RMSNet: Real, Moving, and Semantic Segmentation of Radar Point Clouds

by

Jiaxuan Chen

to obtain the degree of Master of Science in Robotics
at the Delft University of Technology,
to be defended publicly on Wednesday August 21, 2024.

Student number: 5076412
Project duration: Dec 11, 2023 – Aug 21, 2024
Thesis committee: Dr. A. Palffy, TU Delft, supervisor
Dr. H. Caesar, TU Delft daily supervisor
Dr. N. J. Myers, TU Delft

TU Delft

RMSNet: Real, Moving, and Semantic Segmentation of Radar Point Clouds

Jiaxuan Chen Andras Palffy Holger Caesar

Abstract

Automotive radars are increasingly used in automated driving systems due to their cost effectiveness, ease of integration, and ability to withstand adverse weather conditions. Semantic segmentation for radar point clouds is a crucial step in radar pre-processing, which can be used on almost all downstream tasks of radars, such as detection and tracking. Radar point clouds are noisier compared to LiDAR point clouds due to sensor noise and the multi-path propagation, which makes the segmentation task for radar more challenging.

In this paper, we address the problem of segmentation in noisy radar point clouds in terms of ghost target vs. real detection, moving vs. static objects as well as semantic segmentation of moving road users. We demonstrate how these three tasks can be performed in a single, unified pipeline using an auto-labeled radar dataset. Our approach, called Real, Moving, and Semantic Segmentation Network (RMSNet), is able to output point-wise labels for all three tasks simultaneously. On our dataset, RMSNet attains an IoU of 82.5% for real detection segmentation, 66.9% IoU for moving object segmentation, and 64.9% mIoU for semantic segmentation. We also did a live demonstration, using the model on two different radars. The result shows that the model is capable of running inference in real-time and demonstrates good ability in model generalization.

1. Introduction

Autonomous driving relies on various modalities of sensors to increase safety redundancy. Agents need to accurately understand their surroundings to navigate safely in dynamic, real-world environments. This involves reliably identifying moving objects and their semantic information from noisy data, and predicting future states for effective path planning. Sensors such as cameras and LiDAR are able to provide rich semantic information, however, they struggle in adverse weather conditions like fog, rain, and snow. To enhance overall safety, redundant sensor setups are commonly used to avoid the risk of critical malfunctions. Radar sensors are more robust under these conditions and measure Doppler velocity and radar cross-section values, aiding in the detection of moving agents. However, fewer

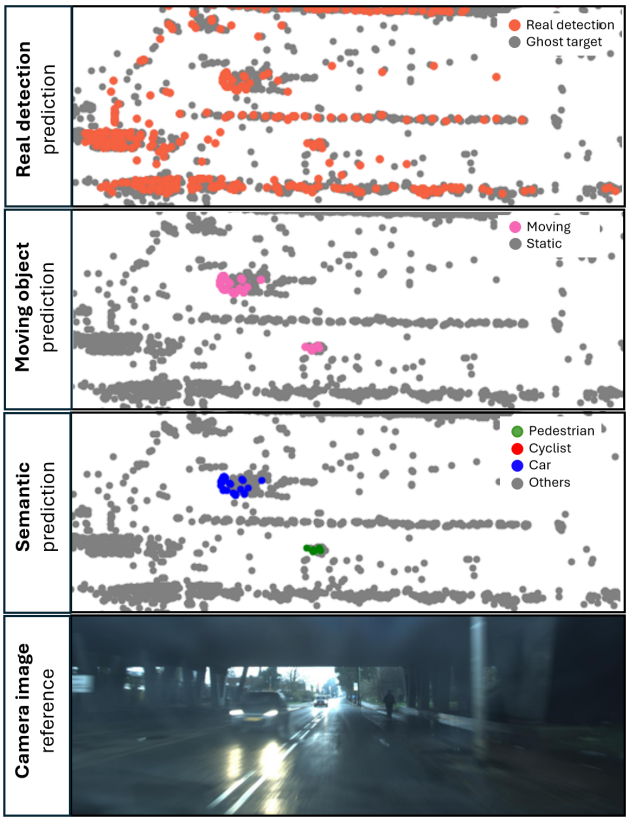

Figure 1. Our method performs real detection segmentation, moving object segmentation, and semantic segmentation at the same time. We aggregate 5 scans of radar point clouds, apart from geometric information from radar point clouds, we also feed our model with compensated velocity, radar cross section (RCS), and time info.

contributions are made in the radar point cloud segmentation field compared with LiDAR due to the fact that radar point clouds are sparse and noisy. Despite being affected by noise from multi-path propagation and ego-motion, radar can be essential for comprehensive scene interpretation. Integrating radar with cameras and LiDAR ensures safer autonomous mobility. It is crucial for applications such as mobile robots, off-road vehicles, and self-driving cars.

Ghost targets are erroneous detections in radar data that occur due to multiple indirect reflections between the target and the sensor, causing them to appear at incorrect loca-

tions. The presence of ghost targets in radar data diminishes the sensors' reliability and the robustness of their detections. Distinguishing ghost targets from real detection is a crucial step for further segmenting moving objects and extracting their semantic information. In this work, we tackle the problem of real vs. ghost, static vs. moving segmentation, along with the semantic segmentation of the radar point clouds. These three tasks require differentiating between ghost target and real detection, moving and static objects, and for those moving objects, assigning a class label to each radar detection, as illustrated in Figure 1.

Most works for segmentation in radar point clouds are single-task learning. [24, 37] address the problem of semantic segmentation for moving objects, assigning a class label to each radar point cloud. [38] tackles the problem of moving object segmentation in noisy radar point clouds, which requires differentiation between moving and stationary objects. [3] introduced the first point-based deep learning approach for ghost target detection in 3D radar point clouds by extending the PointNet [20] network architecture. However, to the best of our knowledge, no existing work addresses all three tasks simultaneously.

Therefore, we present our approach to address all these three tasks using one single pipeline. It takes sparse and noisy point cloud representations of radar scans as input and outputs three different types of labels for each point, namely, ghost target & real detection, moving & static, and class label. To extract discriminative point-wise features for radar point clouds, we extend the Point Transformer network architecture by modifying its input to include radar point features beyond spatial information and introducing three sequential heads for each task. To the best of our knowledge, there is no existing radar dataset that provides three types of point-wise labels for each of the tasks. Thus, we developed an auto-labelling pipeline for annotating noisy radar point clouds which provides point-wise labels for the three tasks described above.

Our work makes the following contributions:

- We propose a novel, multi-task radar point cloud segmentation network called the RMSNet, which is a multi-task architecture with a sequential head design that predicts ghost targets, moving objects, and semantic information.
- Since point-level annotation for radar data is rare, we introduce a novel radar auto-labeling pipeline that provides point-wise labels for ghost target & real detection, moving & static, and semantic labels.
- By recording different radar sensors' outputs and applying our auto-labeling pipeline, we demonstrate that our method reaches state-of-the-art results in all three tasks while remaining highly hardware agnostic.

2. Related Work

We conducted extensive literature research on the semantic segmentation method for point clouds and summarized some of the most popular radar datasets.

2.1. Semantic Segmentation on Point Cloud

Most of the research in semantic segmentation is from LiDAR, and can be summarized in three categories: projection-based, voxel-based, and point-based.

Projection-based methods Projection-based methods extract features by projecting 3D point clouds to front view or bird's-eye view images and subsequently fuse these features for label prediction. This approach leverages more semantic information from large-scale scenes than a single-view image, while also reducing the data size of a 3D scene compared to point clouds. Methods such as SqueezeSeg [31], SqueezeSegV2 [32], RangeNet++ [16], and SalsaNext [5] project the entire point cloud onto a sphere to exploit 2D convolutions. Additionally, several techniques [17, 39] utilize a polar bird's-eye view representation to improve performance and accommodate the varying distribution of points along the radial axis. However, projection-based approaches are susceptible to the influence of synthetic viewpoints and do not fully exploit the geometric structural information of point clouds. The process of reconstructing 3D information from 2D data inevitably leads to information loss, resulting in suboptimal segmentation accuracy.

Voxel-based methods Voxels divide the 3D space into many volumetric grids with a specific size and discrete coordinates. It contains more geometric information about the scene compared to projected images. Since point clouds on the outdoor landscape are sparse and vary in density, the voxel-based approach also takes advantage of the sparsity, as most voxels are usually unoccupied [14]. VoxSegNet [30] voxelizes point clouds into dense cuboids, utilizing atrous 3D convolution and attention-based aggregation to enhance feature extraction even under limited resolution. Recent approaches [15, 22] combine multiple representations, such as bird's-eye-view [39] with fine-grained voxel features, to reduce runtime and improve accuracy. Approaches based on sparse convolutions, where the convolution kernel is only evaluated at occupied voxels, can further reduce computation and memory requirements [36]. However, they also lose geometric detail because of quantization onto the voxel grid, which is its major drawback. It is also sensitive to the selection of the grid size; a small grid leads to a higher computation burden, while a large grid can cause more information loss, resulting in lower accuracy [10].

Point-based methods Point-based methods overcome the information loss in the encoding by directly processing the point clouds and keeping the spatial information intact The PointNet [41] is a pioneering work that directly pro-

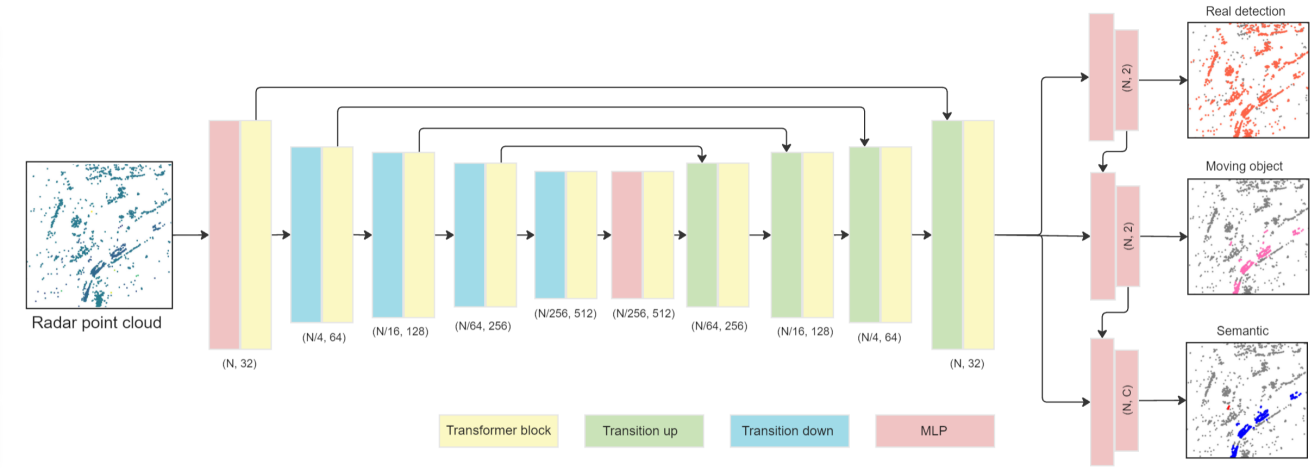

Figure 2. Overview of our method. The architecture, based on Point Transformer [41] takes accumulated radar point clouds as input. Three task specific heads, namely, the real detection head, the moving object head, and the semantic head. We make these heads sequential by concatenating the output from the previous head to the next.

cesses point clouds. It uses shared MLP to exploit pointswise features and adopts a symmetric function such as maxpooling to collect these features into a global feature representation. Building on PointNet, multiple approaches such as PointNet++ [21], use a hierarchical learning architecture that improves the network performance by exploiting local features. For radar point processing, Schumann et al. [24] adapt the approach by aggregating multiple radar point clouds for semantic segmentation and improve the approach by adding a memory module and additional features [26].

Kernel-based convolutions [27, 35] offer another approach to learning local features per point. Unlike 2D convolution, the weight function of point convolution needs to be learned from point geometric information adaptively. PointConv [34] employs a Multi-Layer Perceptron, while KPConv [27] defines an explicit convolution to directly learn the kernel. Nobis et al. [18] extended on KPConv by exploiting the time dimension of multiple radar scans for object detection. Furthermore, graph-based methods improve the connection of individual points by passing messages on constructed graphs [13]. RadarGNN [8] inherits the idea of the graph neural network, which is designed to consider both point features and point pair features, embedded in the edges of the graph. However, graph-based networks use supervoxels or over-segmented clusters to combine local areas, which are inappropriate for sparse point clouds where single detection can represent complete instances [36].

Transformer-based methods leverage the self-attention mechanism [29] and have achieved remarkable results not only in natural language processing [1, 29] but also across various tasks, including point cloud understanding [9, 12, 41]. Point Transformer, introduced by Zhao et al. [41], incorporates MLP-based position encoding into vector attention and employs a KNN-based downsampling module to reduce point resolution. The subsequent Point Stratified Transformer [12] applies dot-product attention and increases the effective receptive field by a window-based key-sampling strategy. For radar point cloud processing, Zeller et al. [37] propose an optimized transformer architecture for semantic segmentation of sparse radar point clouds using just a single scan. Furthermore, recent work elaborates positional encoding to enhance accuracy and keep position information throughout the network [4, 40].

2.2. Dataset

There are several publicly accessible datasets exist for automotive radar perception, each differing in scale and data formats. Among all 3D radar datasets, the nuScenes [2] dataset is widely recognized for its comprehensive sensor suite, which includes six cameras, one LiDAR, and five radars. The RadarScenes [25] is the first dataset that provides pointwise annotations, covering 158 diverse scenarios and includes 11 classes of moving road users. The new-generation 4D radar is becoming more and more popular because it provides enhanced resolution and improved elevation measurement capabilities. The VoD [19] dataset contains 8693 frames of synchronized and calibrated LiDAR, camera, and 4D radar data collected in diverse traffic environments, and it includes annotations for 3D bounding boxes. Similarly, the Tj4DRadSet [42] dataset contains 7,757 synchronized frames captured in complex driving scenarios, offering 3D bounding boxes and trajectory IDs. Unfortunately, these 4D datasets lack pointwise annotations and are unsuitable for semantic segmentation.

Dataset	Radar type	Annotation
nuScenes [2]	Continental ARS408	3D bounding box
RadarScenes [25]	Continental Radar	Point-wise
VoD [19]	ZF FRGen 21	3D bounding box
TJ4DRadSet [42]	Oculii Eagle	3D bounding box

Table 1. Popular radar dataset.

3. Proposed Method

Our goal is to achieve real detection segmentation, moving object segmentation, and semantic segmentation at the same time using aggregated scans from radar point clouds. We introduce a point-based framework that adapts Point Transformer [41] as the backbone, which directly processes the input radar point cloud to avoid information loss, and builds upon the effective self-attention mechanism throughout the network. We add three heads based on the architecture, so it jointly performs real detection segmentation, moving object segmentation, and semantic segmentation in a sequential manner as depicted in Figure 2.

3.1. Backbone

A point-based framework is introduced to directly process the input point clouds to avoid information loss and build upon the successful self-attention mechanism throughout the network. We adopt the encoder-decoder structure of the Point Transformer [41] which has five stages that operate on progressively downsampled point sets. The downsampling rates for the stages are [1, 4, 4, 4, 4], thus the cardinality of the point set produced by each stage is [N, N/4, N/16, N/64, N/256], where N is the number of input points. Consecutive stages are connected by transition modules: transition down for feature encoding and transition up for feature decoding.

The transformer block incorporates several components, including the self-attention mechanism, linear projections for dimensionality reduction and processing acceleration, and a residual connection. The input is a set of radar feature vectors which in our case is the 3D coordinate location of the radar point clouds, the ego-motion compensated velocity $\hat{v}_r$, radar cross section rcs, and t which indicates from which scan the point clouds are coming from. This block enables the exchange of information between these localized feature vectors, resulting in the generation of new feature vectors for all data points.

The transition down module is achieved by performing farthest point sampling [21] to identify a well-spread subset of the original point set. Feature vectors from the original point set are then pooled onto the reduced set using the k-nearest neighbors (kNN) graph, followed by a linear transformation, batch normalization, ReLU activation, and max pooling.

The transition-up module adopts a U-net [23] design. It maps features from the downsampled input point set to a higher resolution point set. Each input feature is processed by a linear layer, followed by batch normalization and ReLU activation. The features are then mapped onto the higher-resolution point set via trilinear interpolation.

3.2. Segmentation Heads

We adopt a sequential heads design to jointly perform the segmentation tasks. The process begins with the real detection segmentation head, where the final decoder stage generates a feature vector for each point in the input set. A multi-layer perceptron (MLP) maps these features into per-point logits, enabling the network to effectively distinguish between real detections and ghost targets. This initial step filters out irrelevant data and focuses subsequent processes on points of interest.

The moving object segmentation head concatenates the output from the real detection head with features from the backbone network, creating a richer feature set. This enriched data set is passed through another MLP with two linear layers to obtain per-point logit values for moving object segmentation. The sequential arrangement allows this head to capitalize on the distinctions made by the real detection head, reducing redundancy and enhancing feature specificity.

Finally, for further segmenting class information from the moving objects, the semantic segmentation head integrates outputs from the moving object head with additional backbone features. Similarly, this head outputs the final logits by passing through an MLP, classifying each point into pedestrians, cyclists, cars, and background.

Each head benefits from its predecessors, providing contextually enriched input and capturing complex interdependencies, thereby improving segmentation accuracy and robustness. This cohesive framework is particularly effective in scenarios requiring a nuanced understanding of complex environments.

3.3. Joint Training

The full network is trained end-to-end via a multi-task loss function. Denote L as a loss function. For a given input data frame, the multi-task loss function, L_{MTL}, is formulated as a weighted sum of the task-specific losses:

$$L_{\text{MTL}} = \sum_{i \in \{\text{real,moving,semantic}\}} w_i L_i,$$

where w_i and L_i are the task-specific loss weights and loss functions, respectively. To learn $y_{\text{real}}, y_{\text{moving}}, y_{\text{semantic}}$, we use the standard cross entropy loss following the approach by Schumann [24].

The weight w_i determines the impact of each task on the overall loss. For our model, we adopt the uncertainty

	range	velocity	azimuth	elevation
ZF FRGen 21				
Accuracy	≤0.02 m	0.01 m/s	0.15°	0.3°
Resolution	≤0.2 m	0.1 m/s	1.5°	1.5°
Continental ARS540				
Accuracy	≤0.02 m	0.03 m/s	0.1°	0.1°
Resolution	≤0.22 m	0.1 m/s	1.2°	2.3°

Table 2. Radar Specifications.



Figure 3. Sensor setup. Our recording platform is equipped with a monocular camera setup, three 3D LiDAR sensors, a ZF FRGen 21 radar, a Continental ARS540 radar, and a combined GPS/IMU inertial navigation system.

weighting strategy proposed by Kendall et al. [11] which leverages task-dependent uncertainty, parameterized by the noise parameter σ to balance the task-specific losses. These noise parameters are optimized jointly during training, resulting in an adaptive multi-task loss function $L_{\text{MTL}}^{\text{adaptive}}$ formulated as:

$$L_{\text{MTL}}^{\text{adaptive}} = \sum_{i \in \{\text{real,moving,semantic}\}} \left(\frac{1}{2\sigma_i^2} L_i + \frac{1}{2} \log \sigma_i^2 \right).$$

4. Dataset

The model is trained on Perciv AI's proprietary multi-modal dataset which is recorded in and around Delft, the Netherlands. We recorded the data with one ZF FRGen21 radar, one Continental ARS540 radar, together with three LiDARs, and one stereo camera for labeling purposes. We develop an auto-labeling pipeline that can provide point-wise labels on radar point clouds.

4.1. Sensors Setup

We recorded the output of the following sensors: a ZF FRGen21 radar and a Continental ARS540 radar (see Table 2 for specification) mounted in the front bumper, a stereo camera mounted on the windshield, an Ouster OS1-128, a Livox MID-70 and a Robosense M1 scanner on the roof of the car, and the ego vehicle's odometry. All sensors were jointly calibrated. See Figure 3 for a general overview of the sensor setup.

The dataset is in synchronized "frames" similar to [2], consisting of a LiDAR point cloud, a rectified mono camera image, a radar point cloud, and a transformation describing the odometry. All radar point clouds are ego-motion compensated.

4.2. Auto-labeling pipeline

The exploration of auto-labeling techniques to generate point-wise annotations for the radar dataset presents a promising avenue to mitigate the burden of manual data labeling. The annotation process for radar point clouds is complicated and tedious, while annotation precision is essential to model training. In total three different types of labels are desired for the dataset: ghost target & real detection, moving & static, and three types of road users. We first perform ego-motion compensation on original radar point clouds. For labeling ghost targets, our approach is to project radar point clouds to the LiDAR frame, and compare the distance between the radar point cloud and its closest LiDAR point clouds in 3D coordinates, if their distance is within the distance threshold, a real detection label is assigned to the radar point cloud, if it is beyond the threshold, then it is marked as a ghost target. For labeling moving objects, we also make use of the semi-automatic bounding box annotation generated from MS3D [28] using LiDAR, each bounding box is also assigned with a velocity based on tracking. In order to compensate for the vanishing horizontal component of the object velocity due to its Doppler effect limitations, for example, a pedestrian walking cross the road in front of the car. We label the moving radar points by setting a velocity threshold of $0.3m/s$ for the bounding box based on empirical evidence [24]. Radar point clouds beyond this threshold are labeled as moving. Due to the fact that these bounding boxes generated are not the ground truth, meaning some missing detections are expected. Taking that into consideration, for the rest of the radar point clouds, we again apply this velocity threshold on the ego-motion compensated velocity of the radar point clouds and perform the DBSCAN algorithm [6] to cluster the miss detected moving objects. There are in total three types of road users that we are interested in, namely, pedestrians, bikes, and cars which are also aligned with the classes given from the bounding boxes. We annotate radar point clouds by aligning the bounding boxes in the radar frame, if radar points fall inside the bounding box, it is annotated with the class of the bounding box. With this auto-labeling pipeline, we are able to generate all three types of point-wise labels on all the radar datasets mentioned in Table 1, as long as it is annotated with bounding boxes and calibrated LiDAR point clouds are provided.

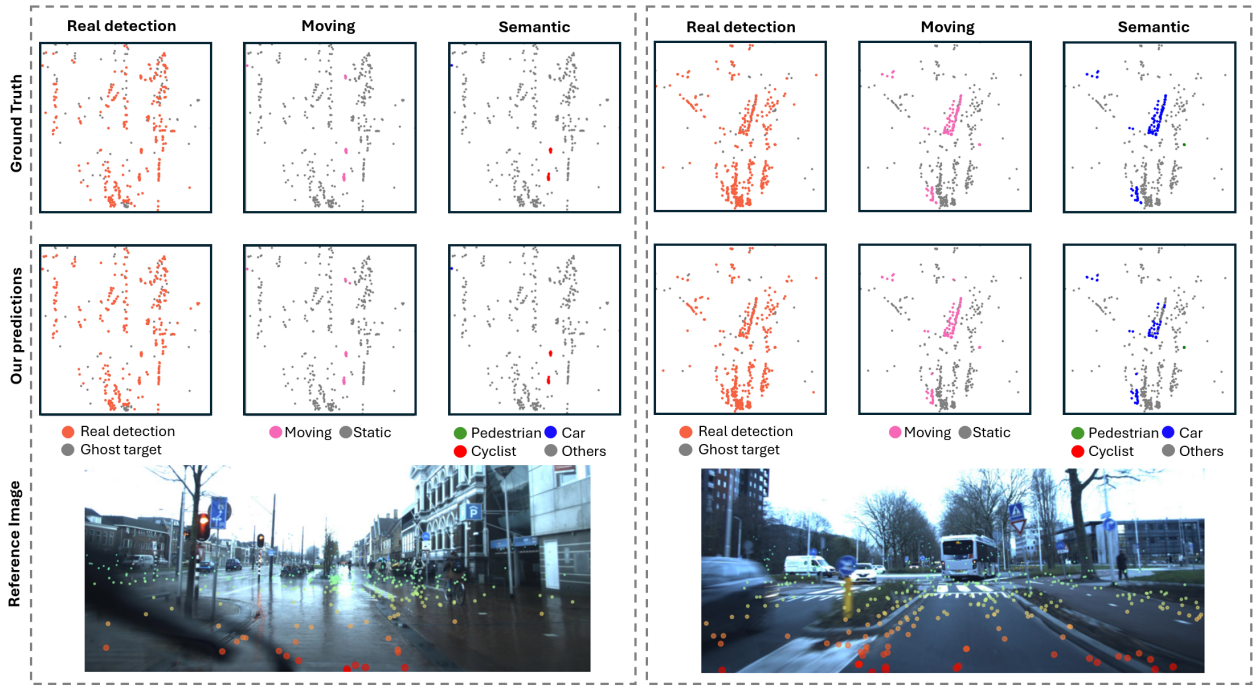

Figure 4. Visualization of our three heads segmentation results on our test set. The top row is the ground truth for real detection, moving objects, and semantic segmentation. In the middle are the predictions from our network. At the bottom are the camera images with radar point clouds projected for reference.

Method	Input	Real detection	Moving object	pedestrian	bike	car	background	Semantic mIoU
Threshold $\|v_i\| > 0.3m/s$	single-scan	-	26.0	-	-	-	-	-
Stratified Transformer [12]	single-scan	-	60.7	-	-	-	-	-
Gaussian Radar Transformer [37]	single-scan	-	-	38.6	50.8	**64.1**	**99.4**	63.0
RadarPNv1 [24]	aggregation	-	-	36.8	52.6	57.4	99.2	61.5
Ours	aggregation	82.5	**66.9**	**41.9**	**54.5**	63.8	99.3	**64.9**

Table 3. Results for real detection, moving object, and semantic segmentation tasks on our test set in terms of IoU. RMSNet is the only network that can address all three tasks simultaneously.

5. Experiments

We train and evaluate our model on our own dataset. The pseudo ground truth is generated using the auto-labeling pipeline mentioned in subsection 4.2. For each of the segmentation tasks, we compare it with the baseline method and evaluate it in terms of IoU. An ablation study is also performed on the input features. Moreover, we test our model's generalization ability by evaluating its prediction result on a different radar. At last, we evaluate our model's inference speed and test it in a live demonstration.

5.1. Experimental Setup

We train and evaluate our method on our own dataset, in total, we have 151 annotated scenes, with 508k+ annotations. We use in total 128 scenes from the dataset for training, and split the remaining 23 scenes into validation and test set. We aggregate 5 scans together to generate a denser point cloud, using the point clouds from 4 previous timestamps, and for each scan, we label them correspondingly from -4 to 0 as the time information. We use $\text{IoU} = \frac{TP}{TP+FN+FP}$ [7] and mean Intersection over Union $\text{mIoU} = \frac{1}{C}\sum_{i=1}^{C} \text{IoU}_i$ scores to evaluate the performance based on the latest scan for all three heads. Our RMSNet is trained over 50 epochs with AdamW optimizer with an initial learning rate of 0.001 with a weight decay of 0.0001. The learning rate is subjected to a decay factor of 0.7 every 20 epochs to ensure gradual convergence. We use weighted cross-entropy, following the approach by Schumann et al. [24] set the weights for moving objects to 8.0 and for static ones to 0.5. For number of neighbors k in the transition down module, which is used in determining the local neighborhood around each point, we set it to 16 following the

ablation study made in [41]. We train the network with one Nvidia RTX 4090 GPU and a batch size of 16. To reduce overfitting, we further apply data augmentation, using normalization.

5.2. Segmentation Performance

Figure 4 shows some qualitative results on the test set. We compare our RMSNet with the state-of-the-art method: the modified PointNet++ [21] by Schumann, which we refer to as RadarPNv1 [24] in our table. Gaussian Radar Transformer [37] and Stratified Transformer [12] are also selected as the baseline. We implement and reproduce their code, and test on our own dataset. We also use a threshold-based method as the baseline to compare for the moving segmentation evaluation as shown in Table 3. We select RadarPNv1 as a reference since the method meets the following requirements: (1) aggregation input or comparability; (2) RadarPNv1 is built on PointNet++, which is also a point-based method; (3) it has good performance on different benchmarks including semantic scene understanding. For semantic segmentation, our RMSNet outperforms the RadarPNv1 in all four classes.

The resulting confusion matrix is shown in Figure 6. The majority of classes with label static show the highest true positive value. The second comes the car which achieves 87.2%, followed by the bike 73.9% and the pedestrian 70.3%. It is also worth mentioning that around 16% of the bikes are misclassified as pedestrians and 15% of the pedestrians are misclassified as bikes. The main cause is the imbalance of the dataset and the limitation of our auto-labeling pipeline. For example, some motorcycles are labeled as bikes in the dataset, however, they have a much higher velocity which makes them potentially much easier to be misclassified as cars. One should also keep in mind that the differentiation between reflections from moving and non-moving objects is significantly more complex than merely applying a threshold to the Doppler velocity and classifying reflections with velocities below this threshold as static. In practical scenarios, numerous reflections from the object that has a low radial velocity, for example, a pedestrian walking perpendicular to the direction of the sensor is pointing.

5.3. Variation of the Input Features

To understand which features contribute the most to the model performance, we conduct an ablation study using four distinct sets of input features: $f_1 = (x, y, z, \hat{v}_r)$, $f_2 = (x, y, z, rcs)$, $f_3 = (x, y, z, t)$, and $f_4 = (x, y, z)$. These results are then compared to the original features $f_0 = (x, y, z, \hat{v}_r, rcs, t)$. The IoU scores for each input configuration are presented in Table 4. As shown in the table, the ego-motion compensated velocity is arguably the most important feature both for moving and semantic segmentation tasks. Moreover, by adding rcs and t, we improve the moving object IoU by 4 absolute percentage points and another 4 percentage for semantic mIoU. Regarding the IoU for real detection, since we label them using LiDAR point clouds, the geometric information plays the most important role here in distinguishing the ghost targets.

Input Feature	Real detection IoU	Moving object IoU	Semantic mIoU
$x^F = (x, y, z)$	80.1	27.4	38.7
$x^F = (x, y, z, \hat{v}_r)$	80.5	62.6	60.2
$x^F = (x, y, z, rcs)$	80.3	34.8	46.6
$x^F = (x, y, z, t)$	80.4	28.5	39.3
$x^F = (x, y, z, \hat{v}_r, rcs, t)$	**82.5**	**66.9**	**64.9**

Table 4. Ablation study on the different input features in terms of IoU on our data set.

5.4. Model Generalization

Our model also demonstrates good generalization capability when inferencing on a different radar. Figure 5 shows the inference result of two different radars in our test set. The model is only trained on data from ZF FRGen 21, but also shows good results when inferencing on Continental ARS540 radar, which our model has never seen before. To evaluate the model's inference performance quantitatively, we also generate labels for Continental ARS540 radar using our auto-labeling pipeline. For real detection and moving object segmentation, our model achieves an IoU of 65.7 and 61.7 respectively. For semantic segmentation, it obtains an mIoU of 49.5 (pedestrian: 21.08, cyclist: 30.2, car: 47.96, background: 98.82).

5.5. Inference speed

We also evaluate the inference time of our approach, demonstrating that it runs sufficiently fast to support real-time processing in vehicles. Our tests were conducted on a Nvidia RTX 4090 GPU. Given the variability in the number of detections across point clouds, we assessed 500 scans randomly selected from the validation set. The average runtime was 19.38 milliseconds per scan with a standard deviation of 4.67 milliseconds, corresponding to a processing rate of 51 Hz, which is significantly higher than the output frame rate of the tested sensors.

5.6. Live Demonstration

To further demonstrate the performance and the speed of our model, we made a ROS node and tested it on the Rosbags that we recorded. We tested it both on ZF FRGen 21 which our model is trained on and Continental ARS540 radar. Figure 7 is a screenshot of the moving object segmentation inference result on Continental ARS540 radar. This example Rosbag is recorded on a highway scene. From the

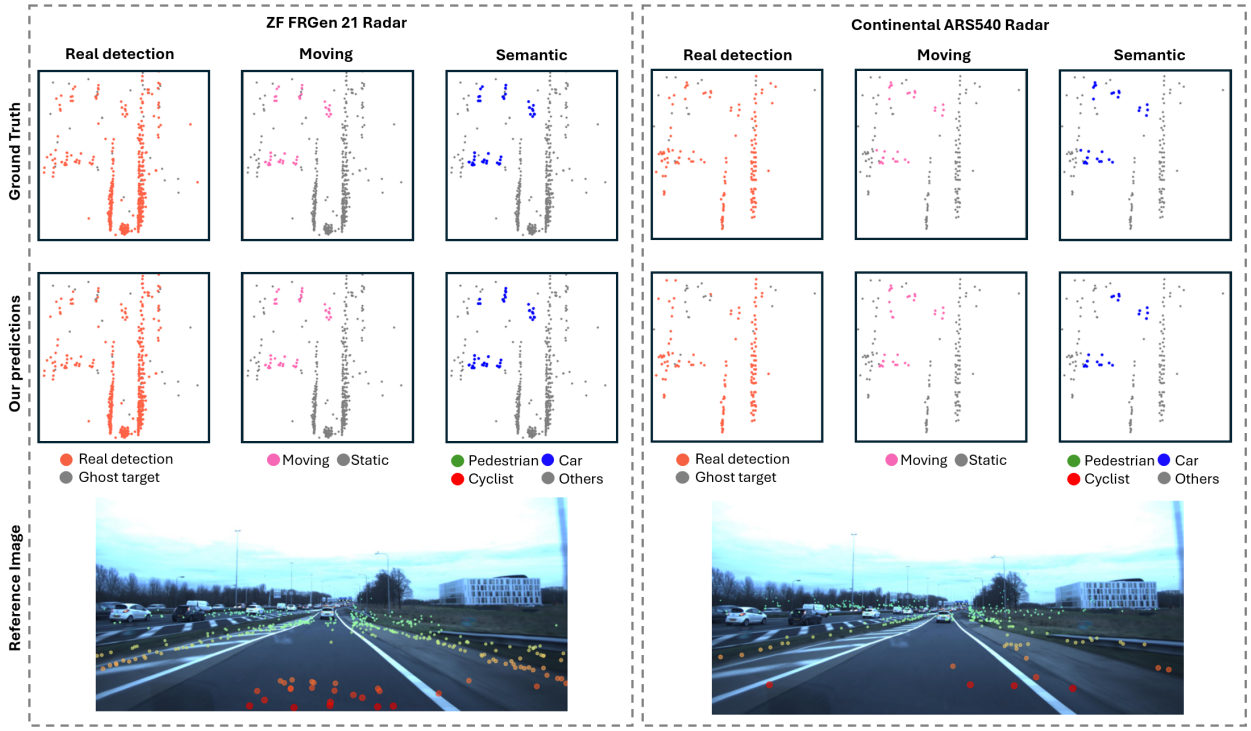

Figure 5. Visualization of the results on one frame from the test set using two different radars. To the left is the result from ZF FRGen 21 radar, where our model is trained on. To the right is the inference result on Continental ARS540 radar, which our model has never seen before.

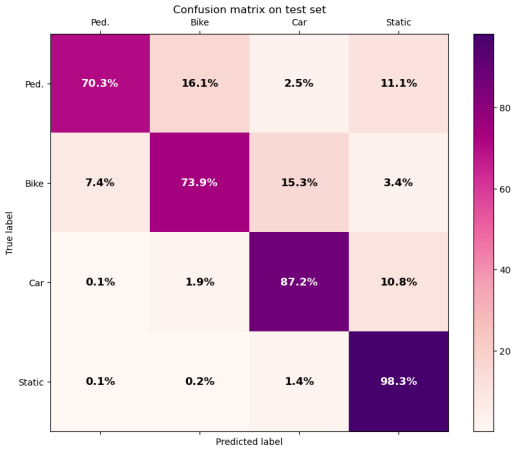

Figure 6. Confusion matrix for the four semantic classes.

figure, cars in between the guardrails on two sides are correctly identified as moving objects, however, there are still some ghost targets to the left of the guardrail that are labeled as moving, which are not desired.

6. Conclusion

In this work, we present our RMSNet which uses Point Transformer [41] as the backbone to achieve real detection,

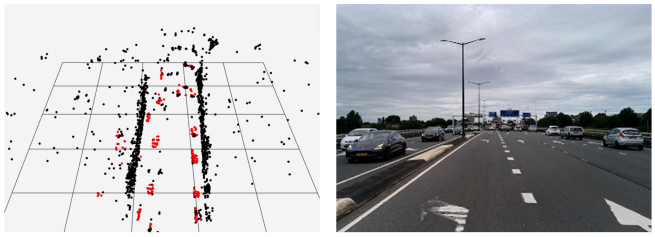

Figure 7. Inference result for moving object segmentation on Continental ARS540 radar.

moving objects, and semantic segmentation at the same time. With this sequential three heads design tailored for radar point cloud processing, to the best of our knowledge, we are the first network to achieve this multi-task segmentation on radar point clouds. We also develop an auto-labeling pipeline that can provide three different types of point-wise labels on radar point clouds, namely, real detection & ghost target, moving & static, and three types of road user labels (pedestrian, bike, car). We train and evaluate the network on our dataset. The model was also deployed and tested in a live demo and demonstrated good results even on a different radar which our model has never seen before. Overall, our work shows a comparable result with state-of-the-art paper, taking a step forward towards reliable multi-scan radar point cloud segmentation and sensor redundancy for autonomous

vehicles.

Future work: Future work could focus on a more advanced and accurate auto-labeling pipeline for the radar point clouds. In our work, we only make use of the LiDAR point cloud to generate bounding boxes for the classes of our interest, which inevitably leads to miss detections and some faulty bounding boxes. In this case, it affects the quality of the semantic labels. To tackle that, camera information can be further exploited by subtracting its rich semantic info to label road users. Moreover, to further improve the moving label quality, especially targeting objects with low radial velocity, M-detector [33] can be considered, as it segments moving LiDAR point clouds from static ones. Based on the M-detector, we can further ensemble it with bounding boxes and DBSCAN to achieve more accurate moving object labels.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3, 4, 5

[3] Mahdi Chamseddine, Jason Rambach, Didier Stricker, and Oliver Wasenmuller. Ghost target detection in 3d radar data using point cloud based deep neural network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10398–10403. IEEE, 2021. 2

[4] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556, 2021. 3

[5] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, pages 207–222. Springer, 2020. 2

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 5

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 6

[8] Felix Fent, Philipp Bauerschmidt, and Markus Lienkamp. Radargnn: Transformation invariant graph neural network for radar-based perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–191, 2023. 3

[9] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021. 3

[10] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey. *arXiv preprint arXiv:2103.05423*, 2021. 2

[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 5

[12] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 3, 6, 7

[13] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 3

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2

[15] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11809–11818, 2022. 2

[16] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019. 2

[17] Sambit Mohapatra, Mona Hodaei, Senthil Yogamani, Stefan Milz, Heinrich Gotzig, Martin Simon, Hazem Rashed, and Patrick Maeder. Limoseg: Real-time bird's eye view based lidar motion segmentation. *arXiv preprint arXiv:2111.04875*, 2021. 2

[18] Felix Nobis, Felix Fent, Johannes Betz, and Markus Lienkamp. Kernel point convolution lstm networks for radar point cloud segmentation. *Applied Sciences*, 11(6):2599, 2021. 3

[19] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian FP Kooij, and Dariu M Gavrila. Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022. 3, 4

[20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification

and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3, 4, 7

[22] Haibo Qiu, Baosheng Yu, and Dacheng Tao. Gfnet: Geometric flow network for 3d point cloud semantic segmentation. *arXiv preprint arXiv:2207.02605*, 2022. 2

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 4

[24] Ole Schumann, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Semantic segmentation on radar point clouds. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2179–2186. IEEE, 2018. 2, 3, 4, 5, 6, 7, 11

[25] Ole Schumann, Markus Hahn, Nicolas Scheiner, Fabio Weishaupt, Julius F Tilly, Jürgen Dickmann, and Christian Wöhler. Radarscenes: A real-world radar point cloud data set for automotive applications. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021. 3, 4

[26] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wöhler, and Jürgen Dickmann. Scene understanding with automotive radar. *IEEE Transactions on Intelligent Vehicles*, 5(2):188–203, 2019. 3

[27] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 3

[28] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Ms3d: Leveraging multiple detectors for unsupervised domain adaptation in 3d object detection. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 140–147. IEEE, 2023. 5

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[30] Zongji Wang and Feng Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE transactions on visualization and computer graphics*, 26(9):2919–2930, 2019. 2

[31] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE, 2018. 2

[32] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 international conference on robotics and automation (ICRA)*, pages 4376–4382. IEEE, 2019. 2

[33] Huajie Wu, Yihang Li, Wei Xu, Fanze Kong, and Fu Zhang. Moving event detection from lidar point streams. *nature communications*, 15(1):345, 2024. 9

[34] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019. 3

[35] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021. 3

[36] Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li, Xiangyu Sha, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu, et al. Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review. *IEEE Transactions on Intelligent Vehicles*, 2023. 2, 3

[37] Matthias Zeller, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Gaussian radar transformer for semantic segmentation in noisy radar data. *IEEE Robotics and Automation Letters*, 8(1):344–351, 2022. 2, 3, 6, 7

[38] Matthias Zeller, Vardeep S Sandhu, Benedikt Mersch, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Radar velocity transformer: Single-scan moving object segmentation in noisy radar point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7054–7061. IEEE, 2023. 2

[39] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9610, 2020. 2

[40] Ziwei Zhang, Jun Liu, and Guangfeng Jiang. Spatial and temporal awareness network for semantic segmentation on automotive radar point cloud. *IEEE Transactions on Intelligent Vehicles*, 2023. 3

[41] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2, 3, 4, 7, 8

[42] Lianqing Zheng, Zhixiong Ma, Xichan Zhu, Bin Tan, Sen Li, Kai Long, Weiqi Sun, Sihan Chen, Lu Zhang, Mengyue Wan, et al. Tj4dradset: A 4d radar dataset for autonomous driving. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 493–498. IEEE, 2022. 3, 4

7. Appendix

In the Appendix section, we add more supplementary material in support of the thesis. We further analyze the

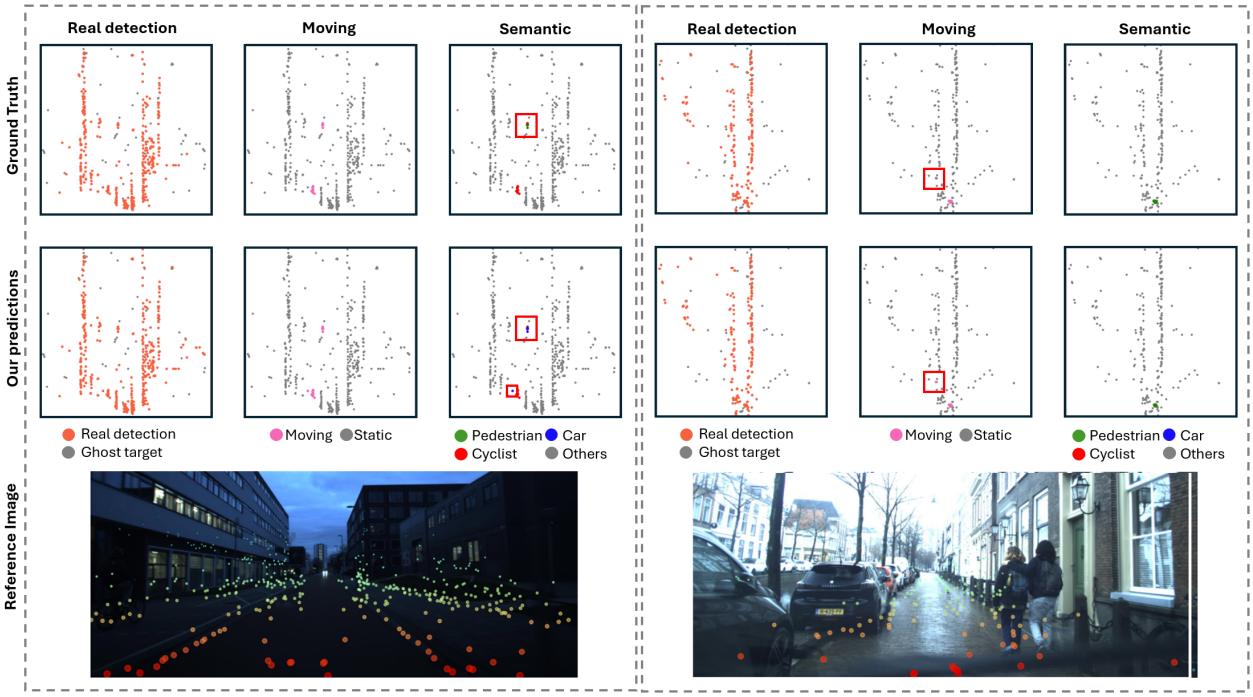

Figure 8. Examples of correct and incorrect segmentation results on our dataset. The top row is the ground truth for real detection, moving objects, and semantic segmentation. In the middle are the predictions from our network. At the bottom are the camera images with radar point clouds projected for reference. The red rectangles indicate the misclassifications.

quality of our auto-labeling pipeline and our model's prediction result with more example frames from the dataset.

7.1. Auto-labeling

Figure 9 shows a comparison of the moving objects annotation result using our auto-labeling pipeline and the trivial method ($|V_{\text{compensated}}| > 0.3\,\text{m/s}$ [24]. Compared with the trivial method, our method demonstrates a good ability to reduce false positives in noisy radar point clouds. It is also obvious from the camera image that the three-wheeler which is miss-detected by the bounding box is correctly compensated by the DBSCAN algorithm shown in the green box.

7.2. Result Analysis

Figure 8 shows the predictions result from our model. For the example on the left, the cyclist is hardly visible from the camera image, our model sees the moving object but falsely predicts it as a car. It is also worth pointing out that for this frame our auto-labeling pipeline falsely labels the cyclist as a bike and a pedestrian. For the second example, we can see the ghost target, and reflections to the right side of the road are clearly identified, as it is the building block on the right. The edges of the road can be clearly seen from the real detection plot. There is one miss classification in moving object prediction as shown in the red box, but it is later correctly identified in the semantic segmentation as a background point.

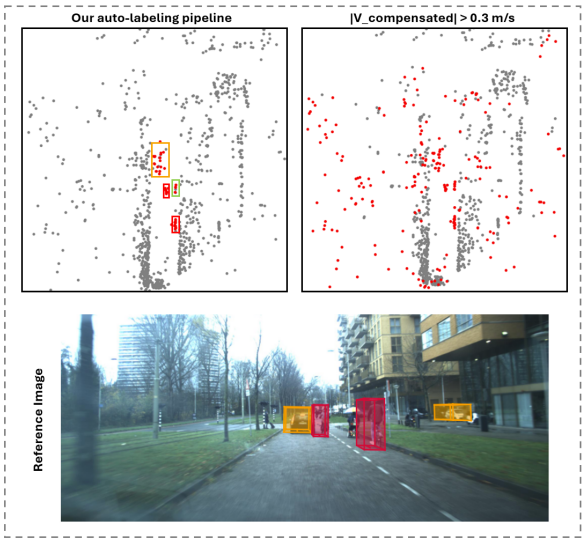

Figure 9. Comparison of our auto-labeling pipeline and the trivial method for labeling moving objects.