV fused with bev features, “BEV $\leftarrow$ RV” indicates bev fused with RV features, and “RV+BEV” represents dual-stream branches fused only at the output-level.

our proposed modules added, the final performance surpasses that of the RV+BEV method (+1.1%), showcasing the module’s potential and advantages in multi-view fusion. Fig. 14 shows comparative segmentation results across different distance intervals. Compared to single-view network models, our approach demonstrates superior overall mIoU, exhibiting high robustness and accuracy. Furthermore, the comparative results before and after view fusion indicate significant complementarity between different network layers. The M2CNet shows substantial improvement in segmentation across all distance layers, further validating the effectiveness of multi-view information fusion.

2) *Modules Effectiveness Analysis*: To evaluate the contribution of each component, we perform ablation experiments on the RELLIS-3D validation set, as summarized in Table VII. The Backbone (mode 1) alone yields the lowest inference time (50.8 ms) and parameter count (59.3 M), but with

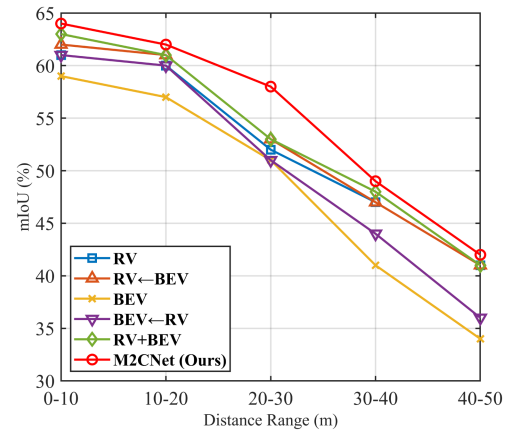


Fig. 14. Performance comparison of ablation studies of distance metrics on the SemanticKITTI validation set.

TABLE VII

PERFORMANCE COMPARISON OF ABLATION STUDIES ON THE RELIS-3D VALIDATION SET, INCLUDING NETWORK MODEL INFERENCE TIME PER FRAME AND NUMBER OF PARAMETERS

Mode	Backbone	MFCFM	MESCASPP	KPConv	Params(M)	Time(ms)	mIoU(%)
1	✓	✗	✗	✗	59.3	50.8	35.4
2	✓	✓	✗	✗	78.1	58.0	38.4
3	✓	✓	✓	✗	87.0	67.2	39.3
4	✓	✓	✗	✓	78.7	61.7	38.6
5	✓	✗	✓	✓	68.8	57.5	37.1
6	✓	✓	✓	✓	87.6	71.4	40.0

TABLE VIII

COMPARISON OF THE EFFECTIVENESS OF MESCASPP ON THE SEMANTICKITTI VALIDATION SET

Methods	mIoU(%)	Δ	Params	FLOPS
ASPP [44]	59.5	+0.0	6.56M	7.42G
ASPP(w/ ESCM)	61.8	+2.3	7.09M	7.73G
MESCASPP (w/o ESCM)	60.3	+0.8	6.30M	7.27G
MESCASPP	63.0	+3.5	6.83M	7.58G

TABLE IX

ANALYSIS OF WEIGHTED PARAMETERS α IN $Loss_F$, β IN $Loss_R$, AND λ IN $Loss_B$

α	0.5	1.0	1.5	2.0	2.5
mIoU(%)	59.8	61.2	62.1	63.0	61.8
β	0.5	1.0	1.5	2.0	2.5
mIoU(%)	60.2	63.0	62.3	61.7	60.8
λ	0.5	1.0	1.5	2.0	2.5
mIoU(%)	60.5	63.0	61.9	61.0	60.1

limited accuracy (35.4%). Adding MFCFM (mode 2) improves mIoU by +3.0% by aligning RV and BEV features. Introducing MESCASPP (mode 3) brings an additional +0.9% gain by enhancing multi-scale context. In contrast, replacing MFCFM with KPConv (mode 4) reduces 8.3 M parameters but slightly lowers performance. With all modules enabled (mode 6), M2CNet reaches the highest accuracy (40.0%) with acceptable overhead, confirming the effectiveness of our full design.

To evaluate the effectiveness of the proposed MESCASPP module, we compare it with ASPP and its variants, as shown in Table VIII. Compared with standard ASPP, MESCASPP achieves the highest mIoU of 63.0% (a 3.5% improvement) with a marginal increase in parameters and FLOPs. Removing the ESCM leads to significant performance drops, demonstrating that both components contribute positively. These results confirm the advantages of jointly enhancing spatial context modeling and multi-scale fusion in our design. To further validate the performance of the MESCASPP module, we present error maps comparing three models: the baseline model, the model incorporating MESCASPP, and the model excluding MESCASPP (see Fig. 15). These error maps visually confirm improved multi-scale segmentation performance and a notable reduction in segmentation errors.

3) *Loss Weight Distribution Analysis*: As shown in Table IX, we investigate the impact of three weighted parameters on the SemanticKITTI validation set. Each parameter is initialized to 1.0 and adjusted with a step size of 0.5. First, by varying α in

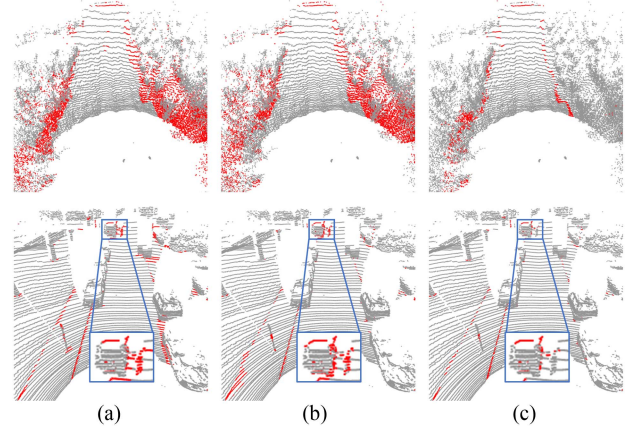


Fig. 15. Error maps comparing segmentation results of models with and without the MESCASPP module. To highlight the differences, correct and incorrect predictions are represented in gray and red respectively. (a) Baseline. (b) w/o MESCASPP. (c) w/ MESCASPP.

the dual-branch fusion loss $Loss_F$ while fixing $\beta = \lambda = 1.0$, the model achieves the best performance of 63.0% when $\alpha = 2.0$. Next, for β in the range-view loss $Loss_R$, with $\alpha = 2.0$ and $\lambda = 1.0$ fixed, the peak mIoU of 63.0% is obtained at $\beta = 1.0$. Lastly, when adjusting λ in the BEV branch loss $Loss_B$ with $\alpha = 2.0$ and $\beta = 1.0$, the best result of 63.0% is again achieved at $\lambda = 1.0$. These results suggest that α has a stronger influence on performance.

V. CONCLUSION

In this paper, we propose a novel multi-level multi-view cross-attention fusion network for point cloud semantic segmentation, named M2CNet, which significantly improves segmentation performance, particularly for autonomous vehicles in complex driving environments. The proposed method integrates semantic information from different views to compensate for the limitations of a single view. Specifically, we employ the MFCFM to integrate cross-view features by using a multi-head cross-attention mechanism, maximizing the complementary advantages of each view. To further improve the network's ability to capture multi-scale contextual information, the MESCASPP module is employed to enhance the expression of multi-scale contextual features and strengthen inter-pixel correlations. Finally, the post-processing method is employed to mitigate inverse projection errors, further improving segmentation accuracy. Experimental results demonstrate the effectiveness and scene adaptability of our method in structured and unstructured

scenarios. In future, we will focus on optimizing the network for lightweight operation to further improve real-time performance, and exploring more efficient fusion strategies to further enhance model performance.

REFERENCES

- [1] H.-F. Wang et al., "Vehicle-road environment perception under low-visibility condition based on polarization features via deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17873–17886, Oct. 2022.
- [2] Y. Zhao, C. Lei, Y. Shen, Y. Du, and Q. Chen, "Improving autonomous vehicle visual perception by fusing human gaze and machine vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12716–12725, Nov. 2023.
- [3] Y. Li et al., "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021.
- [4] B. Gao, A. Xu, Y. Pan, X. Zhao, W. Yao, and H. Zhao, "Off-road drivable area extraction using 3D LiDAR data," in *Proc. IEEE Intell. Symp.*, Paris, France, 2019, pp. 1505–1511.
- [5] L. Wang et al., "Multi-modal and multi-scale fusion 3D object detection of 4D radar and LiDAR for autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 5628–5641, May 2023.
- [6] J. Bai, L. Zheng, S. Li, B. Tan, S. Chen, and L. Huang, "Radar transformer: An object classification network based on 4D MMW imaging radar," *Sensors*, vol. 21, no. 11, Jun. 2021, Art. no. 3854.
- [7] J. Liao et al., "Road garbage segmentation and cleanliness assessment based on semantic segmentation network for cleaning vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8578–8589, Sep. 2021.
- [8] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for LiDAR-based 3D recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 17545–17555.
- [9] L. Kong et al., "Rethinking range view representation for LiDAR segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 228–240.
- [10] C. R. Qi, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 77–85.
- [11] 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.*, Long Beach, CA, USA, Dec. 2017, pp. 5105–5114.
- [12] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Macau, China, 2019, pp. 4213–4220.
- [13] 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.*, Salt Lake City, UT, USA, 2018, pp. 4490–4499.
- [14] Y. Zhang et al., "PolarNet: An improved grid representation for on-line LiDAR point clouds semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 9598–9607.
- [15] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds," in *Proc. Int. Symp. Vis. Comput.*, San Diego, CA, USA, Oct. 2020, pp. 207–222.
- [16] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 11105–11114.
- [17] H. Thomas, C. R. Qi, J.-E. Deschard, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 6410–6419.
- [18] Z. Lin et al., "SWCF-Net: Similarity-weighted convolution and local-global fusion for efficient large-scale point cloud semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Abu Dhabi, UAE, 2024, pp. 2602–2609.
- [19] J. Chen, F. Xia, J. Mao, H. Wang, and C. Zhang, "SIESEF-FusionNet: Spatial inter-correlation enhancement and spatially-embedded feature fusion network for LiDAR point cloud semantic segmentation," 2024, *arXiv:2411.06991*.
- [20] T. Feng, W. Wang, F. Ma, and Y. Yang, "LSK3DNet: Towards effective and efficient 3D perception with large sparse kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 14916–14927.
- [21] H. Tang et al., "Searching efficient 3D architectures with sparse point-voxel convolution," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 685–702.
- [22] X. Zhu et al., "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 9934–9943.
- [23] J. Park, C. Kim, S. Kim, and K. Jo, "PCSCNet: Fast 3D semantic segmentation of LiDAR point cloud for autonomous car using point convolution and sparse convolution network," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118815.
- [24] L. Zhao, S. Xu, L. Liu, D. Ming, and W. Tao, "SVASeg: Sparse voxel-based attention for 3D LiDAR point cloud semantic segmentation," *Remote Sens.*, vol. 14, no. 18, Sep. 2022, Art. no. 4471.
- [25] B. Ding, "LENet: Lightweight and efficient LiDAR semantic segmentation using multi-scale convolution attention," 2023, *arXiv:2301.04275*.
- [26] Z. Wang, Z. Liao, B. Zhou, G. Yu, and W. Luo, "SwinURNNet: Hybrid transformer-CNN architecture for real-time unstructured road segmentation," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 5035816.
- [27] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 5240–5250.
- [28] E. E. Aksoy, S. Baci, and S. Cavdar, "SalsaNet: Fast road and vehicle segmentation in LiDAR point clouds for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, USA, 2020, pp. 926–932.
- [29] W. Song, Z. Liu, Y. Guo, S. Sun, G. Zu, and M. Li, "DGPolarNet: Dynamic graph convolution network for LiDAR point cloud semantic segmentation on polar BEV," *Remote Sens.*, vol. 14, no. 15, Aug. 2022, Art. no. 3825.
- [30] S. Qiu, X. Li, X. Xue, and J. Pu, "PC-BEV: An efficient polar-cartesian BEV fusion framework for LiDAR semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, Philadelphia, PA, USA, Feb. 2025, pp. 6612–6620.
- [31] M. Gerdzhev, R. Razani, E. Taghavi, and B. Liu, "TORNADO-net: Multiview total variation semantic segmentation with diamond inception module," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 9543–9549.
- [32] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation," 2020, *arXiv:2012.04934*.
- [33] Y. A. Alnaggar, M. Afifi, K. Amer, and M. Elhelw, "Multi projection fusion for real-time semantic segmentation of 3D LiDAR point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, 2021, pp. 1799–1808.
- [34] X. Li, G. Zhang, H. Pan, and Z. Wang, "CPGNet: Cascade point-grid fusion network for real-time LiDAR semantic segmentation," in *Proc. IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, 2022, pp. 11117–11123.
- [35] H. Qiu, B. Yu, and D. Tao, "GFNet: Geometric flow network for 3D point cloud semantic segmentation," 2022, *arXiv:2207.02605*.
- [36] J. Hindel, R. Mohan, J. Bratulić, D. Cattaneo, T. Brox, and A. Valada, "Label-efficient LiDAR semantic segmentation with 2D-3D vision transformer adapters," 2025, *arXiv:2503.03299*.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.
- [38] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 13708–13717.
- [39] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 9992–10002.
- [40] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4413–4421.
- [41] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 9296–9306.
- [42] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 11618–11628.
- [43] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 1110–1116.

- [44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.



Mu Zhou (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in information and communication engineering from the Harbin Institute of Technology, Harbin, China, in 2006, 2008, and 2012, respectively. He is currently the Dean of the School of Electronic Science and Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China, and Vice Director of Mobile Communications Engineering Research Center, Ministry of Education. His research interests include quantum radar, wireless localization and sensing, and information fusion. He is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the Early Career Advisory Board Member of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, Area Editor of the *Physical Communication* (Elsevier), and Editor of the *EURASIP Journal on Wireless Communications and Networking* (Springer).



Rui Xiang received the B.S. degree in electronic information science and technology from the Chengdu University of Information Technology, Chengdu, China, in 2023. He is currently working toward the M.S. degree with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include LiDAR point cloud segmentation and deep learning.



Nan Du received the M.S. degree in 2017 from Chongqing University, Chongqing, China, where she is currently working toward the Ph.D. degree with the School of Communications and Information Engineering. Her research interests include LiDAR point cloud processing, LiDAR SLAM, and mobile multi-sensor mapping systems.



Wei He received the B.S. and Ph.D. degrees in information and communication engineering from the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2017 and 2024, respectively. She was a Joint-cultivated Ph.D. Student with Kanazawa University, Kanazawa, Japan. She joined CQUPT. Her research interests include quantum localization and computing, radar based target perception, and location.



Zhao Jiang is currently working toward the bachelor's degree with the College of Civil Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests include the Internet of Things and communication resource optimization.