SemRaFiner: Panoptic Segmentation in Sparse and Noisy Radar Point Clouds

Matthias Zeller[1]   Daniel Casado Herraez[2]   Bengisu Ayan[3]
Jens Behley[4]   Michael Heidingsfeld[5]   Cyrill Stachniss[6]

Abstract—Semantic scene understanding, including the perception and classification of moving agents, is essential to enabling safe and robust driving behaviours of autonomous vehicles. Cameras and LiDARs are commonly used for semantic scene understanding. However, both sensor modalities face limitations in adverse weather and usually do not provide motion information. Radar sensors overcome these limitations and directly offer information about moving agents by measuring the Doppler velocity, but the measurements are comparably sparse and noisy. In this paper, we address the problem of panoptic segmentation in sparse radar point clouds to enhance scene understanding. Our approach, called SemRaFiner, accounts for changing density in sparse radar point clouds and optimizes the feature extraction to improve accuracy. Furthermore, we propose an optimized training procedure to refine instance assignments by incorporating a dedicated data augmentation. Our experiments suggest that our approach outperforms state-of-the-art methods for radar-based panoptic segmentation.

Index Terms—Semantic Scene Understanding, Deep Learning Methods

I. INTRODUCTION

AUTONOMOUS vehicles require reliable perception and semantic scene understanding to enable safe driving behaviour. In particular, dynamic traffic participants must be correctly identified to ensure secure autonomous mobility. While cameras and LiDARs enhance the overall scene understanding, these sensors face limitations under adverse weather. Radar sensors work under adverse weather conditions, including rain, fog and snow, overcoming the shortcomings of cameras and LiDARs. Furthermore, radar sensors provide additional information, such as the Doppler velocity, the relative radial velocity between the sensor and the detected object, and the radar cross section (RCS) value, which depends on the surface, the material and the shape of the target. However, radar measurements are sparse and affected by noise due to ego-motion and multi-path propagation.

In this paper, we tackle the problem of panoptic segmentation, combining instance and semantic segmentation, in sparse
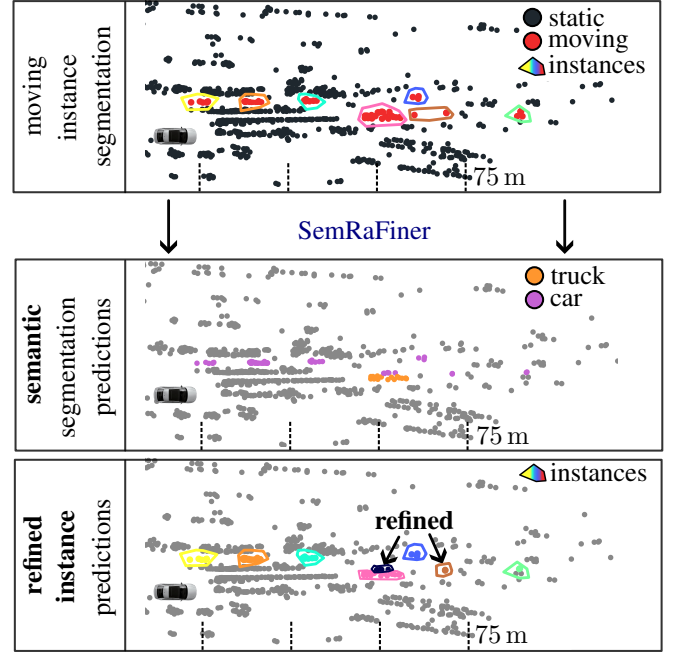
Fig. 1: Our method processes moving instance segmentation predictions (a) to predict point-wise semantic classes (b) and refine the instance assignment (c) to solve the panoptic segmentation task from sparse radar point clouds. In the image of the point cloud (a), each polygon represents a different instance of moving objects.

and noisy radar point clouds. We focus on moving instances, which first require the differentiation of the moving and static parts of the environment, such as parked and moving vehicles. In the second step, we separate the moving agents into individual instances within the surrounding and assign them a corresponding semantic class. This incorporates refining the semantic prediction to enhance the scene understanding, resulting in panoptic segmentation. Radar sensors are particularly suitable for identifying moving agents because they provide the Doppler velocity of the detection. We leverage moving instance predictions and propose a semantic instance refinement illustrated in Fig. 1. Compared to state-of-the-art approaches [28], [43], we are able to work on single-scan radar data and do not rely on scan aggregation, which can increase memory consumption and induce latencies, making it unsuitable for tasks requiring immediate feedback, such as collision avoidance.

The main contribution of this paper is a novel approach for accurate panoptic segmentation in sparse and noisy radar point clouds. Our approach, called SemRaFiner, leverages

the advantages of moving instance segmentation, which does not require differentiation between semantic classes but allows us to enhance the overall performance. We optimize our network and propose a new self-attention module, the radius transformer layer, to account for the changing density of radar point clouds, especially when processing moving instance predictions. We utilize dedicated data augmentation to further refine the instance prediction and enhance panoptic segmentation. We construct an efficient network incoporating the individual modules and an optimized training approach to improve the performance of the overall estimation task.

In sum, we make four key claims: First, our SemRaFiner shows state-of-the-art performance for panoptic segmentation in sparse and noisy radar point clouds. Second, our radius transformer layer enhances feature extraction, especially for processing the moving object prediction. Third, our optimized training process, including dedicated data augmentation to refine instances and semantic prediction, enhances the overall performance. Fourth, our proposed approach runs faster than the sensor frame rate. These claims are backed up by the paper and our experimental evaluation.

II. RELATED WORK

The panotic segmentation [14], [29] unifies semantic segmentation [5], [19], [31] and instance segmentation [16], [25], [34]. To provide a complete overview, we include different tasks in point cloud processing and structure the extensive related work between projection-based, voxel-based, point-based, transformer-based, and hybrid approaches.

Projection-based methods [3], [29], [31] project the point clouds into 2D range images to apply successful convolutional neural networks. These approaches are highly effective and profit from advanced image-based algorithms. However, the range representation results in occlusion artefacts among instances. To account for that, several approaches [10], [24] project the point cloud to a polar bird's eye view. PanopticPolarNet [47] adopts the representation and incorporates an adaptive point cloud pruning technique and instance augmentation to enhance panoptic segmentation. However, the 2D representation leads to back projection errors and discretization artefacts that harm accuracy [19].

Voxel-based approaches [15], [24] prevent back projection errors by keeping the 3D information of the point clouds. Since point clouds are unevenly distributed, the voxel representation results in empty voxels. To overcome latency issues, state-of-the-art methods [16], [34] apply submanifold sparse convolutions [9] to process occupied voxels and reduce the computational complexity. Further, hybrid approaches [14], [24] extend the voxel-based processing and include multiple point cloud representations to alleviate limitations and extract valuable features. Hence, Su et al. [30] adopted RPVNet [39] to include various representations to enhance fine-grained segmentation. Since voxels inherently introduce discretization artefacts, reducing information loss is beneficial, especially in sparse radar data.

Point-based methods [6], [23], [26] utilize the advantage of directly processing the unordered point cloud data. Qi et

al. [22] propose an innovative approach to use shared multi-layer perceptrons (MLPs) and max pooling to extract features. PointNet++ [23] extends this approach to capture strong local features by hierarchical feature extraction from larger local regions. Schumann et al. [26] optimize PointNet++ to process sparse radar data. However, for sparse radar data, the processing often relies on aggregated point clouds [26], [28], which can be disadvantageous because of the increased latencies and memory consumption.

Another way of extracting valuable point-wise features within local areas is kernel point convolutions [5], [32]. Gasperini et al. [8] extend the kernel point convolutions backbone with class-dependent post-processing to achieve precise instance segmentation. Nevertheless, the kernel point convolution approaches are often outperformed by attention-based approaches [45], which extract stronger dependency information that is helpful for sparse radar point clouds [41].

Transformer-based methods made tremendous progress within recent years and led to major improvements starting from natural language processing [33] to point-based scene understanding [12], [13], [38], [43]. Zhao et al. [45] propose vector attention [44] on point clouds based on k nearest neighbours to weight the individual feature channels to enable fine-grained feature extraction. PTv2 [36] and PTv3 [35] take this idea further and propose group vector attention and point cloud serialization, respectively. Lai et al. [38] utilize a stratified key-sampling strategy within a grid representation to increase the receptive fields of the attention mechanism. Recently, masked-based predictions [18], [25], [37] further enhanced the performance. To leverage the potential, several transformer-based approaches [41], [42] are optimized to process sparse and noisy radar data.

In contrast to the related work, we propose a novel method inspired by self-attention that accounts for the sparsity and changing density of radar point clouds. Our SemRaFiner directly utilizes moving instance predictions to leverage the advantage of the radar data to solve the task of panoptic segmentation. We incorporate point-based refinement to extract valuable features and enhance instance segmentation. Furthermore, we optimize the learning process to achieve state-of-the-art performance of panoptic segmentation in sparse and noisy radar data. This makes our approach stand out and different to the existing methods.

III. OUR APPROACH TO PANOPTIC SEGMENTATION

The goal of our approach is to achieve precise panoptic segmentation in sparse and noisy radar point clouds to enhance scene understanding of autonomous vehicles. Our SemRaFiner architecture, illustrated in Fig. 2, utilizes the prediction of the Radar Instance Transformer [40], [42] and extends the backbone to refine instance assignments and include semantic classes. Our network is a point-based transformer network that builds upon the self-attention mechanism [33]. We directly process the individual points using our radius transformer layer to enhance feature extraction by addressing the specific spatial distribution of radar points. We further enhance the instance prediction by enabling the network to account for
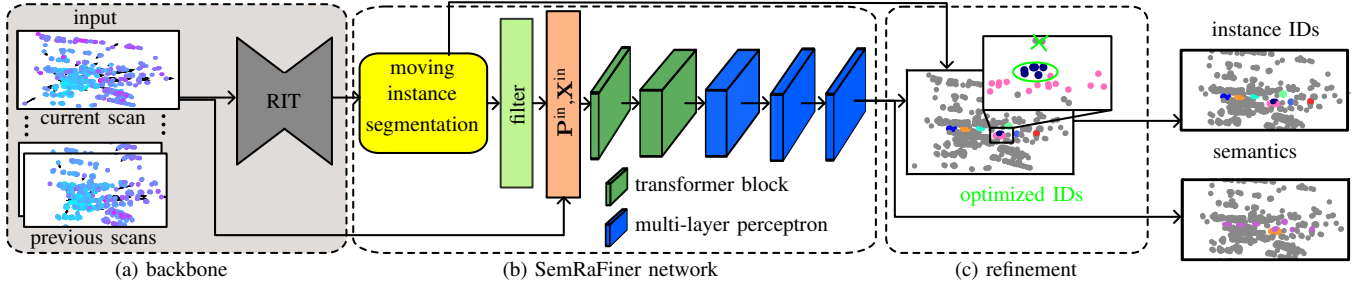
Fig. 2: The detailed design of the individual modules of our SemRaFiner. (a) The backbone provides the moving instance predictions. (b) Our network extracts valuable features from filtered moving instances. (c) The semantic prediction of our approach refines the instance assignment. The final output includes the semantic predictions and corrected instance IDs. The depth of the transformer blocks and multi-layer perceptrons correspond to the feature channels.

false predictions through our training procedure. The final point-based prediction includes the corresponding class and refined instance assignments.

A. Moving Instance Segmentation Backbone

Radar data is typically sparse and noisy, but it provides additional Doppler information that is useful for moving instance segmentation. Due to the enhanced performance of moving instance segmentation [40], [42] compared to standard semantic segmentation [41], our idea is to exploit this performance advantage to improve accuracy. Therefore, we utilize the state-of-the-art moving instance segmentation method Radar Instance Transformer (RIT) [40], [42] to reliably determine moving instances to improve panoptic segmentation.

RIT enriches the current radar point cloud $\mathcal{P}^t$ at time t, consisting of the coordinates of the points and the radar features, including the Doppler velocity and RCS values, with temporal information from T previous scans $\mathcal{P}^{t-T}, \ldots, \mathcal{P}^{t-1}$ within an optimized module to reduce computational complexity. The enriched point cloud is processed by transformer blocks [44] and attention-based graph clustering to determine the instance IDs $\mathcal{I} = \{I_1, \ldots, I_N\}$ with $I_i \in \mathbb{N}$. Additionally, the outputs include the moving object segmentation labels $\mathcal{S}^{\mathrm{MOS}}$. We utilize the predicted labels to filter the point clouds for moving prediction, which is the input to our method. Since we do not rely on extracted features of the backbone, we can potentially substitute the network for other moving instance segmentation approaches. However, the good performance of the RIT is crucial for our method to improve panoptic segmentation.

B. Radius Transformer Layer

In sparse radar point clouds, individual points can represent whole instances, such as distant vehicles or nearby pedestrians. Therefore, they contain important information for safe autonomous mobility. Hence, the feature extraction is key to enhancing performance for downstream tasks. Since we process the filtered point cloud, often including only moving instances, the point clouds are sparse and unevenly distributed compared to the unfiltered point cloud, including static points. To address this issue, especially for moving instance prediction, we propose a radius transformer, which combines a ball query sampling strategy with a vector attention mechanism. We argue that the most important information for

correctly classifying individual points is within the local area of the object. Most state-of-the-art approaches also focus on the local area. However, due to the filtering of the point clouds explained in Sec. III-A, individual instances are distributed over the whole field of view.

Performing attention between the k nearest neighbours [36], [42], [45] or determining the corresponding points with serialization algorithms [35] can result in interconnections of distant points, which is likely to harm the feature extraction capability for our type of problem. In contrast, our radius-based approach limits the considered related local area. Other approaches [13], [38] restrict the local area by performing attention on predefined areas. However, the grid representation is fixed, which does not necessarily align with the instances. Therefore, we propose a radius instance transformer layer, defining the local area for each point individually to overcome this shortcoming.

The input to our transformer layer contains the information about the moving instances within the single current scan $\mathcal{P}^t$ at time t. This comprises the point coordinates $\mathbf{P}^{\mathrm{in}} = [\mathbf{p}_1, \ldots, \mathbf{p}_{N^{\mathrm{mov}}}]^\top \in \mathbb{R}^{N^{\mathrm{mov}} \times 2}$ and the encoded point-wise features $\mathbf{X}^{\mathrm{in}} = [\mathbf{x}_1, \ldots, \mathbf{x}_{N^{\mathrm{mov}}}]^\top \in \mathbb{R}^{N^{\mathrm{mov}} \times D_1}$, where $\mathbf{p}_i \in \mathbb{R}^2$ and $\mathbf{x}_i \in \mathbb{R}^{D_1}$ for N^{mov} moving points. The features $\mathbf{X}^{\mathrm{in}}$ are the original radar features of the subset of points belonging to moving instances. During the training, we utilize the ground truth labels to select the corresponding moving points, and during inference, the points correspond to the semantic and instance predictions of the backbone. We do not process the static predictions. In our transformer layer, we first encode the features of the moving points $\mathbf{X}^{\mathrm{in}}$ as keys $\mathbf{K}$, queries $\mathbf{Q}$, and values $\mathbf{V}$ as follows:

$$\mathbf{Q} = \mathbf{X}^{\mathrm{in}} \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}^{\mathrm{in}} \mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}^{\mathrm{in}} \mathbf{W}_V, \quad (1)$$

where $\mathbf{W}_K$, $\mathbf{W}_Q$, and $\mathbf{W}_V \in \mathbb{R}^{D_1 \times D_1}$ are learned linear projections. To adaptively combine the features within the local areas and extract valuable features for individual points, we need to group the points. As mentioned before, to account for the changing density of points and replace the grid representation with a flexible solution, we sample the points within a circle defined by the radius r, around the individual points. We determine the neighbourhood of the corresponding points with point coordinates $\mathbf{p}$ by the relative position $\mathbf{r}_{i,j} = \mathbf{p}_i - \mathbf{p}_j$ where $\mathbf{p}_i$ and $\mathbf{p}_j \in \mathbf{P}^{\mathrm{in}}$. The domain of definition for our self-attention is then the circle $\mathcal{B}_r(\mathbf{p}_i) = \{\mathbf{p}_i \in \mathbf{P}^{\mathrm{in}} \mid \|\mathbf{r}_{i,j}\| \leq r\}$

with radius r, similar to the ball query of PointNet++ [23]. However, we replace the processing within the local areas of PointNet++ with a powerful attention mechanism. Furthermore, we observe that having many interconnections between points can make it difficult to extract valuable features within local regions, consisting of different semantic classes, due to the information exchange within the attention mechanism. Hence, we restrict the maximum number of points N^{max} within the radius, resulting in an optimized neighbourhood representation to enhance feature extraction.

We sample the points using a ball query [32] to extract the related queries, keys, and values, resulting in $\mathbf{Q}^{\text{r}}, \mathbf{K}^{\text{r}}$, and $\mathbf{V}^{\text{r}} \in \mathbb{R}^{N^{\text{mov}} \times N^{\text{max}} \times D_1}$ within the local neighbourhoods. If the local area around the points contains less than N^{max} points, the remaining keys, queries, and values are zero-padded. Furthermore, we utilize the relative position $\mathbf{r}_{i,j}$, which we calculate to determine the local areas to calculate the relative positional encoding. We process the relative positions $\mathbf{r}_{i,j}$ by two linear layers with weight matrices $\mathbf{W}_1^p \in \mathbb{R}^{2 \times 2}$ and $\mathbf{W}_2^p \in \mathbb{R}^{2 \times D_1}$, batch normalization [11], and the ReLU activation function [20] to determine the relative positional encoding $\mathbf{R} \in \mathbb{R}^{N^{\text{mov}} \times N^{\text{max}} \times D_1}$ [45]. We adopt vector attention [44] and subtract the keys from the queries, to calculate the attention weights $\mathbf{A} \in \mathbb{R}^{N^{\text{mov}} \times N^{\text{max}} \times D_1}$ as follows:

$$\mathbf{A}_{i,j} = (\mathbf{Q}_{i,j}^{\text{r}} - \mathbf{K}_{i,j}^{\text{r}}) + \mathbf{R}_{i,j}, \tag{2}$$

where we add the relative positional encoding $\mathbf{R}$ to include fine-grained position information in the weighting. We process the resulting attention weights by a multi-layer perceptron (MLP), including two linear layers with weight matrices $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{D_1 \times D_1}$, two batch normalization [11] layers, and ReLU activation [20]. We then apply the softmax function to derive the final attention weights $\hat{\mathbf{A}}_{i,j}$. The output features $\mathbf{X}^{\text{out}} \in \mathbb{R}^{N^{\text{mov}} \times D_1}$ of the transformer layer are the sum of the element-wise multiplication, indicated by $\odot$, and the relative positional encoding, determined as follows:

$$\mathbf{X}_i^{\text{out}} = \sum_{j=1}^{N^{\text{max}}} \hat{\mathbf{A}}_{i,j} \odot (\mathbf{V}_{i,j}^{\text{r}} + \mathbf{R}_{i,j}). \tag{3}$$

Besides the features, which comprise the information within the local area, the output of the transformer layer includes the point coordinates $\mathbf{P}^{\text{in}}$ of the moving points. The point coordinates are kept as is to include fine-grained position information within the consecutive layers.

C. SemRaFiner Network

We aim for an efficient processing of the radar data to reduce latencies. Consequently, our SemRaFiner includes only two transformer blocks and three multi-layer perceptrons, to derive the refined instance IDs $\mathcal{I}^r = \{I_1^r, \ldots, I_N^r\}$ with $I_i^r \in \mathbb{N}$ and semantic labels $\mathcal{S}^{\text{sem}}$, see Fig. 2.

We embed our radius transformer layer into a residual transformer block to extract valuable features, see Fig. 3. We first increase the dimension of the features $\mathbf{X}^{\text{in}}$, which we then process within the radius transformer layer. The output features of the radius transformer layer are added to the features of the skip connection and processed by the
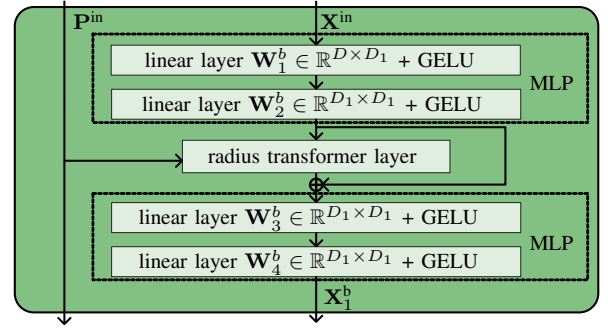

Fig. 3: Our transformer block is a residual block that includes our radius transformer layer and two multi-layer perceptrons with two linear layers and GELU activation functions. The input includes the point coordinates $\mathbf{P}^{\text{in}}$ and the point-wise features $\mathbf{X}^{\text{in}}$.

second MLP. The point coordinates are directly passed into the transformer layer to calculate the positional encoding to include fine-grained position information within the local areas. The resulting output includes the point-wise features and coordinates to keep position information intact. The resulting feature after the second transformer block $\mathbf{X}_2^{\text{b}} \in \mathbb{R}^{N^{\text{mov}} \times D_2}$ are processed by the three consecutive MLPs, illustrated by the blue blocks in Fig. 2, to reduce the dimensionality and derive the final predictions. The first two MLPs consist of two linear layers, where the first linear layer keeps the dimension of the features, and the second reduces the dimension by a factor of two. To derive the final semantic predictions, the last MLP first reduces the dimensionality by a factor of two before predicting the semantic labels. The final semantic prediction $\mathcal{S}^{\text{sem}} = \{s_1^{\text{sem}}, \ldots, s_N^{\text{sem}}\}$ where $s_i^{\text{sem}} \in \{1, \ldots, C\}$ combine the static predictions of the backbone and the refined semantic class predictions.

D. Data Augmentation

We utilize ground truth annotations to train our model to classify instances within sparse radar point clouds reliably. During inference, we leverage the moving instance prediction to perform point-wise semantic segmentation and refine the instance prediction. The major problem is that the ground truth does not include failure cases such as static points predicted as moving. To account for the false prediction of static points, we augment our training data. Our idea is to enable our network to correct false predictions, which cannot be learnt by only considering the ground truth annotations. Therefore, we incorporate static points into the training process and add static points close to an instance by the probability of p_I. This approach also aligns with the false predictions, which often occur at the boundary of the instance. Additionally, we observe that the moving instance predictions include false predictions for small instances within the individual scans. Therefore, we add instances comprising only static ground truth annotations, including one to five points by the probability of p_S.

E. Instance Refinement

Our data augmentation enables us to correct false predictions and remove static points from instance predictions. Furthermore, we utilize the semantic predictions to correct

instance assignments. Since the classification of moving and static points does not include further differentiation, we observe that nearby instances belonging to different semantic classes, such as trucks and cars, may get clustered together. However, our panoptic segmentation incorporates that knowledge, and hence, we can resolve the false assignments. We combine the instance assignments and semantic predictions within our instance refinement module. Due to the point-wise processing, we derive a semantic class for each individual radar point. We combine these predictions and the instance assignment per point and assign different instance IDs if the semantic classes differ. The overall refinement thus accounts for false semantic predictions and instance assignments.

F. Implementation Details

We train the backbone separately and adopt the original parameters [42]. We implemented our SemRaFiner in Py-Torch [21] and trained our network with one Nvidia A100 GPU and a batch size of 64 over 80 epochs. As optimizer, we utilize AdamW [17], set the initial learning rate to 0.001 and drop the learning rate by a factor of 10 after 60 epochs. We combine the Lovász loss [2] and cross-entropy loss to learn the point-wise classification using the same weighting for all classes.

Additionally, we introduce a consistency loss to enforce the same class prediction for the individual instances. We construct a loss based on the semantic predictions $\mathcal{I}_h^{\text{sem}}$ for the individual instances h. The ideal case is that each instance belongs to the same class. However, this limits the performance since we cannot correct false predictions during instance refinement. Therefore, it is important to maintain point-wise classification while preserving consistency within the true instances. We utilize the number of distinct classes within an instance $|\mathcal{I}_h^{\text{sem}}|$ to calculate the loss over the number of all instances N_h as follows:

$$\mathcal{L}_c = \frac{1}{N_h} \sum_{h=1}^{N_h} 1 - \frac{1}{|\mathcal{I}_h^{\text{sem}}|}. \tag{4}$$

We combine all three losses without weighting to derive the final training loss for our network.

We process the input features with dimension $D = 5$, comprising the point coordinates (x_i^C, y_i^C, z_i^C), the radar cross section σ_i, and the ego-motion-compensated Doppler velocity v_i, resulting in a five-dimensional input vector $\mathbf{x}_i = (x_i^C, y_i^C, z_i^C, \sigma_i, v_i)$. The coordinate $z_i^C = 0$ is added to apply the pose compensation. We increase the per-point features to $D_1 = 64$, $D_2 = 256$ within our transformer blocks. We set the parameters for our radius transformer layer to $r = 5.0\,\text{m}$ and the maximum considered neighbours within the ball query to $N^{\text{max}} = 24$. The first two MLPs reduce the dimensionality to 128 and 64, respectively. In the third MLP, the first linear layer has an output dimension of 32 before predicting C classes by the second linear layer. We set the probabilities of augmenting the instances p_I and the scan p_S to $40\,\%$.

IV. EXPERIMENTAL EVALUATION

The main focus of this work is to achieve accurate panoptic segmentation in sparse radar point clouds. We present our

experiments to show the capabilities of our method. The results of our experiments support our key claims, including that our approach achieves state-of-the-art performance in panoptic segmentation without aggregating scans. Moreover, our radius instance transformer layer accounts for the changing density in sparse radar data to enhance panoptic segmentation. Our dedicated data augmentation and point-wise processing enables instance refinement to improve overall performance. Our SemRaFiner is efficient and only adds a small overhead to the backbone to incorporate semantic classes.

A. Experimental Setup

We train and evaluate our model on the RadarScenes [27] dataset since it is the only large-scale open-source radar dataset [46] that includes per-point annotations for moving instance segmentation and panoptic segmentation under different weather conditions and driving scenarios. As described by Zeller et al. [42], we split the 158 sequences into 130 sequences for training, 6 for validation, and 22 for testing. We perform all ablation studies on the validation set. We adopt the sparse representation of the single scan data processing [42] containing up to four radar sensors to cover the surroundings of the vehicle.

We utilize the panoptic quality (PQ) [1] to evaluate the panoptic segmentation. Additionally, we report the mean intersection over union (mIoU) to evaluate the semantic segmentation performance in detail. To enable a detailed evaluation, we further differentiate between the six classes, including static, pedestrian (ped.), pedestrian group (ped. grp.), car, truck, and bike.

B. Panoptic Segmentation

The first experiment evaluates the performance of our approach and its outcomes support the claim that our approach achieves state-of-the-art performance in panoptic segmentation in sparse and noisy radar scans. We compare our method to high-performing networks with strong performance in point-based instance segmentation and radar segmentation. We furthermore utilize Mask3D [25] as a baseline and extend Gaussian Radar Transformer (GRT) [41] with DBSCAN [7] to reliably cluster instances. We extend Mask3D [25] with state-of-the-art post-processing for mask predictions [4] and set the score threshold to 0.5 to enhance performance and derive the panoptic labels. Additionally, we report the segmentation results of RadarPNv1 [26], RadarPNv2 [28], and STA-Net [43], which rely on scan aggregation and do not consider single-scan processing.

Our SemRaFiner outperforms the existing single-scan approaches, especially in terms of PQ, by a solid margin, as displayed in Tab. I. The performance improves, especially for small instances comprising only a few points such as the pedestrians. We observe that our dedicated network, which utilizes the moving instance segmentation as input, is able to extract valuable features, particularly for smaller classes, due to the reduced class imbalance and optimized architecture. We emphasize that our performance depends on the prediction of

Method	Input	PQ	mIoU	PQ						IoU					
				static	car	ped.	ped. grp.	bike	truck	static	car	ped.	ped. grp.	bike	truck
RadarPNv1 [26]		-	61.0	-	-	-	-	-	-	98.7	58.2	36.0	58.7	58.4	56.1
RadarPNv2 [28]	multi-scan	-	61.9	-	-	-	-	-	-	98.7	63.8	38.8	58.5	51.0	61.0
STA-Net [43]		-	**70.4**	-	-	-	-	-	-	**99.8**	**75.9**	**43.1**	**82.0**	**67.1**	54.6
Mask3D [25]		56.9	56.1	98.4	70.6	18.4	31.9	57.7	64.3	98.6	68.7	11.8	35.9	57.7	63.9
GRT [41] + DBSCAN [7]	single-scan	56.3	56.9	98.7	59.0	29.7	38.3	54.6	57.3	98.7	58.5	21.7	46.0	54.6	61.8
Ours		**81.4**	**70.4**	**99.7**	**85.7**	**59.2**	**83.1**	**78.9**	**82.1**	99.4	74.9	42.1	71.0	65.2	**69.6**

TABLE I: Panoptic segmentation results on the RadarScenes test set in terms of PQ and IoU scores. The results of RadarPNv1 [26], RadarPNv2 [28] and STA-Net [43] are adapted from the original papers. The gray rows correspond to multi-scan segmentation methods.

#	Model	$r\,[m]$	N^{max}	PQ	mIoU
[A]	Stratified Transformer [13]			55.0	42.6
[B]	Point Transformer [44]			75.8	63.1
[C]	KPConv [32]			81.0	69.6
[D]	Ours	5.0	24	**83.0**	**72.2**
[E]	Ours	6.0	24	82.3	71.3
[F]	Ours	6.0	34	82.4	71.5
[G]	Ours	5.0	34	82.7	72.0
[H]	Ours	3.0	24	81.7	70.4
[I]	Ours	5.0	6	78.7	66.4

TABLE II: Ablation study results on the RadarScenes validation set in terms of PQ and mIoU scores.

the RIT, and therefore, precise moving instance segmentation is important to enhance the accuracy.

Another way to enhance segmentation performance is by aggregating radar point clouds. However, the aggregation of point clouds can induce disadvantageous latencies, which is relevant when performing our approach on a vehicle in real time. Additionally, our method performs on par in terms of mIoU with the best-performing semantic segmentation model STA-Net, which aggregates scans over $500\,\mathrm{ms}$ with input point clouds of 3072 points. In contrast, our single scans include an average number of points of 539 in the test set. The RIT utilizes two previous scans to enrich the current scan, but only the single current scan is processed within the backbone. Therefore, we achieve similar results using five times fewer points and also include instance segmentation.

GRT achieves a higher mIoU compared to Mask3D for the single-scan approaches. We assume that the optimized radar-specific architecture performs better on radar data. However, the powerful mask predictions lead to better performance in terms of PQ but come with a higher computational cost. Overall, both methods struggle to reliably segment instances in sparse and noisy radar point clouds. We enhance the PQ for all classes, leading to an improvement of more than 20 percentage points on the overall PQ.

C. Ablation Study on the Radius Transformer Layer

The second experiment evaluates our radius transformer layer and illustrates that our approach is capable of extracting valuable features from sparse point clouds, enhancing the overall performance. We replace the radius transformer layer with the point transformer layer [44] used for radar signal processing [42], the Stratified transformer layer [13], and KPConv [32], illustrating the ablation results in Tab. II. Additionally, we evaluate the performance of our approach based on the selection of the hyperparameters, namely the radius r and the maximum number of neighbours N^{max}.

The gain obtained by our radius transformer layer is directly visible by the increase compared to the other transformer
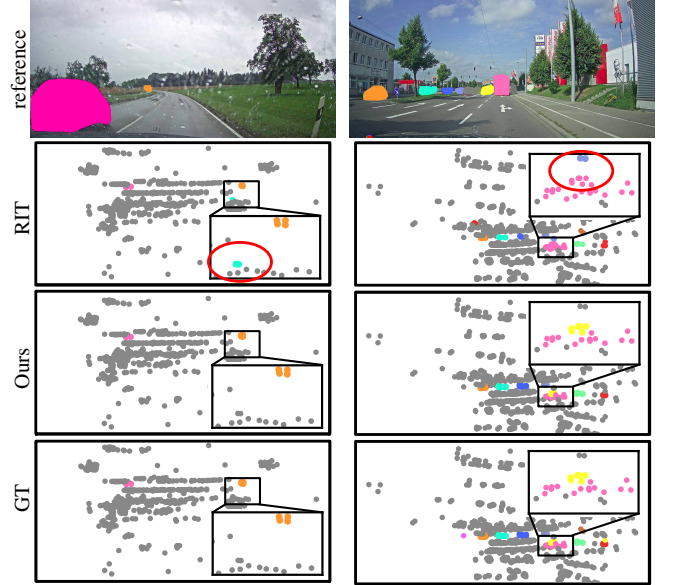

Fig. 4: Qualitative results of our SemRaFiner and the Radar Instance Transformer (RIT) [42] on the RadarScenes dataset. The camera images are anonymized and just shown for reference. The left column is from sequence 79 (fog), and the right is from sequence 107 (urban environment). Each colour in the image of the predictions represents a different instance (static is grey).

modules, enhancing the performance by more than 6 absolute percentage points compared to the point transformer layer. We argue that the kNN-based processing results in the interconnection of distant points, and the grid representation does not align with the instances, both harming the accuracy. As expected, the KPConv approach, Tab. II [C], leads to the best result using the ball query to determine the local areas. However, our attention-based processing enhances the PQ by 2 absolute percentage points compared to KPConv.

Furthermore, the comparison of the setup of $r = 5\,\mathrm{m}$ and $N^{\max} = 6$, Tab. II [I], which utilizes the same number of neighbours as the point transformer, underlines that the additional neighbourhood restrictions by the radius are advantageous. We argue that especially for small instances such as pedestrians but also far away cars or bikes lead to too many interconnections harm the accuracy. However, for larger instances, we observe that increasing the number of neighbours helps to classify them reliably.

The performance decreases if we increase the maximum number of points from 24 to 34, Tab. II [D] and [G], respectively. We assume that too many interactions result in a mix of information during self-attention, which hinders the clear separation of the information. We also see that for a larger number of neighbours, the performance of smaller instances, such as pedestrians, drops from $65.2\,\%$ to $64.0\,\%$ for $N^{max} = 24$ and

Model	PQ
Ground truth semantics	77.1
Ours w/o data augmentation	81.5
Ours	**83.0**

TABLE III: Panoptic segmentation result on the RadarScenes validation set in terms of PQ and mIoU scores.

Model	parameters (M)	mean runtime (ms)
STA-Net [43]	7.4	34.9
RadarPNv1 [26]	1.8	278.1
Mask3D [25]	39.6	85.2
Gaussian Radar Transformer [41]	8.4	24.0
Radar Instance Transformer [42]	3.8	31.7
Ours + RIT [42]	4.5	42.1

TABLE IV: The evaluation of the mean runtime and the number of parameters of the models on an Nvidia RTX A6000 GPU based on 1000 randomly sampled point clouds of the RadarScenes data set. The runtime and the parameters of STA-Net and RadarPNv1 are copied from Zhang et al. [43] using a different hardware setup.

$N^{max} = 34$, respectively. Since close instances might still be included within the radius, the performance of small instances, especially, is affected. Therefore, the radius and the maximum number of parameters depend on the task and the segmentation of the specific instances. Additionally, the parameters may depend on the resolution of the radar sensor, which is an interesting starting point for future research to automatically determine the optimal settings. Overall, both parameters are essential to enhance radar-based panoptic segmentation.

D. Ablation Study on the Instance Refinement

The third experiment supports the claim that our method incorporates efficient instance refinement to enhance overall performance, as depicted in Tab. III. The central parts that enable instance refinement are data augmentation, including static points and point-wise processing. To verify the assumptions, we first evaluate the performance of the moving instance segmentation prediction in combination with the semantic ground truth labels. We, therefore, do not optimize the instance prediction of the Radar Instance Transformer (RIT) to illustrate the limitation of keeping the instance assignments under perfect segmentation results. In the second step, we remove the data augmentation introduced in Sec. III-D. The resulting method accounts for false predictions within the different object classes but cannot correct false static predictions because these errors are excluded in the training process. However, our instance refinement helps to optimize the instance prediction compared to RIT even under the condition that our semantic predictions are not perfect. Furthermore, the data augmentation enhances the panoptic quality by more than two absolute percentage points. Therefore, data augmentation and point-wise processing are essential to improve performance, especially for the annotated moving objects in RadarScenes. Furthermore, we removed the consistency loss for the class consistency within instances, which resulted in a PQ of $82.9\,\%$, leading to a small decrease in performance. Furthermore, our SemRaFiner works reliably under versatile scenes, as illustrated in Fig. 4. Our approach reliably refines the moving instances under different and changing scenarios.

E. Runtime

Finally, we analyze the runtime and the number of parameters of the approaches and, in this way, support the claim that our approach runs fast enough to support online processing. We, therefore, tested our approach on an Nvidia RTX A6000 GPU on 1,000 real-world radar scans that we randomly selected from the RadarScenes validation set. The mean runtime for one radar scan is illustrated in Tab. IV. The Gaussian Radar Transformer has the lowest runtime utilizing optimized farthest point sampling and kNN algorithm in C++ [45]. The multi-scan approaches increase the runtime compared to the Gaussian Radar Transformer, but the comparison is difficult since the reported results are evaluated on different hardware. Additionally, Mask3D uses four times more parameters to enhance the panoptic segmentation. The Radar Instance Transformer and our approach combined still have fewer parameters compared to STA-Net since our model only adds a small overhead. The combined runtime of 42.1 ms, equal to 24 Hz, is faster than the frame rate of 17 Hz of the radar sensors.

In summary, our evaluation suggests that our method provides competitive panoptic segmentation in sparse radar point clouds. The instance refinement and the utilization of moving instance predictions outperform state-of-the-art approaches. Thus, we support all our claims made in this paper through our experimental evaluation.

V. CONCLUSION

In this paper, we presented a novel approach for panoptic segmentation of sparse and noisy radar point clouds. Our method exploits the advantages of moving instance segmentation to refine the predictions and enhance the overall system performance. Our method efficiently exploits the sparse representation of moving points by limiting the self-attention mechanism within a local neighbourhood. We overcome the limitations of grid representations and unbounded local regions in transformer networks to predict the semantic classes. Our point-wise processing enables us to refine instance prediction and correct assignment errors. This allows us to successfully identify individual instances and enhance feature extraction by optimizing the training procedure. We implemented and evaluated our approach on the RadarScenes dataset and provided comparisons to other existing techniques. The experiments suggest that our proposed modules and training strategy are essential to achieve good performance on panoptic segmentation, supporting all claims made in this paper. Overall, our approach outperforms the state-of-the-art approaches, taking a step forward towards sensor redundancy for semantic segmentation for autonomous robots and vehicles.

REFERENCES

[1] J. Behley, A. Milioto, and C. Stachniss. A Benchmark for LiDAR-Based Panoptic Segmentation Based on KITTI. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[2] M. Berman, A.R. Triki, and M.B. Blaschko. The Lovász-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss. 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, 2021.

[4] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] C. Chi, P. Li, X. Chen, and X. Yang. PUA-MOS: End-to-End Point-wise Uncertainty Weighted Aggregation for Moving Object Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[6] A. Dubey, A. Santra, J. Fuchs, M. Lübke, R. Weigel, and F. Lurz. HARadNet: Anchor-free target detection for radar point clouds using hierarchical attention and multi-task learning. *Machine Learning with Applications (MLWA)*, 8:100275, 2022.

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996.

[8] S. Gasperini, M.N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. Panoster: End-To-End Panoptic Segmentation of LiDAR Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3216–3223, 2021.

[9] B. Graham, M. Engelcke, and L. van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] S. Huang, Z. Gojcic, J. Huang, A. Wieser, and K. Schindler. Dynamic 3D Scene Analysis by Point Cloud Accumulation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[11] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2015.

[12] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu. Rethinking range view representation for lidar segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision*, 2023.

[13] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia. Spherical transformer for lidar-based 3d recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] J. Li, X. He, Y. Wen, Y. Gao, X. Cheng, and D. Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia. Unifying voxel-based representation with transformer for 3d object detection. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[16] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision*, 2021.

[17] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2017.

[18] R. Marcuzzi, L. Nunes, L. Wiesmann, E. Marks, J. Behley, and C. Stachniss. Mask4D: End-to-End Mask-Based 4D Panoptic Segmentation for LiDAR Sequences. *IEEE Robotics and Automation Letters (RA-L)*, 8(11):7487–7494, 2023.

[19] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss. Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7503–7510, 2022.

[20] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2010.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

[22] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] C.R. Qi, L. Yi, H. Su, and L.J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[24] H. Qiu, B. Yu, and D. Tao. GFNet: Geometric Flow Network for 3D Point Cloud Semantic Segmentation. *Trans. on Machine Learning Research (TMLR)*, 2022.

[25] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe. Mask3D for 3D Semantic Instance Segmentation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[26] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler. Semantic Segmentation on Radar Point Clouds. In *Proc. of the Intl. Conf. on Information Fusion*, 2018.

[27] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J.F. Tilly, J. Dickmann, and C. Wöhler. RadarScenes: A real-world radar point cloud data set for automotive applications. In *Proc. of the Intl. Conf. on Information Fusion*, 2021.

[28] O. Schumann, J. Lombacher, M. Hahn, C. Wöhler, and J. Dickmann. Scene Understanding With Automotive Radar. *IEEE Trans. on Intelligent Vehicles*, 5(2):188–203, 2019.

[29] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada. EfficientLPS: Efficient LiDAR Panoptic Segmentation. *IEEE Trans. on Robotics (TRO)*, 38(3):1894–1914, 2021.

[30] S. Su, J. Xu, H. Wang, Z. Miao, X. Zhan, D. Hao, and X. Li. Pups: Point cloud unified panoptic segmentation. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2023.

[31] J. Sun, Y. Dai, X. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen. 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)*, 2022.

[32] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[34] T. Vu, K. Kim, T.M. Luu, T. Nguyen, and C.D. Yoo. SoftGroup for 3D Instance Segmentation on Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] X. Wu, L. Jiang, P.S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao. Point Transformer V3: Simpler, Faster, Stronger. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[36] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[37] Z. Xiao, W. Zhang, T. Wang, C.C. Loy, D. Lin, and J. Pang. Position-guided point cloud panoptic segmentation transformer. *arXiv preprint*, arXiv:2303.13509, 2023.

[38] L. Xin, L. Jianhui, J. Li, W. Liwei, Z. Hengshuang, L. Shu, Q. Xiaojuan, and J. Jiaya. Stratified Transformer for 3D Point Cloud Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[39] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision*, 2021.

[40] M. Zeller, D. Casado Herraez, J. Behley, M. Heidingsfeld, and C. Stachniss. Radar Tracker: Moving Instance Tracking in Sparse and Noisy Radar Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[41] M. Zeller, J. Behley, M. Heidingsfeld, and C. Stachniss. Gaussian Radar Transformer for Semantic Segmentation in Noisy Radar Data. *IEEE Robotics and Automation Letters (RA-L)*, 8(1):344–351, 2023.

[42] M. Zeller, V.S. Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss. Radar instance transformer: Reliable moving instance segmentation in sparse radar point clouds. *IEEE Trans. on Robotics (TRO)*, 40:2357–2372, 2024.

[43] Z. Zhang, J. Liu, and G. Jiang. Spatial and Temporal Awareness Network for Semantic Segmentation on Automotive Radar Point Cloud. *IEEE Trans. on Intelligent Vehicles*, 9(2):3520–3530, 2024.

[44] H. Zhao, J. Jia, and V. Koltun. Exploring self-attention for image recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] H. Zhao, L. Jiang, J. Jia, P.H. Torr, and V. Koltun. Point Transformer. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision*, 2021.

[46] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors*, 22(11):4208, 2022.

[47] Z. Zhou, Y. Zhang, and H. Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.