RadarMOSEVE: A Spatial-Temporal Transformer Network for Radar-Only Moving Object Segmentation and Ego-Velocity Estimation

Changsong Pang[1,2*], **Xieyuanli Chen**[4*], **Yimin Liu**[3], **Huimin Lu**[4], **Yuwei Cheng**[2,3†]

[1]Northwestern Polytechnical University
[2]ORCA-UBOAT
[3]Tsinghua University
[4]National University of Defense Technology, College of Intelligence Science and Technology
dreamdusty1113@gmail.com, chengyw18@tsinghua.org.cn
yiminliu@tsinghua.edu.cn, {xieyuanli.chen,lhmnew}@nudt.edu.cn

Abstract

Moving object segmentation (MOS) and Ego velocity estimation (EVE) are vital capabilities for mobile systems to achieve full autonomy. Several approaches have attempted to achieve MOSEVE using a LiDAR sensor. However, LiDAR sensors are typically expensive and susceptible to adverse weather conditions. Instead, millimeter-wave radar (MWR) has gained popularity in robotics and autonomous driving for real applications due to its cost-effectiveness and resilience to bad weather. Nonetheless, publicly available MOSEVE datasets and approaches using radar data are limited. Some existing methods adopt point convolutional networks from LiDAR-based approaches, ignoring the specific artifacts and the valuable radial velocity information of radar measurements, leading to suboptimal performance. In this paper, we propose a novel transformer network that effectively addresses the sparsity and noise issues and leverages the radial velocity measurements of radar points using our devised radar self- and cross-attention mechanisms. Based on that, our method achieves accurate EVE of the robot and performs MOS using only radar data simultaneously. To thoroughly evaluate the MOSEVE performance of our method, we annotated the radar points in the public View-of-Delft (VoD) dataset and additionally constructed a new radar dataset in various environments. The experimental results demonstrate the superiority of our approach over existing state-of-the-art methods. The code is available at https://github.com/ORCA-Uboat/RadarMOSEVE.

Introduction

Simultaneously moving object segmentation (MOS) and ego velocity estimation (EVE) is a challenging task in computer vision, robotics, and autonomous driving. The goal of MOS is to accurately distinguish between moving and static objects in a scene, which is an important component in many downstream tasks, including collision detection, path planning, and navigation. Recently, many studies have been conducted on MOS using LiDAR (Chen et al. 2021; Kim, Woo, and Im 2022; Sun et al. 2022; Mersch et al. 2022) and image (Cheng et al. 2017; Voigtlaender et al. 2019; Wang et al. 2019; Patil et al. 2021) data, which have demonstrated promising performance. However, both LiDAR and camera sensors are susceptible to bad weather conditions, which can cause a substantial reduction in MOS performance when utilized in outdoor real-world applications.

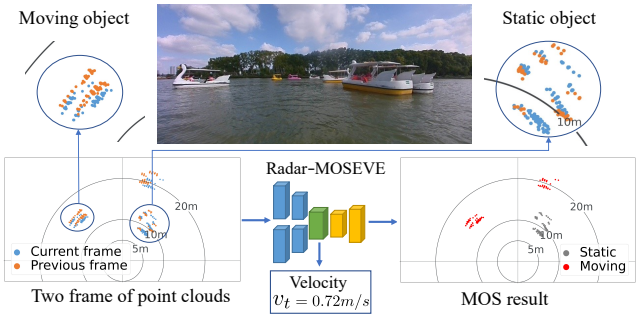


Figure 1: Our MOSEVE network takes two frames of radar point clouds as input and outputs the current ego velocity of the robot. The MOS module takes the velocity-calibrated point clouds to provide the moving segmentation.

With the advancement of integrated circuits, 77 Ghz millimeter-wave radar (MWR) (Sun, Petropulu, and Poor 2020; Li et al. 2022) has been applied in mobile robots and autonomous driving (Cheng et al. 2022). MWR exhibits strong robustness to different environments and harsh weather conditions (Lu et al. 2020a,b). Additionally, it provides radial velocity information of the measurement point, making it a valuable alternative sensor for MOS tasks. However, few studies have been conducted on MOS using radar data, and directly applying LiDAR MOS methods yields suboptimal results due to the sparsity and noise in the radar points. Furthermore, existing MOS approaches assume accurate ego-motion estimation from odometry or SLAM systems, which is not always reliable when both the robot and surrounding objects are in motion, making radar MOS tasks highly challenging.

We propose a novel radar point cloud transformer network to simultaneously achieve robust radar MOS and EVE, named RadarMOSEVE. We first modify the self- and cross-

*These authors contributed equally.
†Corresponding author

attention mechanisms and introduce a novel radar transformer that exploits the velocity information, which is suited for the sparsity characteristics of radar points. The proposed radar transformer is then used to estimate the ego velocity of the robot by utilizing consecutive radar point clouds as input, eliminating the need for odometry estimation from other sources. With the estimated ego velocity, we compensate for the radial velocity of consecutive radar point clouds and input them into a novel MOS module to accurately segment the moving objects in the current observation.

As no public dataset exists for evaluating both MOS and EVE using radar data, we create a novel dataset by utilizing two different platforms in water scenes and ground driving environments. We also annotate the radar point cloud on the public View-of-Delft (VoD) (Palffy et al. 2022) dataset. We evaluate our method and compare it to existing approaches on both datasets and demonstrate its superiority for radar MOSEVE through comprehensive experiments.

To sum up, our main contributions are threefold:

- We propose a novel radar transformer with devised self- and cross-attention mechanisms, which fits the MWR data well and extracts distinctive features from sparse and noisy radar points.
- We propose a novel radar MOSEVE framework that fully utilizes the radar Doppler velocity so that the network tackles the MOS and EVE simultaneously.
- Our RadarMOSEVE network achieves state-of-the-art performance for two tasks on the VoD dataset and our dataset.

Related Work

Moving Object Segmentation

Many works have been proposed for MOS using image (Voigtlaender et al. 2019; Wang et al. 2019; Goel, Weng, and Poupart 2018; Patil et al. 2021) and LiDAR (Chen et al. 2021; Kim, Woo, and Im 2022; Sun et al. 2022; Gu et al. 2022) data. In the vision domain, some methods detect moving objects based on optical flow (Luiten, Voigtlaender, and Leibe 2019; Dosovitskiy et al. 2015; Gong, Holsinger, and Yeung 2021). For example, Cheng (Cheng et al. 2017) fuse target segmentation features with optical flow features to improve MOS accuracy, while Yang (Yang et al. 2019) propose an adversarial network to check inconsistencies of optical flow to detect moving objects. Other deep network-based methods (Voigtlaender et al. 2019; Wang et al. 2019; Goel, Weng, and Poupart 2018; Patil et al. 2021) have also been proposed, such as Goel et al. (2018) introduce deep reinforced learning for MOS on image data, and Patilet et al. (2020) propose an end-to-end multi-frame multi-scale encoding-decoding adversarial learning network for segmenting moving objects.

The methods for MOS using LiDAR data can be classified into two main categories: projected range images-based (Chen et al. 2021; Kim, Woo, and Im 2022; Sun et al. 2022; Gu et al. 2022; Chen et al. 2022) and point cloud-based (Mersch et al. 2022; He et al. 2022; Liu et al. 2015; Mersch et al. 2023; Wang et al. 2023) methods. The former involves subtracting the range images of consecutive frames to obtain residuals, which are then used to extract spatio-temporal information. Chen et al. (2021; 2022) and Sun et al. (2022) fuse the residual image with range context to enhance the accuracy of LiDAR MOS. The latter category of methods uses sparse convolution to construct a network for segmenting moving point clouds. Some recent works such as Mersch et al. (2022) and He et al. (2022) utilize sparse convolution to extract dynamic temporal and spatial features from original point clouds using AR-SI theory (He et al. 2019). These features are then employed for MOS using sparse convolution.

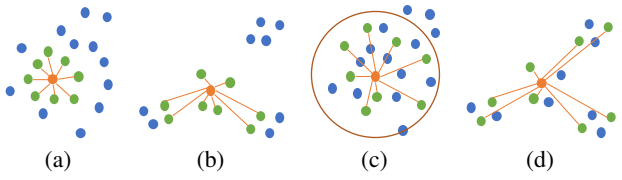
Both LiDAR and camera sensors are susceptible to adverse weather conditions, while MWR is a cost-effective and robust alternative. However, there is little research focused on MOS using radar data. One recent study by Zeller et al. (2022) utilizes a Gaussian transformer to achieve semantic movable object segmentation in radar data. Ding et al. (2022) utilize a self-supervised scene flow for radar MOS and in their later research (Ding et al. 2023), the performance of MOS is improved by cross modal supervision.

Ego-velocity Estimation

Several ego velocity estimation methods based on 4D MWR have been proposed (Monaco and Brennan 2020; Kellner et al. 2013; Park et al. 2021; Steiner, Hammouda, and Waldschmidt 2018). Most of them use RANSAC (Kellner et al. 2013) to estimate the robot's velocity. ICP (Besl and McKay 1992) can estimate a transformation between two consecutive point clouds for calculating velocity but cannot work well for noisy radar data. Cen et al. (2018) detect landmarks in MWR point clouds to estimate the relative velocity and later (Cen and Newman 2019) perform 3-DOF ego-motion calculation through the separation of key points and graph matching algorithm. Haggag et al. (2022) utilized a probabilistic model without point-to-point correspondence for ego-velocity estimation. However, it relies on static objects for ego-velocity estimation and may not perform well in the presence of moving targets. All the above-mentioned methods only focus on ego-velocity estimation and cannot segment the moving objects in radar data. To the best of our knowledge, our proposed Radar-MOSEVE method is the first work achieving both MOS and EVE using radar data.

Radar Transformer

Before presenting details of our MOSEVE network, we first introduce a novel radar transformer module. With the advent of the transformer network (Vaswani et al. 2017; Devlin et al. 2018; Liu et al. 2021), the performance of point cloud segmentation has been significantly improved (Guo et al. 2021; Engel, Belagiannis, and Dietmayer 2021; Zhang et al. 2022; Zhao et al. 2021). However, most existing methods focus on LiDAR point cloud processing, and when applied directly to radar point clouds, the performance significantly degrades due to the sparsity of radar data. To overcome this, we propose a novel radar transformer consisting of radar self-attention and cross-attention mechanisms to extract distinctive features on sparse radar data.



(a) (b) (c) (d)

Figure 2: The orange point is the source point, the green points are the points sampled by the source point and the blue points are the other points. (a) is the sampling result if k is small, (b) is the sampling result if k is large, (c) is the sampling strategy of Object Attention and (d) is the sampling strategy for Scenario Attention.

Radar Self-Attention

The original self-attention mechanism in the Point-Transformer (PT) (Zhao et al. 2021) calculates the affinity between a point and its neighbor points. Given one point cloud $\mathcal{P}_t = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^N$, $\boldsymbol{p}_i = [x_i, y_i, z_i]^\intercal$ with N points at time t. The original PT first uses multilayer perceptrons (MLPs) to generate a D dimensional feature vector $\boldsymbol{x}_i \in \mathbb{R}^D$ for each point $\boldsymbol{p}_i$ in $\mathcal{P}_t$. K-nearest neighbors (kNN) (Keller, Gray, and Givens 1985) is then used to find the neighbor point set $\mathcal{Q}_{\boldsymbol{p}_i} = \{\boldsymbol{p}_j \in \mathbb{R}^3\}_{j=1}^K \subseteq \mathcal{P}_t$ of point $\boldsymbol{p}_i$, where K denotes the number of neighbor points. The feature set $\mathcal{X}_{\boldsymbol{p}_i}$ contains feature vectors of every neighbor point in neighbor point set $\mathcal{Q}_{\boldsymbol{p}_i}$, and the affinity can be then formulated as:

$$\boldsymbol{y}_i = \sum_{\boldsymbol{x}_j \in \mathcal{X}_i} \theta(\delta(\alpha(\boldsymbol{x}_i) - \beta(\boldsymbol{x}_j) + \omega)) \odot (\gamma(\boldsymbol{x}_j) + \omega), \quad (1)$$

where α, β and γ are shared learnable linear transformations, δ is an MLP, θ is a softmax function and ω is the position encoding between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. The position encoding ω is the residual coordinate between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ through an MLP with two linear layers and one ReLU nonlinear layer. More details are referred to the original paper (Zhao et al. 2021).

The original point transformer is unsuitable for radar points, since it does not exploit the useful velocity information and the employed kNN can hardly work with sparse and noisy radar point clouds. As shown in Fig. 2(a) and Fig. 2(b), due to the sparsity of radar point clouds, it can be challenging for a single point to find sufficient neighbors belonging to the same object using the original PT. This can lead to misclassifying points from other objects and generating unsuitable features for MOSEVE. To address this issue, we propose a novel radar self-attention mechanism consisting of object attention and scenario attention mechanisms. They are designed to better capture the relevant spatial information in radar point clouds and to extract more informative features for MOSEVE.

Object Attention. We first augment the input with the radial velocity v_i provided by the radar sensor, resulting in a 4D radar point denoted as $\boldsymbol{p}_i = [x_i, y_i, z_i, v_i]^\intercal$. We then introduce an object-size attention operation by exploiting a ball query for the self-attention mechanism to better identify neighbors than simply using K-nearest neighbors in a radar point cloud. As shown in Fig. 2(c), a spherical region $\mathcal{Q}'_{\boldsymbol{p}_i}$ with radius r is centered at $\boldsymbol{p}_i$, and K points are randomly sampled within this region to ensure that each point has the same number of neighbors during network training. If the number of points is less than K, some points will be repeatedly sampled. By this, $\boldsymbol{p}_i$ can attend to all points in the region $\mathcal{Q}'_{\boldsymbol{p}_i}$ after several sampling, providing a relatively stable receptive field that facilitates the understanding of the object that the $\boldsymbol{p}_i$ belongs to.

Scenario Attention. The spatial distribution of static objects remains stable in consecutive frames, while that of moving objects can vary largely. Consequently, the relationships between points in the scenario can help reason about motion. However, the original self-attention in PT and object attention have limited receptive fields, thus unable to capture inter-object information. Therefore, we propose scenario-level self-attention for a radar point cloud $\mathcal{P}_t$. We perform interval sampling on $\mathcal{P}_t$ based on the distance to $\boldsymbol{p}_i$ to create the point set $\mathcal{Q}''_{\boldsymbol{p}_i}$, which expands the receptive field of $\boldsymbol{p}_i$ as depicted in Fig. 2(d). This process allows the embedding of $\boldsymbol{p}_i$ to incorporate features from many other objects. After multiple downsampling $\mathcal{P}_t$ using the farthest point sampling (FPS) (Moenning and Dodgson 2003), the scenario attention generates a feature that encapsulates the relationships between the object of $\boldsymbol{p}_i$ and other objects in the scene.

By leveraging both object and scenario attention, our proposed radar self-attention mechanism effectively extracts valuable features from sparse and noisy radar points. The former enhances feature sharing within individual object points, improving object motion feature extraction from sparse radar points. The latter captures scene-level features for each point by analyzing the scene's point characteristics.

Radar Cross-Attention

To fully exploit the spatio-temporal information of 4D radar point clouds, it is natural to utilize sequential data from consecutive frames for ego velocity and MOS estimation. Unlike existing work using sparse convolutions (Mersch et al. 2022), in this work, we propose a novel radar cross-attention mechanism to effectively capture the spatio-temporal dependencies among two radar point clouds. We take the previous frame $\mathcal{P}_{t-a} = \{\boldsymbol{p}_k \in \mathbb{R}^4\}_{k=1}^M$ together with the current frame $\mathcal{P}_t$ as the input of the proposed radar cross-attention, where a denotes the time interval between $\mathcal{P}_{t-a}$ and $\mathcal{P}_t$. We use the proposed ball sampling to sample K points from $\mathcal{P}_{t-a}$ to form the neighbor point set $\mathcal{S}_{\boldsymbol{p}_i} = \{\boldsymbol{p}_j\}_{j=1}^K \subseteq \mathcal{P}_{t-a}$ of $\boldsymbol{p}_i \in \mathcal{P}_t$. Then, $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are transformed into the embedding features $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ by our radar self-attention module. $\mathcal{Y}_{\boldsymbol{p}_i} = \{\boldsymbol{y}_j\}_{j=1}^K$ is the feature embedding set of all points in $\mathcal{S}_{\boldsymbol{p}_i}$. We conduct cross-attention on $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ as:

$$\boldsymbol{z}_i = \sum_{\boldsymbol{y}_j \in \mathcal{Y}(i)} \theta(\delta(\alpha(\boldsymbol{y}_i) - \beta(\boldsymbol{y}_j) + \epsilon)) \odot (\gamma(\boldsymbol{y}_j) + \epsilon), \quad (2)$$

where α, β, γ, δ, and θ are as same linear function as those in Eq. (1), but with different parameters. ϵ is the position encoding between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, which is the residual coordinate between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ through an MLP. The MLP structure of EVE differs from MOS. EVE has two linear layers and one

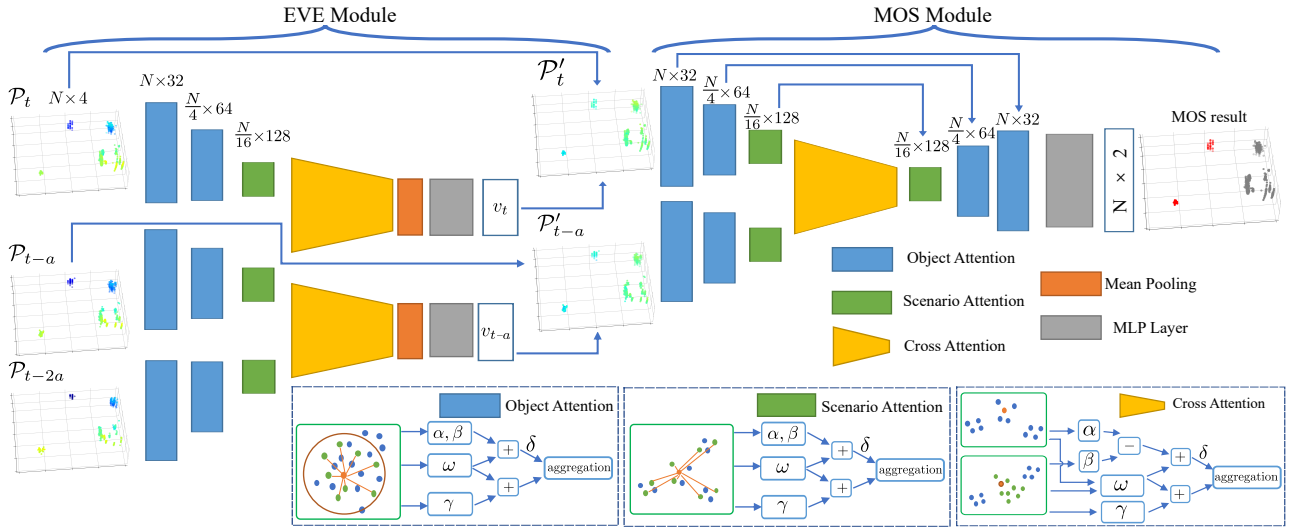


Figure 3: MOS-EVE network for ego-velocity estimation(EVE) and moving object segmentation(MOS module)

ReLU, while MOS has three linear layers and three ReLU nonlinear layers since MOS is more challenging and needs large models to learn.

Radar MOSEVE Network

Overview

We aim to achieve reliable and accurate MOSEVE simultaneously using MWR point clouds. To this end, we propose a novel multi-level attention-based network based on our proposed radar transformer modules, as depicted in Fig. 3. The network comprises two main modules, the EVE module and the MOS module, designed to accomplish both tasks. The EVE module employs two frames of 4D radar point clouds $\mathcal{P}_t$ and $\mathcal{P}_{t-a}$ to estimate the robot's ego velocity v, as detailed in **??** . To improve the accuracy of the ego-velocity estimation, we propose a novel Doppler loss during training. In the MOS module outlined in **??** , we first compensate the radial velocity of radar point clouds using the EVE results, ensuring that the radial velocity of points on surrounding static objects is close to zero. We refer to the velocity-calibrated point clouds as $\mathcal{P}'_t$ and their previous frame as $\mathcal{P}'_{t-a}$. Our MOS module takes these calibrated point clouds as input and segments the moving objects in the current frame. The training process is detailed in **??** .

Ego-velocity Estimation

EVE Backbone. It is composed of four stages. The first three stages extract intra-frame features from two radar point clouds, $\mathcal{P}_{t-a}$ and $\mathcal{P}_t$ using our radar self-attention module. It first applies the object attention module at two different resolutions to extract object features for each point in two point clouds. Then, the scenario attention module aggregates the features of different objects in each radar observation. At the final stage, the radar cross-attention module is applied to fuse point features in two point clouds and generate the inter-frame feature by incorporating spatio-temporal information from the two radar point clouds. As the network deepens, the point clouds are gradually downsampled with sampling rates of [1, 4, 4, 1]. Thus, the point set for each stage is $[\mathrm{N_p}, \mathrm{N_p}/4, \mathrm{N_p}/16, \mathrm{N_p}/16]$, where $\mathrm{N_p}$ is the number of input points. We utilize the FPS method to obtain a well-spread downsampled subset of the point cloud.

EVE Head. In the EVE head, we apply global average pooling to obtain a 128-dimensional global feature vector. This vector is then passed through an MLP consisting of three linear layers and two ReLU non-linear layers to predict the robot's velocity. The output sizes of the MLP layers are [256, 64, 1]. The final one-dimensional output represents the robot's velocity in that frame.

The velocity of static objects in the environment should be zero. However, as the robot moves, the raw velocity measurements of radar points are relative to the robot. Suppose the robot moves forward at a velocity of v. We project it in the direction of $\boldsymbol{p}_i$, obtaining the radial velocity as:

$$\hat{v}_i = -v \cdot \frac{y_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}. \tag{3}$$

When $\boldsymbol{p}_i$ belongs to a static object, its radial velocity v_i measured by radar should be equal in magnitude to the projection velocity $\hat{v}_i$ of the robot's ego velocity v. Therefore, enforcing the radial projection of the EVE network output to be equal in magnitude to the radial velocity of the static point allows the network to learn the ego velocity of the robot. We present more training details in **??** .

Moving Object Segmentation

Velocity Compensation. Assuming the robot moves forward in a short time interval a, it is necessary to convert the velocity measurement of a 4D radar point to the global coordinate system before conducting radar MOS. Because the raw velocity measurement is relative to the robot motion, it may confuse the network in determining the motion state of a point. To tackle this, we use our EVE network output $\hat{v}$ to compensate for the radial velocity v_i of the point cloud,

bringing the absolute velocity of static points close to zero. The calibrated velocity v_i' of $\boldsymbol{p}_i$ is calculated as

$$v_i' = \hat{v} \cdot \frac{y_i}{\sqrt{x_i^2 + y_i^2}} - v_i \cdot \frac{\sqrt{x_i^2 + y_i^2 + z_i^2}}{\sqrt{x_i^2 + y_i^2}}. \quad (4)$$

Since pitch, roll, and z-axis changes are typically small for ground and water vehicles in a short time, we calculate the radial velocity of the point cloud in the XOY plane.

After velocity calibration, the radial velocity of static points is approximately zero, while some moving points may also have velocities close to zero due to measurement noise and movement direction. Therefore, determining the motion state of a radar point cloud based solely on radial velocities is challenging. Thus, we use the calibrated point clouds $P_t' = \{[x_i, y_i, z_i, v_i']\}_{i=1}^{N}$ and P_{t-a}' as input data for our MOS network to reason about the motion of objects.

MOS Backbone. We use a U-net design for the MOS network using the same encoder structure as that of our EVE module. For decoding, we use trilinear interpolation to upsample the features to high-resolution point clouds. These features are concatenated with the corresponding resolution encoding features via a skip connection. We use object attention and scenario attention in different resolutions to fuse the different features, while different modules are connected during the decoding stage via the transition-up module (Zhao et al. 2021). The output of the module is a moving segmentation feature of size (N,32).

MOS Head. Given the moving segmentation feature for each point in the point cloud $\mathcal{P}_t'$, we use an MLP consisting of three linear layers and two ReLU nonlinear layers to convert these features into the final logits for MOS.

Network Training

We first train our EVE module, then use the EVE estimates to compensate for the radar point velocities, and finally train the MOS module. The EVE training loss L_{EVE} consists of an MSE loss L_{mse} and proposed new Doppler loss L_{dop} as

$$L_{EVE} = L_{dop} + L_{mse}. \quad (5)$$

Given the points $\boldsymbol{p}_i$ of all static objects form the point set $\mathcal{P}_s = \{\boldsymbol{p}_i \in \mathbb{R}^4\}_{i=1}^{N_s} \subseteq \mathcal{P}_t$ using our MOS labels, the Doppler loss is defined as

$$L_{dop} = \frac{1}{N_s} \sum_{i=1}^{N_s} \left| \hat{v} \cdot \frac{y_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}} - v_i \right|, \quad (6)$$

which makes the output $\hat{v}$ of EVE network equal to the robot ego velocity v. MSE loss $L_{mse} = \frac{1}{N_b} \sum^{N_b} (v - \hat{v})^2$ is used to provide additional supervision to the EVE estimates by comparing them to the ground truth velocity.

The calibrated 4D radar point clouds and the corresponding MOS labels are then used to train the MOS module with a typical weighted cross-entropy loss

$$L_{mos} = -\sum_{c=1}^{C} w_c \, l_c \, \log\left(\hat{l}_c\right), \quad (7)$$

where C contains moving and static two classes, w_c is the weight factor for class c, $\hat{l}_c$ and l_c are the predicted and ground truth labels.

Experimental Results

In this section, We present our experiments to illustrate that our approach is able to achieve both MOS and EVE using only radar data and outperforms existing state-of-the-art methods on our dataset and the VoD dataset. In addition, we provide multiple ablation studies to experimentally validate the effectiveness of all our network designs and architecture.

Experimental Setup

Dataset. We evaluate our methods on our dataset, including 13,654 frames of point clouds. We perform moving object segmentation using LiDAR data and then label the corresponding radar points as moving. To ensure accuracy, we manually verify and correct the labels. In addition, we also validate the advancement of our method on the open-sourced VoD dataset. The MOS label created by (Ding et al. 2023) exists lots of errors, so we annotate the dataset again. We choose the suitable data and re-split them to trainval and test datasets. The final dataset contains about 3,000 frames. More details of datasets are provided in the supplementary.

Metrics. For MOS, intersection-of-union(IoU) is calculated as a MOS metric to compare with other baselines. Since MOS is a binary classification task for each point, We also use F1-Score and accuracy to comprehensively evaluate the MOS performance. For EVE, the mean absolute error(MAE) and mean square error(MSE) of velocity metric the accuracy and robustness of EVE well. On some scenarios, high precision is more effective for path planning and robotic odometry. Therefore, we compare all methods with several precision results at different ego velocity thresholds, i.e. the ratio of estimates with errors smaller than a threshold.

Implementations Details. In all experiments, we set the time interval $a = 10$, the number of neighbors $k = 16$ in our radar transformer, and the sampling interval $g = 2$ in scenario attention. We randomly sample 512 points of every frame to test the model performance. We implement our Radar-MOSEVE network in PyTorch. We use the Adam optimizer with a weight decay of 0.001 to train our network of two tasks. The batch size is set to 4. For EVE, we train for 60 epochs. For MOS, we train for 50 epochs. The initial learning rate is 0.001. The learning rate decay ratio is set at 0.5, and it occurs every 10 epochs for MOS and 20 epochs for EVE. We train our model on a GTX3060 GPU, taking 12 hours in total. For all experiments, we use the same seed and take the average results of our trained models three times.

Evaluation on MOS

In the MOS task, we compare our method against both traditional and deep learning-based approaches using the IoU, F1 and Accuracy. These metrics are calculated separately for the static and moving classes, as well as for the average scores. For traditional methods, we utilize the point-to-point ICP algorithm (Besl and McKay 1992) to compute the transformation between two radar point clouds. We then use position residuals with a threshold to identify moving and static points. Once we obtain the EVE of the robot, we can check the difference between the EVE results and radial velocities of the point raw measurement

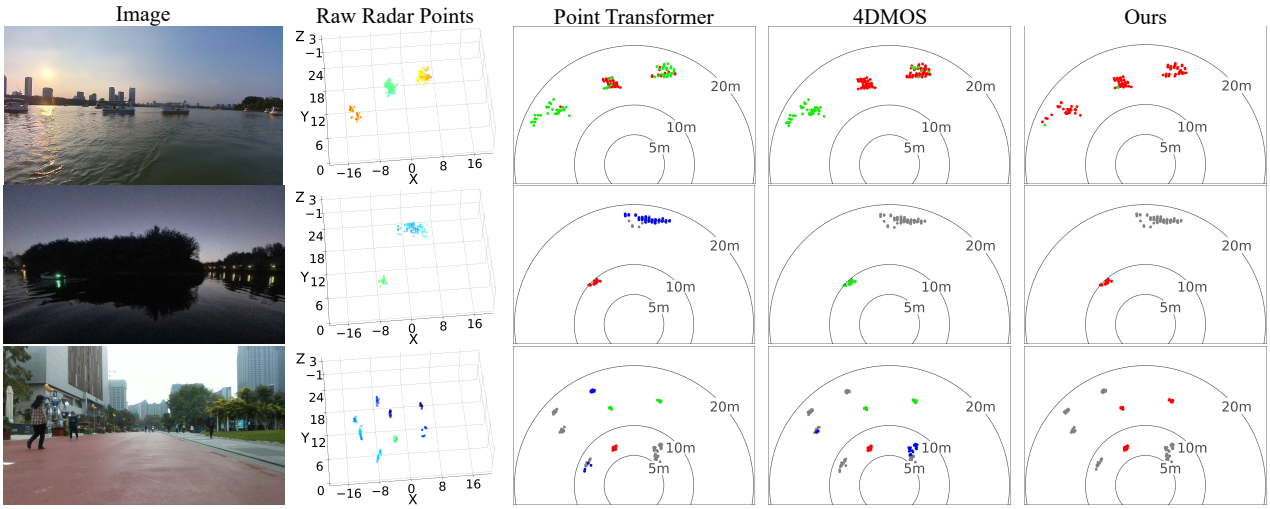


Figure 4: Qualitative results of PT, 4DMOS and ours. Red points are the true moving points, gray points are the true static points, green points are the false moving points, and blue points are the false static points. Better view with colors.

Method	IoU↑			F1↑			Acc↑		
	Static	Moving	Avg	Static	Moving	Avg	Static	Moving	Avg
ICP (1992)	23.6	26.7	25.2	33.8	38.3	36.1	48.9	39.3	44.1
RANSAC (2013)	36.2	29.1	32.6	49.0	40.7	44.9	55.4	44.8	50.1
PT (2021)	62.7	46.9	54.8	72.5	54.2	63.4	74.2	70.3	72.2
Stratified-Transformer (2022)	58.9	54.5	56.7	74.1	70.6	72.3	74.3	71.0	72.6
Point-BERT (2022)	59.9	45.1	52.5	70.1	51.5	60.8	77.5	62.8	70.1
4DMOS (2022)	67.1	54.6	60.8	76.5	62.3	69.4	78.3	73.2	75.8
PT+V	66.3	56.7	61.5	74.1	63.9	69.0	81.7	72.5	76.2
4DMOS+V	67.9	55.4	61.7	76.6	62.5	69.6	**85.0**	66.7	76.0
Ours	**73.3**	**67.2**	**70.2**	**79.8**	**73.2**	**76.5**	81.4	**82.5**	**81.9**

Bold numbers indicate the best performance while the underlined ones are the second best. +V represents using additional radial velocity.

Table 1: The MOS experimental results on our dataset.

against a threshold to determine moving and static points. Therefore, we use RANSAC (Kellner et al. 2013) as another baseline by employing it to calculate the radial velocity of the robot. As for deep learning methods, We adopt Point Transformer (PT) (Zhao et al. 2021) as our baseline, as it has demonstrated good performance in other point cloud semantic segmentation tasks. We also compare our method with other high-performing transformer-based methods, Stratified-Transformer (Lai et al. 2022) and Point-BERT (Yu et al. 2022). Additionally, we adopt the state-of-the-art LiDAR method 4DMOS (Mersch et al. 2022) as our baseline, and also evaluate the performance of PT and 4DMOS with additional radial velocity information.

As shown in Tab. 1, our method significantly outperforms all baseline methods across all evaluated metrics on average over all classes. Particularly in terms of mIoU, our method achieves 70.2% and surpasses the state-of-the-art 4DMOS by 9.4%. Additional radial velocity information can improve the performance, while our method still outperforms the improved baselines, especially in moving objects with more than 10 % improvement. Fig. 4 shows the qualitative results on the test set of different methods. Compared to other methods, our approach demonstrates superior capabilities in accurately segmenting the edges of objects and multiple objects in the entire scene into moving and static.

Method	MAE↓	MSE↓	Precision↑		
			<0.1m/s	<0.3m/s	<0.5m/s
ICP (1992)	0.842	0.870	1.1	6.9	25.2
RANSAC (2013)	0.601	0.531	10.2	25.9	49.6
PT (2021)	0.330	0.175	21.5	52.8	76.5
Ours	**0.182**	**0.065**	**43.3**	**79.7**	**94.3**

Table 2: The EVE experimental results on our dataset.

Evaluation on EVE

For the EVE tasks, we compare our method against both traditional and deep learning-based approaches using MAE and MSE of velocity estimates. We additionally provide the precision results at different ego velocity thresholds. As mentioned, we take RANSAC (Kellner et al. 2013) as a baseline in estimating the radial velocity measured by radar points. The transformation generated by ICP (Besl and McKay 1992) can be used to calculate the velocity. We also train a

Modules			MOS			EVE	
Ra-OA	Ra-SA	Ra-CA	**mIoU↑**	**F1↑**	**mAcc↑**	**MAE↓**	**MSE↓**
✗	✗	✗	65.7	72.9	77.7	0.251	0.101
✗	✗	✓	66.9	73.8	80.3	0.227	0.093
✓	✗	✓	69.0	75.5	80.2	0.197	0.071
✗	✓	✓	67.5	74.1	80.1	0.199	0.075
✓	✓	✗	66.6	73.8	79.1	0.198	0.072
✓	✓	✓	**70.2**	**76.5**	**81.9**	**0.182**	**0.065**

Table 3: Ablation study on MOS and EVE with each module

Velocity		MOS			EVE	
uncalib-EVE	calib-EVE	**mIoU↑**	**F1↑**	**mAcc↑**	**MAE↓**	**MSE↓**
✗	✗	61.1	69.3	76.6	0.301	0.161
✓	✗	65.6	73.3	78.8	**0.182**	**0.065**
✗	✓	**70.2**	**76.5**	**81.9**	-	-

Table 4: Ablation study on MOS and EVE with velocity

PT (Zhao et al. 2021) for EVE as a learning-based baseline.

The EVE results are presented in Tab. 2. As shown, our method attains the smallest MAE and MSE in EVE, which outperforms all the baseline methods. Especially for precision with a threshold of 0.5 m/s, our method achieves 94.3% surpasses PT by more than 17%. Our method consistently performs stable EVE even when the vehicle moves fast.

Evaluation on VoD dataset

To verify the generalization of our method, we evaluate our method on the VoD dataset. Apart from the methods mentioned earlier, we also compare our methods with radar-based deeping learning methods, RaFlow (Ding et al. 2022) and CMFlow (Ding et al. 2023). We only compare them on the VoD dataset, because the label of scene flow is necessary for the training stage of their networks.

Due to page limitation, we provide more experiments and qualitative results on VoD dataset in the supplementary. The experimental results indicate our method obtains competitive results both on MOS and EVE tasks in the VoD dataset.

Ablation Study

Study on Radar Transformer. The first ablation study validates the effectiveness of our proposed radar cross-attention (CA) and self-attention including object-attention (OA) and scenario-attention (SA) modules. We evaluate our method under different setups, including the one without all attention modules, while utilizing the original point self-attention module, the one only with CA, the one without CA but both self-attention modules, the one with one kind self-attention module and CA, as well as ours, using all attention modules. All setups use our velocity compensation. Ablation results for both MOS and EVE are shown in Tab. 3. As shown, the setup with all attention modules (Ours) performs the best, while the performance significantly degraded if any module was missing. The biggest decline in performance was observed when all modules were disabled, indicating the importance of our proposed modules.

Study on Velocity Compensation. To validate the benefits of using velocity measurements and compensation for MOS and EVE, we conducted an ablation study to compare the performance with and without uncompensated radial velocity and velocity compensation. The results are presented in Tab. 4. It shows that the performance gap between MOS models with and without using raw velocity measurements is large, with a 4.5% improvement in mIoU. Compensation for radial velocity results in an additional 4.6% improvement, leading to a total significant improvement of 9.1% in mIoU. Furthermore, using raw velocity data from radar points improves the EVE results, reducing errors by approximately half for both MAE and MSE. Overall, our results demonstrate that our design of using radial velocity measurements and compensation significantly improves the performance of our method for both MOS and EVE.

Additionally, We investigated the impact of the Doppler loss for EVE, the influence of the ball radius in OA, and the effects of the k value in the kNN of SA for MOS, respectively. We also explored different training strategies of multi-task multi-head networks for radar MOSEVE tasks. Due to page limitations, the details of these ablations studies are presented in the supplementary materials.

Conclusion

In this paper, we introduced a novel radar transformer network to address both MOS and EVE tasks using sparse radar point clouds. Our approach is based on our proposed radar self- and cross-attention mechanisms, which can effectively extract distinctive features from sparse radar points. Additionally, our method estimates the radial velocity of radar point clouds and utilizes it to compensate for the raw radar velocity measurements. Finally, it takes two compensated radar point clouds as input to generate the MOS results. We evaluate the performance of different methods on radar MOS and EVE tasks in our dataset and the VoD dataset. The experimental results demonstrate that our proposed method outperforms existing state-of-the-art methods in both tasks, obtaining improvements of more than 17 % in EVE precision and 9.4 % in MOS mIoU. We release the implementations of our method and dataset with the annotations to facilitate future research on Radar MOSEVE tasks.

Limitation. Although our method achieves good performance, there are still complex scenarios where it struggles to perform well. For instance, when the scene contains only moving objects, and the sparse radar measurements are solely on these objects, it creates an ill-defined MOSEVE problem, particularly when the object's motion is similar to that of the robot. One potential solution is to incorporate additional sensor modalities, such as images and LiDAR, to provide more environmental measurements and improve MOSEVE performance in such challenging situations.

Societal Impacts. Our approach is capable of accurately estimating the ego velocity and detecting potentially moving objects, such as pedestrians, in driving environments. This is particularly important for safety-critical real-world applications, such as autonomous cars and mobile robots.

Acknowledgments

This work was partly supported by the ORCA-UBOAT company and the National Science Foundation of China under Grant U1913202, U22A2059, and 62203460, as well as the Natural Science Foundation of Hunan Province under Grant 2021JC0004 and 2021JJ10024.

References

Besl, P. J.; and McKay, N. D. 1992. Method for registration of 3-D shapes. In *Sensor Fusion*, volume 1611, 586–606.

Cen, S. H.; and Newman, P. 2018. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 6045–6052.

Cen, S. H.; and Newman, P. 2019. Radar-only ego-motion estimation in difficult settings via graph matching. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 298–304.

Chen, X.; Li, S.; Mersch, B.; Wiesmann, L.; Gall, J.; Behley, J.; and Stachniss, C. 2021. Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters (RA-L)*, 6(4): 6529–6536.

Chen, X.; Mersch, B.; Nunes, L.; Marcuzzi, R.; Vizzo, I.; Behley, J.; and Stachniss, C. 2022. Automatic labeling to generate training data for online LiDAR-based moving object segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 7(3): 6107–6114.

Cheng, J.; Tsai, Y.-H.; Wang, S.; and Yang, M.-H. 2017. Segflow: Joint learning for video object segmentation and optical flow. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 686–695.

Cheng, Y.; Su, J.; Jiang, M.; and Liu, Y. 2022. A Novel Radar Point Cloud Generation Method for Robot Environment Perception. *IEEE Transactions on Robotics*, 38(6): 3754–3773.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, F.; Palffy, A.; Gavrila, D. M.; and Lu, C. X. 2023. Hidden Gems: 4D Radar Scene Flow Learning Using Cross-Modal Supervision. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 9340–9349.

Ding, F.; Pan, Z.; Deng, Y.; Deng, J.; and Lu, C. X. 2022. Self-Supervised Scene Flow Estimation With 4-D Automotive Radar. *IEEE Robotics and Automation Letters (RA-L)*, 1–8.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2758–2766.

Engel, N.; Belagiannis, V.; and Dietmayer, K. 2021. Point transformer. *IEEE Access*, 9: 134826–134840.

Goel, V.; Weng, J.; and Poupart, P. 2018. Unsupervised video object segmentation for deep reinforcement learning. *Advances in neural information processing systems*, 31.

Gong, J.; Holsinger, F. C.; and Yeung, S. 2021. FlowVOS: Weakly-Supervised Visual Warping for Detail-Preserving and Temporally Consistent Single-Shot Video Object Segmentation. *arXiv preprint arXiv:2111.10621*.

Gu, S.; Yao, S.; Yang, J.; and Kong, H. 2022. Semantics-Guided Moving Object Segmentation with 3D LiDAR. *arXiv preprint arXiv:2205.03186*.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.

Haggag, K.; Lange, S.; Pfeifer, T.; and Protzel, P. 2022. A credible and robust approach to ego-motion estimation using an automotive radar. *IEEE Robotics and Automation Letters (RA-L)*, 7(3): 6020–6027.

He, Z.; Chen, Y.; Huang, E.; Wang, Q.; Pei, Y.; and Yuan, H. 2019. A system identification based oracle for control-cps software fault localization. In *Proc. of IEEE/ACM Intl. Conf. on Software Engineering (ICSE)*, 116–127.

He, Z.; Fan, X.; Peng, Y.; Shen, Z.; Jiao, J.; and Liu, M. 2022. EmPointMovSeg: Sparse Tensor-Based Moving-Object Segmentation in 3-D LiDAR Point Clouds for Autonomous Driving-Embedded System. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(1): 41–53.

Keller, J. M.; Gray, M. R.; and Givens, J. A. 1985. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4): 580–585.

Kellner, D.; Barjenbruch, M.; Klappstein, J.; Dickmann, J.; and Dietmayer, K. 2013. Instantaneous ego-motion estimation using doppler radar. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 869–874.

Kim, J.; Woo, J.; and Im, S. 2022. RVMOS: Range-View Moving Object Segmentation Leveraged by Semantic and Motion Features. *IEEE Robotics and Automation Letters (RA-L)*, 7(3): 8044–8051.

Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; and Jia, J. 2022. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8500–8509.

Li, P.; Wang, P.; Berntorp, K.; and Liu, H. 2022. Exploiting temporal relations on radar perception for autonomous driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 17071–17080.

Liu, B.; Wang, M.; Foroosh, H.; Tappen, M.; and Pensky, M. 2015. Sparse convolutional neural networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 806–814.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 10012–10022.

Lu, C. X.; Rosa, S.; Zhao, P.; Wang, B.; Chen, C.; Stankovic, J. A.; Trigoni, N.; and Markham, A. 2020a. See through smoke: robust indoor mapping with low-cost mmwave radar. In *Proc. of Conf. on Mobile Systems, Applications, and Services*, 14–27.

Lu, C. X.; Saputra, M. R. U.; Zhao, P.; Almalioglu, Y.; De Gusmao, P. P.; Chen, C.; Sun, K.; Trigoni, N.; and Markham, A. 2020b. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *Proc. of Conf. on Embedded Networked Sensor Systems*, 109–122.

Luiten, J.; Voigtlaender, P.; and Leibe, B. 2019. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 565–580.

Mersch, B.; Chen, X.; Vizzo, I.; Nunes, L.; Behley, J.; and Stachniss, C. 2022. Receding moving object segmentation in 3d lidar data using sparse 4d convolutions. *IEEE Robotics and Automation Letters (RA-L)*, 7(3): 7503–7510.

Mersch, B.; Guadagnino, T.; Chen, X.; Vizzo, I.; Behley, J.; and Stachniss, C. 2023. Building Volumetric Beliefs for Dynamic Environments Exploiting Map-Based Moving Object Segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 8(8): 5180–5187.

Moenning, C.; and Dodgson, N. A. 2003. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory.

Monaco, C. D.; and Brennan, S. N. 2020. Radarodo: Egomotion estimation from doppler and spatial data in radar images. *IEEE Transactions on Intelligent Vehicles*, 5(3): 475–484.

Palffy, A.; Pool, E.; Baratam, S.; Kooij, J. F.; and Gavrila, D. M. 2022. Multi-class road user detection with 3+ 1D radar in the View-of-Delft dataset. *IEEE Robotics and Automation Letters (RA-L)*, 7(2): 4961–4968.

Park, Y. S.; Shin, Y.-S.; Kim, J.; and Kim, A. 2021. 3d egomotion estimation using low-cost mmwave radars via radar velocity factor for pose-graph slam. *IEEE Robotics and Automation Letters (RA-L)*, 6(4): 7691–7698.

Patil, P. W.; Biradar, K. M.; Dudhane, A.; and Murala, S. 2020. An end-to-end edge aggregation network for moving object segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 8149–8158.

Patil, P. W.; Dudhane, A.; Kulkarni, A.; Murala, S.; Gonde, A. B.; and Gupta, S. 2021. An unified recurrent video object segmentation framework for various surveillance environments. *IEEE Transactions on Image Processing*, 30: 7889–7902.

Steiner, M.; Hammouda, O.; and Waldschmidt, C. 2018. Ego-motion estimation using distributed single-channel radar sensors. In *Proc. of Intl. Conf. on Microwaves for Intelligent Mobility (ICMIM)*, 1–4.

Sun, J.; Dai, Y.; Zhang, X.; Xu, J.; Ai, R.; Gu, W.; and Chen, X. 2022. Efficient Spatial-Temporal Information Fusion for LiDAR-Based 3D Moving Object Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 11456–11463.

Sun, S.; Petropulu, A. P.; and Poor, H. V. 2020. MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. *IEEE Signal Processing Magazine*, 37(4): 98–117.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; and Chen, L.-C. 2019. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 9481–9490.

Wang, N.; Shi, C.; Guo, R.; Lu, H.; Zheng, Z.; and Chen, X. 2023. InsMOS: Instance-Aware Moving Object Segmentation in LiDAR Data. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C.; and Ling, H. 2019. Learning unsupervised video object segmentation through visual attention. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 3064–3074.

Yang, Y.; Loquercio, A.; Scaramuzza, D.; and Soatto, S. 2019. Unsupervised moving object detection via contextual information separation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 879–888.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Zeller, M.; Behley, J.; Heidingsfeld, M.; and Stachniss, C. 2022. Gaussian Radar Transformer for Semantic Segmentation in Noisy Radar Data. *IEEE Robotics and Automation Letters (RA-L)*, 8(1): 344–351.

Zhang, C.; Wan, H.; Shen, X.; and Wu, Z. 2022. Patchformer: An efficient point transformer with patch attention. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 11799–11808.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 16259–16268.