Radar Instance Transformer: Reliable Moving Instance Segmentation in Sparse Radar Point Clouds

Matthias Zeller *Student Member, IEEE*   Vardeep S. Sandhu   Benedikt Mersch *Student Member, IEEE*
Jens Behley *Member, IEEE*   Michael Heidingsfeld   Cyrill Stachniss *Member, IEEE*

Abstract—The perception of moving objects is crucial for autonomous robots performing collision avoidance in dynamic environments. LiDARs and cameras tremendously enhance scene interpretation but do not provide direct motion information and face limitations under adverse weather. Radar sensors overcome these limitations and provide Doppler velocities, delivering direct information on dynamic objects. In this paper, we address the problem of moving instance segmentation in radar point clouds to enhance scene interpretation for safety-critical tasks. Our Radar Instance Transformer enriches the current radar scan with temporal information without passing aggregated scans through a neural network. We propose a full-resolution backbone to prevent information loss in sparse point cloud processing. Our instance transformer head incorporates essential information to enhance segmentation but also enables reliable, class-agnostic instance assignments. In sum, our approach shows superior performance on the new moving instance segmentation benchmarks, including diverse environments, and provides model-agnostic modules to enhance scene interpretation. The benchmark is based on the RadarScenes dataset and is available at https://doi.org/10.5281/zenodo.10203864.

Index Terms—Radar Perception, Semantic Scene Understanding, Object Detection, Segmentation and Categorization, Deep Learning in Robotics and Automation

I. INTRODUCTION

THE safe navigation of autonomous vehicles in real-world environments depends on the reliable identification of moving objects. The knowledge about which part of the environment is moving and how many agents are present is essential for reliable future state prediction and path planning. This holds for mobile robots, offroad vehicles, and self-driving cars alike. Often, a redundant sensor setup of autonomous vehicles with different modalities is used and aims to reduce the risk of critical malfunctions. The advantages of individual sensors such as cameras, LiDARs, and radars should be considered to enhance the overall scene interpretation. Radar scans are affected by noise due to multi-path propagation, ego-motion, and sensor noise. At the same time, radar sensors are more robust to adverse weather conditions than cameras and LiDARs, thus essential for safe autonomous mobility. Furthermore, radar sensors measure the Doppler velocity and provide the radar cross section values, which depend on the material, the surface, and the geometry of the so-called

Matthias Zeller and Vardeep S. Sandhu are with CARIAD SE and with the University of Bonn, Germany. Jens Behley and Benedikt Mersch are with the University of Bonn, Germany. Michael Heidingsfeld is with CARIAD SE, Germany. Cyrill Stachniss is with the University of Bonn, with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.
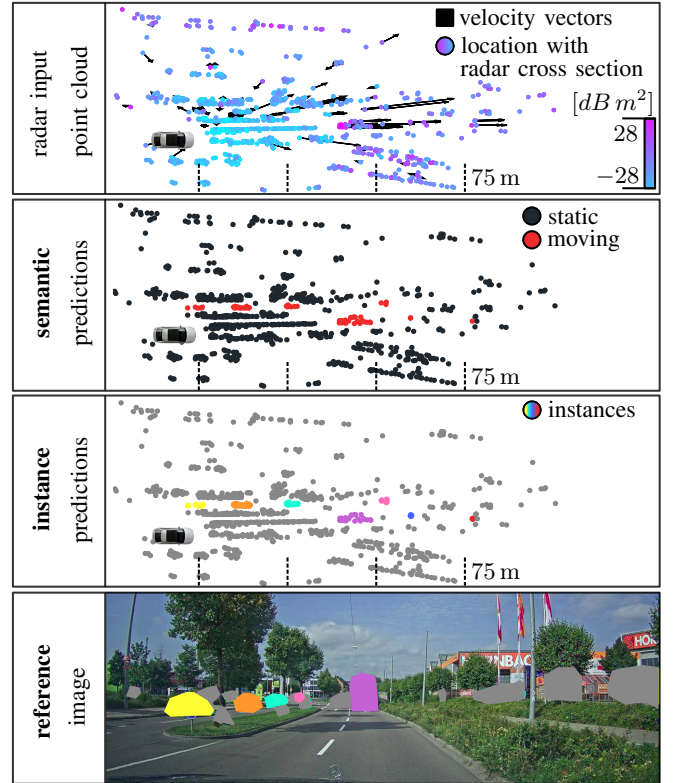
Fig. 1: Our method combines moving object segmentation (b) and instance segmentation (c) to solve the panoptic task of moving instance segmentation from sparse radar point clouds (a). The reference image in (d) illustrates the scene and includes privacy-preserving colored masks. In the image of the point cloud (c), each color represents a different instance of moving objects (static is grey). The colors in (c) and (d) correspond if the object is visible.

radar detections, which could help to identify moving agents accurately [87].

In this paper, we tackle the problem of moving instance segmentation in sparse and noisy radar point clouds. This requires differentiating between moving and static parts of the surrounding and separating individual agents. The task combines instance segmentation and moving object segmentation to a panoptic problem, including the differentiation of object instances, such as parked from moving cars. For many tasks, the information if an object is moving in combination with the velocities is often sufficient, substantially simplifying the labeling efforts. For radar data, the identification of the corresponding class is especially difficult and relies on synchronized LiDAR or camera data. However, semantic classes of the objects include additional information that can help

to differentiate between nearby instances and enable class-dependent clustering. While semantic information could be exploited to reason about the mobility of objects, it also entails the so-called "long-tail" problem; i.e., no matter how much data we collect, we will always have classes that are underrepresented or not even covered by the training data. Moving or non-moving instances are essential for autonomous driving and allow it to potentially cover the aforementioned long-tail classes.

State-of-the-art methods often address the task of moving instance segmentation separately as moving object segmentation [12] and instance segmentation [43]. Moving object segmentation approaches pass aggregated scans [12, 34] through the whole network, which induces latency and is disadvantageous for a task requiring immediate feedback such as collision avoidance. Recent instance segmentation approaches [62, 72, 80] work on single inputs, neglecting the temporal information, and do not differentiate between moving and static objects. Our approach aims to overcome this.

The main contribution of this paper is a novel approach that combines moving object segmentation and instance segmentation within a single network and accurately predicts moving instances in sparse and noisy radar point clouds. Our approach, called Radar Instance Transformer, predicts for each point in the input radar scan if it is moving or static and assigns an instance ID to each moving detection. To reliably identify moving instances, we efficiently incorporate the temporal information within the single current scan by our sequential attentive feature encoding module without passing aggregated scans through the whole network. We optimize the network architecture by processing the point cloud with the original resolution throughout the network to keep as much information as possible and enrich the points with high-level features. We utilize local and global attention to include instance information and propose a graph-based instance assignment to improve performance. In contrast to typical LiDAR approaches, our individual modules tackle the challenges of the sparsity and noise of radar point cloud properties that make radar data interpretation comparably hard. Besides, we extend the RadarScenes [64] dataset and transfer it into the first moving instance segmentation benchmark for point clouds.

In sum, we make five claims: First, our approach shows state-of-the-art performance for moving instance segmentation in sparse and noisy radar point clouds without passing multiple scans through the whole network. Second, our sequential attentive feature encoding extracts valuable temporal information by enriching the features of individual points to enhance accuracy. Third, our attentive instance transformer head is able to incorporate essential instance information, which improves the overall performance. Fourth, our attention-based graph partitioning enhances instance assignments without requiring class-dependent information. Fifth, our backbone enables valuable feature extraction for sparse radar point clouds by processing the full-resolution point cloud, i.e., the original number of points throughout the network.

II. RELATED WORK

Moving instance segmentation in point clouds unifies moving object segmentation [15, 34, 36, 46, 48, 58, 69] and instance segmentation [11, 28, 42, 51, 62, 72] and can be solved as panoptic segmentation [40, 60, 68]. Besides, the task also benefits from semantic segmentation [58, 61, 82, 83, 84, 89, 93] and scene flow estimation [4, 20, 19, 35, 67, 76]. To briefly summarize the extensive literature, including the different tasks, we distinguish between projection-based, voxel-based, point-based, and transformer-based approaches with and without considering temporal information.

Projection-based methods [12, 18, 47, 77, 78] transfer the point cloud into frontal view images or 2D range images to exploit convolutional neural networks (CNNs) [39]. These approaches enable effective processing of the 3D data and also leverage the tremendous architectural advances in image-based perception. Current approaches [12, 34, 69] process the aggregated residual images of previous scans of 2D range images to include temporal information. The aggregation of scans results in repeating computations of individual scans throughout the whole network, which is computationally disadvantageous. Several approaches [48, 59, 91, 32] utilize a polar bird's-eye-view representation to enhance performance and account for the changing distribution of the points along the radial axis. However, projection-based methods face several limitations, including discretization artifacts and back projection errors, which can harm accuracy. The loss of information is impractical for radar data since detections are sparse.

Voxel-based methods [16, 59, 96] mitigate back projection errors by maintaining 3D information of the point clouds. Since not all voxels are occupied, state-of-the-art methods such as Softgroup [72], SSTNet [42], and HAIS [11] utilize submanifold sparse convolutional networks [25] to extract discriminable features from spatially-sparse input data. Choy et al. [16] and Mersch et al. [46] incorporate temporal information by aggregating scans over time and processing the point cloud throughout the network, which increases the computational burden. Recent methods [40, 59] incorporate multiple representations, such as bird's-eye-view representation [91] with fine-grained voxel features to reduce runtime and improve accuracy. Despite the tremendous progress and encouraging results, voxelization inherently introduces discretization artifacts and information loss, which is especially harmful to the targeted processing of sparse radar point clouds [87].

Point-based methods overcome the lossy encoding of aforementioned paradigms by directly processing the point clouds and keeping the spatial information intact. The pioneering work of Qi et al. [56] utilizes shared multi-layer perceptrons (MLPs) to process the point cloud and aggregate nearby information to capture local structures by max pooling operations. Multiple approaches [55, 63, 75] adopt the hierarchical feature extraction from larger local regions of the successor, PointNet++ [57], to capture strong local structures in point clouds. Schumann et al. [63] aggregate multiple radar point clouds for semantic segmentation and improve the approach by adding a memory module and additional features [65]. In contrast, HARadNet [22] drastically restricts the region of
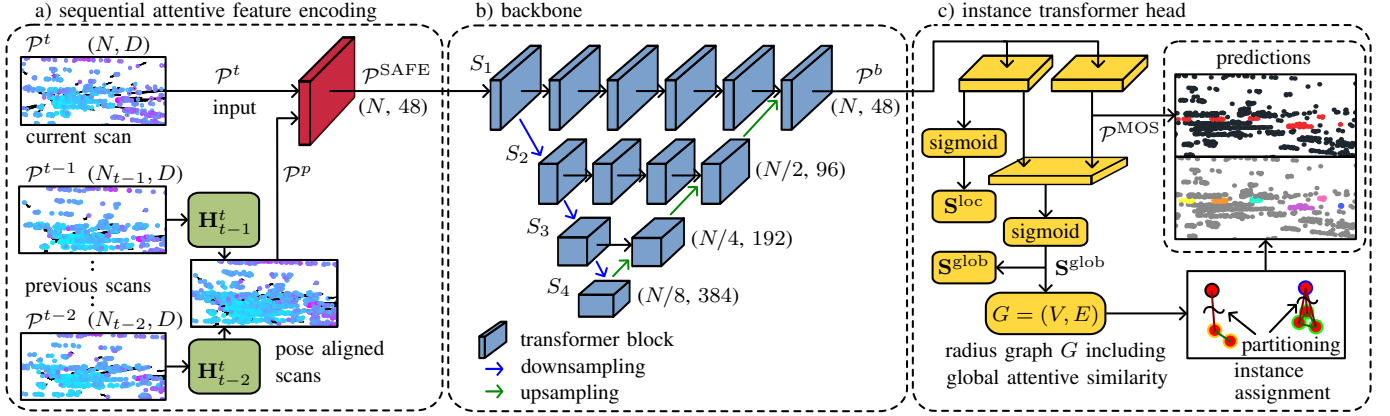
Fig. 2: The detailed design of each module of our Radar Instance Transformer. (a) The sequential attentive feature encoding module combines the pose-aligned point clouds and their corresponding features using the current scan into a featurized point cloud, where the features encode temporal information. (b) This aggregated feature volume is then processed in a multi-pathway transformer backbone, where the transformer layers process the information at different resolutions and aggregate per-point features. (c) The per-point features are then used to estimate the semantics $\mathcal{P}^{\mathrm{MOS}}$ and perform graph partitioning for instance alignment based on the global similarity quantity $\mathbf{S}^{\mathrm{glob}}$. The tuples denote the number of points and feature channels.

interest to the near field to simplify the task and exploits additional properties to improve radar-based instance segmentation. However, for sparse outdoor data, the aggregation of multiple scans is key to including temporal information and extracting valuable features from the surrounding. Nevertheless, the processing of aggregated point clouds through the whole network comes with an increased computational burden.

Another way to exploit stronger connections of the individual points is graph-based methods [38, 74, 94] conducting message passing on the constructed graphs. Furthermore, graph convolutional networks [14, 60] enable the extraction of valuable information within growing receptive fields. However, to reduce the computational burden, the graph representations use super-voxels [27, 66] or over-segmented clusters [60] to combine local areas, which are inappropriate for sparse point clouds where single detection can represent complete instances.

In contrast, kernel-based convolutions [15, 23, 70, 84] process point clouds on a per-point basis and extract features for individual points. State-of-the-art approaches [24, 37] extend these methods with class-dependent post-processing steps to achieve competitive results, which are often inappropriate for moving instances segmentation, including different agents, without having different class labels.

Transformer-based methods exploit the self-attention mechanism [44, 71, 81, 85] and achieve remarkable results within various tasks, from natural language processing [7, 71] to computer vision [10, 13, 21, 86] and point cloud understanding [26, 53, 62, 67, 82, 89, 93].

Zhao et al. [93] utilize vector attention [92] to enable an individual weighting of the feature channels and propose trainable positional encoding [41]. Lai et al. [82] optimize the transformer architecture with a stratified key-sampling strategy to enlarge the effective receptive field, introduce contextual relative positional encoding to keep fine-grained position information throughout the network and use KPConv [70] as the first layer embedding to enhance accuracy. Zeller et al. [87] propose an optimized transformer architecture to

extract features for semantic segmentation of sparse radar point clouds but only consider a single scan.

In contrast to the related work, we propose in this submission a novel architecture inspired by self-attention to efficiently include temporal information and conduct per-point attentive graph-based instance assignments to solve the new task of moving instance segmentation. Our Radar Instance Transformer captures complex structures in noisy and sparse point clouds and extracts valuable features by enriching the information of individual points of the current scan. Furthermore, we are able to include local and global instance information to achieve state-of-the-art performance in moving instance segmentation of radar data.

III. OUR APPROACH TO MOVING INSTANCE SEGMENTATION IN RADAR DATA

Our goal is to achieve precise moving instance segmentation in a sequence of noisy, sparse radar point clouds to enhance scene interpretation of autonomous vehicles. Fig. 2 illustrates the overall architecture of our Radar Instance Transformer, which is a point-based framework and builds upon the self-attention mechanism [71]. We incorporate temporal relationships in a computationally efficient way to enrich the features of the current point cloud, as detailed in Sec. III-A. We furthermore propose a new backbone, processing the full-resolution point clouds throughout the network to extract valuable features and omit information loss in Sec. III-B. We utilize local and global information derived from self-attention to optimize the panoptic task of instance association and semantic segmentation, which we explain in Sec. III-C. Finally, our graph-based instance assignment, detailed in Sec. III-D, incorporates global instance information to enhance accuracy.

A. Sequential Attentive Feature Encoding

The temporal information is essential to enhance scene interpretation in sparse and noisy radar point clouds. Especially for moving instance segmentation, the temporal relationship

helps to identify agents reliably. Additionally, temporal dependencies support the differentiation of moving and static detections, including noise directly definable in the temporal domain due to the often changing appearance in sequences of scans. In contrast to other approaches [34, 46] that pass multiple point clouds through the whole network, we propose the sequential attentive feature encoding (SAFE) module to efficiently enrich the features of a single point cloud with temporal information. Therefore, we only pass the current scan through the whole network and do not increase the number of points that need to be processed by the network, whereas the previous scans are only processed within the SAFE module. The goal is to adaptively combine the temporal information of the previous scans with the features of the current point cloud, hence reducing the computational burden but still keeping important information.

The input to our SAFE module is the current scan $\mathcal{P}^t$ at time t and T previous scans $\mathcal{P}^{t-T}, \ldots, \mathcal{P}^{t-1}$ of a sequence of radar scans, as depicted in Fig. 2. The current scan $\mathcal{P}^t$ includes the point coordinates $\mathbf{P}^t = [\mathbf{p}_1, \ldots, \mathbf{p}_N]^\top \in \mathbb{R}^{N \times 3}$ and point-wise features $\mathbf{X}^t = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$, where $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{x}_i \in \mathbb{R}^D$ for N points. The point-wise features comprise the radar measurements for the individual detections, including velocity, position, and radar cross section, resulting in the input feature dimension D. We aggregate the previous scans into a single scan, which comprises the $N_{t-T} + \cdots + N_{t-1}$ points of all the combined T scans with point coordinates $\mathbf{P}^p$ and point-wise features $\mathbf{X}^p$. We keep the original number of points for each individual point cloud. To align the previous scans with the current scan locally, we assume that the homogeneous relative pose transformation $\mathbf{H}_{t-1}^t \in \mathbb{R}^{4 \times 4}$ between point cloud $\mathcal{P}^t$ and $\mathcal{P}^{t-1}$ is given. The alignment helps to combine the information of the related instances over time and supports the identification of noise within a sequence of scans. The transformation can be derived from GPS, online SLAM algorithms [2, 6, 9, 90], or using wheel encoders and inertial measurement units. In our work, we use the provided vehicle odometry and DGPS information.

Since the data association of individual radar points within unprocessed point clouds is difficult [79, 82], we first process the point-wise features of the current scan $\mathbf{X}^t$ and the features of the previous scans $\mathbf{X}^p$ with a KPConv layer [70] to extract higher dimensional features $\mathbf{x}_i \in \mathbb{R}^{D_1}$. The aggregated features of the previous scans and the features of the current scan are processed by a separate KPConv layer. The resulting features are still point-wise features with dimension D_1. To adaptively combine the features and extract temporal information, we introduce an intra-attention module that is inspired by the Point Transformer layer [93]. We first encode the features of the previous scans $\mathbf{X}^p$ as values $\mathbf{V}$ and keys $\mathbf{K}$ and the features of the current scan $\mathbf{X}^t$ as queries $\mathbf{Q}$ as follows:

$$\mathbf{Q} = \mathbf{X}^t \mathbf{W}_Q, \qquad \mathbf{K} = \mathbf{X}^p \mathbf{W}_K, \qquad \mathbf{V} = \mathbf{X}^p \mathbf{W}_V. \tag{1}$$

The matrices $\mathbf{W}_Q \in \mathbb{R}^{D_1 \times D_2}$, $\mathbf{W}_K \in \mathbb{R}^{D_1 \times D_2}$, and $\mathbf{W}_V \in \mathbb{R}^{D_1 \times D_2}$ are learned linear projections. We increase the feature dimension of the encoding to D_2 to enable fine-grained information aggregation. Since temporal information

is present within the local neighborhood, we restrict the intra-attention of the current scan and the aligned previous scans to local areas. To determine the local areas, we calculate the k-nearest neighbors (kNN) with $k = N^l$ for the points in the current point cloud $\mathcal{P}^t$ within the aligned past point clouds $\mathcal{P}^p$. This results in the combination of the respective detections over time. To extract the related information of the queries, values, and keys within the local areas, we utilize the sample and grouping algorithm [57] resulting in $\mathbf{Q}^*, \mathbf{K}^*$, and $\mathbf{V}^* \in \mathbb{R}^{N \times N^l \times D_2}$. The queries comprise the repeated entries of the current point cloud, and the keys and values include the encoded features of the past point clouds, which belong to the corresponding local neighbors of the N points of the current point cloud. To further include the relative positional encoding [93], we determine the relative positions within the local areas of the two point clouds where $\mathbf{p}_i \in \mathcal{P}^t$ and $\mathbf{p}_j \in \mathcal{P}^p$ resulting in:

$$\mathbf{r}_{i,j} = \mathbf{p}_i - \mathbf{p}_j. \tag{2}$$

Based on the relative positions $\mathbf{r}_{i,j}$, we compute the positional encoding $\mathbf{R} \in \mathbb{R}^{N \times N^l \times D_2}$ by an MLP, including two linear layers with weight matrices $\mathbf{W}_{R_1} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{W}_{R_2} \in \mathbb{R}^{3 \times D_2}$, batch normalization [33], and rectified linear unit (ReLU) activation function [49]. To calculate the attention weights $\mathbf{A} \in \mathbb{R}^{N \times N^l \times D_2}$ for the individual points i, we adopt vector attention [92] and subtract the encoded keys of the past point clouds from the encoded queries of the current point cloud where $\mathbf{Q}_i^*, \mathbf{K}_i^* \in \mathbb{R}^{N^l \times D_2}$. Similarly, we add the relative positional encoding $\mathbf{R}_i$ to include fine-grained position information in the weighting. Hence, the attention weights connect the detections in the current scan with the detections in the aligned previous scans, which we determine as follows:

$$\mathbf{A}_i = \mathrm{softmax}((\mathbf{Q}_i^* - \mathbf{K}_i^*) + \mathbf{R}_i), \tag{3}$$

where we apply the softmax function to each feature of the points within the local area individually. We process the resulting attention weights by a multi-layer perceptron with two fully connected layers, where each layer is followed by batch normalization [33]. To aggregate the weighted features $\mathbf{X}^{\mathrm{temp}}$, which comprise the temporal information of previous scans encoded as values, we calculate the sum of the element-wise multiplication, indicated by $\odot$, as:

$$\mathbf{X}_i^{\mathrm{temp}} = \sum_{j=1}^{N^l} \mathbf{A}_{i,j} \odot (\mathbf{V}_{i,j}^* + \mathbf{R}_{i,j}). \tag{4}$$

Additionally, we include the positional encoding [93] because it provides valuable information about the local areas and their appearance over time. Static objects remain in the same location, whereas moving objects change position within a sequence of scans, which both can be represented within the positional encoding. To combine the information within the current scan, we concatenate the temporal features $\mathbf{X}^{\mathrm{temp}} \in \mathbb{R}^{N \times D_2}$ and the features of the current scan $\mathbf{X}^t \in \mathbb{R}^{N \times D_1}$ after the KPConv layer resulting in the temporal enriched features $\mathbf{X}^{\mathrm{SAFE}} \in \mathbb{R}^{N \times (D_1 + D_2)}$ of our SAFE module. Besides the features, the output point cloud $\mathcal{P}^{\mathrm{SAFE}}$ includes the point coordinates $\mathbf{P}^{\mathrm{SAFE}}$, where $\mathbf{P}^{\mathrm{SAFE}} = \mathbf{P}^t$. The point coordinates

are not processed to include fine-grained position information within the consecutive network. The SAFE module is model agnostic and can be applied to different backbones.

B. Backbone

The motivation behind our backbone design is to keep the full resolution, i.e., the original number of points throughout the network, and thus avoid information loss and extract valuable features from sparse radar scans, as illustrated in Fig. 2. Contrary to the widely-used U-Net [57, 93] with an encoder-decoder architecture including skip connection, we process the full-resolution point cloud in parallel to the downsampled feature maps and enrich the information in the original point cloud with high-level features. In contrast to 3D LiDAR scans, the sparsity of radar data enables processing full-resolution point clouds but also requires keeping as much information as possible to enhance performance. To handle the changing number of points in radar scans, we adopt the advanced batching algorithm [93] to always keep the original number of points. Compared to the first parallel network for image analysis, the HRNet [73], we reduce the processing of the high-level features to a minimum and do not keep the parallel tracks throughout the network. Considering that the processing of these features is computationally expensive, HRNet reduces the input size to keep the parallel feature extraction, which is in contrast to our approach that preserves the information of the point cloud.

The input point cloud $\mathcal{P}^{\text{SAFE}}$ of our backbone includes the temporally enriched features $\mathbf{X}^{\text{SAFE}}$ and the position information $\mathbf{P}^t$ of the current scan. The dimension of the features for the stages S_L with $L = 1, \ldots, 4$ are D_{S_L}, where $D_{S_1} = (D_1 + D_2)$, $D_{S_2} = 2D_{S_1}$, $D_{S_3} = 2D_{S_2}$, and $D_{S_4} = 2D_{S_3}$. The individual building blocks of our backbone are the commonly used transformer blocks as well as up- and downsampling layers. The stages S_1, S_2, S_3, and S_4 comprise 6, 4, 2, and 1 transformer blocks, respectively. The transformer block is a residual block that follows the design of the Stratified Transformer block [82], where we replace the window attention with the Point Transformer layer [93] since it is computationally more efficient. Within the block, we first process the input features by layer normalization [3] before feeding the features and position information into the transformer layer. The resulting output is processed by two fully connected layers, including the Gaussian error linear unit [29] activation function and layer normalization. We add the features of the residual path to the updated features to determine the output features.

The transformer layer adopts the architecture of the Point Transformer layer [93] and follows the implementation of our SAFE module. We utilize the inter-attention operation and encode the features $\mathbf{X}^{\text{SAFE}}$ as values, keys, and queries following Eq. (1). In contrast to our SAFE module, we do not increase the dimension of the encodings by the learned linear projections, hence we set the dimension D_1 and D_2 of $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ to D_{S_L}. We determine the local areas within the current point cloud based on the kNN algorithm with $k = N^l$. To align the queries and keys within the local areas,

we utilize the sample and grouping algorithm, as mentioned in Sec. III-A. We calculate the positional encoding by an MLP, including two linear layers, batch normalization [33], and ReLU activation function [49]. We adjust the output dimension of $\mathbf{W}_{R_2}$ to D_{S_L}. The relative position changes to $\mathbf{r}_{i,j} = \mathbf{p}_i - \mathbf{p}_j$ where $\mathbf{p}_j, \mathbf{p}_i \in \mathcal{P}^t$, to include detailed position information of the current scan. We calculate the final output features of the transformer layer based on Eq. (4). The point coordinates are not further processed to keep the information intact and improve the features with fine-grained position information within consecutive blocks.

For the downsampling, we utilize the commonly used farthest point sampling with k nearest neighbor (kNN) search with $k = N^d$ and max pooling [57]. The top level does not include downsampling since we want to keep the original number of points. From stage S_L to stage S_{L+1}, we sample $N_{S_{L+1}}$ points from N_{S_L} points where $N_{S_{L+1}} = N_{S_L}/2$. We reduce the cardinality of the point cloud by a factor of 2 instead of 4 [57, 93] to keep more information since radar point clouds are sparse, resulting in $[N, N/2, N/4, N/8]$ points for the four stages, where N is the number of points in the current scan. Based on the local areas, we aggregate the features and process them by a linear layer with weight matrix $\mathbf{W}_{S_L}^{\text{down}} \in \mathbb{R}^{D_{S_L} \times D_{S_{L+1}}}$ where $D_{S_{L+1}} = 2 \cdot D_{S_L}$ and layer normalization. Afterward, we apply max pooling to derive the high-level features of the downsampled point cloud and pass the features and the point coordinates of the sampled points to the consecutive stage.

For the upsampling, we adopt the trilinear interpolation based on an inverse distance weighted average [57]. We determine $k = 3$ neighbors of the points of stage S_{L-1} within the downsampled points of stage S_L based on the kNN algorithm. We process the features of stage S_L by a linear layer with $\mathbf{W}_{S_L}^{\text{up}} \in \mathbb{R}^{D_{S_L} \times D_{S_{L-1}}}$ where $D_{S_{L-1}} = D_{S_L}/2$ and layer normalization to adjust the dimensions of the features of the two stages. We interpolate the resulting features and add them to the features of the current stage S_{L-1}, which we additionally process by a linear layer with $\mathbf{W}_{S_{L-1}}^{\text{up}} \in \mathbb{R}^{D_{S_{L-1}} \times D_{S_{L-1}}}$ and layer normalization. The upsampled features are processed within the succeeding stages to integrate the information iteratively. Subsequently, we add high-level features to the full-resolution point cloud of stage S_1 to derive the final output features of our backbone $\mathbf{X}^b \in \mathbb{R}^{N \times D_{S_1}}$. The advanced network architecture enables the extraction of valuable features for sparse radar point clouds to enhance performance.

C. Moving Instance Transformer Head

Our moving instance transformer head combines moving object segmentation with instance segmentation to derive the final panoptic output, as detailed in Fig. 2. For moving instance segmentation, currently moving agents, such as cars, bikes, and trucks, belong to the same moving class without further differentiation. For many tasks, the information of being a moving object in combination with the velocities is often sufficient, simplifying the labeling efforts. However, state-of-the-art approaches for panoptic segmentation and instance segmentation rely on specific semantic labels to enhance

accuracy [11, 24, 40, 72]. To overcome current limitations and further incorporate instance information, we propose a new attention-based module to reliably identify moving instances in sparse and noisy radar point clouds.

First, we predict the per-point moving object segmentation (MOS) class $\mathcal{P}^{\mathrm{MOS}} = \{p_1^{\mathrm{MOS}}, \ldots, p_N^{\mathrm{MOS}}\}$, where $p_i^{\mathrm{MOS}} \in \{\mathrm{static}, \mathrm{moving}\}$ using a multi-layer perceptron. To identify points belonging to the same instances, our idea is to deduce an attentive similarity quantity $\mathbf{S}_{i,j} \in \mathbb{R}$ based on the self-attention mechanism. Therefore, we process the output features of the backbone $\mathbf{X}^b$ by two linear layers to encode the features as keys $\mathbf{K}^b$ and queries $\mathbf{Q}^b$, with $\mathbf{W}_Q^b \in \mathbb{R}^{D_{S_1} \times D_{S_1}}$ and $\mathbf{W}_K^b \in \mathbb{R}^{D_{S_1} \times D_{S_1}}$ following Eq. (1). Since global attention is expensive to compute within the complete point cloud, we restrict the computation to local areas, which we determine by kNN with $k = N^a$. We aggregate the local neighborhoods by sample and grouping, resulting in $\mathbf{Q}^{b*}, \mathbf{K}^{b*} \in \mathbb{R}^{N \times N^a \times S_1}$, as explained in Sec. III-A. We calculate the relative positions within the resulting local areas following Eq. (2), where $\mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}^t$. We reduce the dimensionality of the relative positional encoding $\mathbf{R}^b$ by a linear layer with weight matrix $\mathbf{W}_R^b \in \mathbb{R}^{3 \times 1}$ and ReLU activation function [49] to include the position information. To derive a scalar value for the local similarity quantity, we calculate the dot-product attention $\mathbf{A}^{\mathrm{loc}} \in \mathbb{R}^{N \times N^a}$ for the keys and queries, resulting in:

$$\mathbf{A}_{i,j}^{\mathrm{loc}} = \mathbf{Q}_{i,j}^{b*} \mathbf{K}_{i,j}^{b* \top} + \mathbf{R}_{i,j}^b. \tag{5}$$

The goal is to have local similarity quantities close to 1 for points within the local area that belong to the same instance and values close to 0 for different instances. Therefore, we replace the softmax function with the sigmoid function to determine the final local similarity quantity $\mathbf{S}^{\mathrm{loc}}$ as:

$$\mathbf{S}_{i,j}^{\mathrm{loc}} = \mathrm{sigmoid}(\mathbf{A}_{i,j}^{\mathrm{loc}}) = \frac{1}{1 + \exp(-\mathbf{A}_{i,j}^{\mathrm{loc}})}. \tag{6}$$

The resulting local similarity matrix $\mathbf{S}^{\mathrm{loc}} \in \mathbb{R}^{N \times N^a}$ includes a similarity quantity for all N points of the current point cloud. The ground truth of the local similarity matrix can be directly derived from the indices of the kNN algorithm and the ground truth instance IDs. Since the focus is to extract valuable information for moving instances, we set the similarity quantity for points within the local area that both belong to the static class to 0, which we elaborate in more detail in Sec. V-E.

The restriction to local areas induces the problem that often not all points that belong to one instance have an attentive similarity quantity. To overcome this limitation, we introduce the global attentive similarity quantity $\mathbf{S}^{\mathrm{glob}}$ for moving detections. Hence, the second part of our instance transformer head combines the prediction of the moving object segmentation and the approach of our local similarity quantity. Since computing the global similarity, including all points, comes with a large computational burden, we first select the N^{mov} points that we predict as moving. In contrast to Cheng et al. [13], we select all moving predictions, not just those predicted within the same mask. We already encoded the features as keys $\mathbf{K}^b$ and queries $\mathbf{Q}^b$ for the local attentive similarity such that no additional encoding is required. We

sample and group the global moving predictions for the queries and keys, resulting in $\mathbf{Q}^{b**}, \mathbf{K}^{b**} \in \mathbb{R}^{N^{\mathrm{mov}} \times N^{\mathrm{mov}} \times S_1}$. We use the dot-product to derive the global attention weights $\mathbf{A}^{\mathrm{glob}} \in \mathbb{R}^{N^{\mathrm{mov}} \times N^{\mathrm{mov}}}$ as:

$$\mathbf{A}_{i,j}^{\mathrm{glob}} = \mathbf{Q}_{i,j}^{b**} \mathbf{K}_{i,j}^{b** \top}. \tag{7}$$

We remove the relative positional encoding because the values largely differ in magnitude and, at least in our experience detailed in Sec. V-E, do not provide important information to differentiate instances in the global context. We follow Eq. (6) to calculate the final global similarity quantity $\mathbf{S}^{\mathrm{glob}} \in \mathbb{R}^{N^{\mathrm{mov}} \times N^{\mathrm{mov}}}$. We construct the ground truth matrix for the global similarity matrix based on the semantic predictions of the moving instance detections. The attentive similarity modules are differentiable and can be learned in an end-to-end manner. We emphasize that our attention-based similarity quantity encodes essential information for moving instance association and moving object segmentation in noisy radar point clouds. Further, the local attention mechanism includes the noise in the ground truth and directly penalizes the association of noise with an instance, which is beneficial for radar data processing. Incorporating global information supports the extraction of discriminative features, especially for sparse point cloud processing. The computation of the local and global similarity quantities is model agnostic and can be combined with different backbones.

D. Graph-based Instance Assignment

The final part of our instance transformer head is the graph-based instance assignment module. It utilizes the predictions $\mathcal{P}^{\mathrm{MOS}}$ and the global similarity matrix $\mathbf{S}^{\mathrm{glob}}$, which includes a similarity quantity for each pair of moving predictions to derive the final instance IDs. The key idea is to overcome the limitations of the direct assignment since the global similarity matrix may include multiple instance assignments and different similarity quantities for different instances, which are difficult to solve. Furthermore, we want to remove the class dependencies mentioned in Sec. III-C to derive the final instances.

We first construct a radius graph $G = (V, E)$ based on the prediction of moving detections and the position information to take into account that sparse and noisy radar point clouds differ in density. The moving points represent the vertices $V = \{1, \ldots, N^{\mathrm{mov}}\}$ of the graph. We derive the set of m edges E by connecting each moving point with the neighboring points, which lie within the spatial radius r. The resulting graph can be represented by the square adjacency matrix $\mathbf{A}^{\mathrm{adj}} \in \mathbb{R}^{N^{\mathrm{mov}} \times N^{\mathrm{mov}}}$, where $\mathbf{A}_{i,j}^{\mathrm{adj}} = 1$, if $||p_i - p_j|| \leq r$ and otherwise $\mathbf{A}_{i,j}^{\mathrm{adj}} = 0$. To derive the instance IDs, we partition the graph into $V_1, \ldots, V_{k'}$ disjoint subgraphs, where k' is not given in advance because the number of instances within a scan is unknown. Based on the constructed graph, our idea is that individual instances have highly interconnected nodes and points belonging to different instances are sparsely connected. To exploit this property, we partition the graph by maximizing the modularity via a spectral approach [50]. The modularity depends on the number of edges falling within clusters minus

the approximated expected number of random edges in an equivalent network. For the case of two clusters and $s_i = 1$, if vertex i belongs to the first cluster and $s_i = -1$ if it belongs to the second one. The modularity is:

$$Q = \frac{1}{4m} \sum_{i,j} \left(\mathbf{A}_{i,j}^{\mathrm{adj}} - \frac{k_i k_j}{2m} \right) s_i s_j, \qquad (8)$$

where k_i and k_j are the degrees, the number of edges incident to a vertex, of the corresponding vertices. The adjacency matrix $\mathbf{A}_{i,j}^{\mathrm{adj}}$ contains information about the actual number of edges, and the factor $\frac{k_i k_j}{2m}$ is the approximated expected number of edges between vertices i and j if the edges are placed randomly. The modularity can be further expressed by the modularity matrix $B_{i,j} = \mathbf{A}_{i,j}^{\mathrm{adj}} - \frac{k_i k_j}{2m}$ as:

$$Q = \frac{1}{4m} \mathbf{s}^\top \mathbf{B} \mathbf{s}, \qquad (9)$$

where $\mathbf{s}$ is the vector of the elements s_i. To correctly divide the network into more than two clusters, we construct a modularity matrix $\mathbf{B}^{\mathrm{sub}}$ for each subgraph resulting in:

$$\mathbf{B}_{i,j}^{\mathrm{sub}} = \mathbf{A}_{i,j}^{\mathrm{adj}} - \frac{k_i k_j}{2m} - \delta_{i,j} \left[k_i^{\mathrm{sub}} - k_i \frac{d^{\mathrm{sub}}}{2m} \right], \qquad (10)$$

where k_i^{sub} is the degree of the node i within the subgraph, d^{sub} is the sum of all degrees k_i of the nodes in the specific subgraph, and $\delta_{i,j}$ is the Kronecker delta, which is 1 if both nodes belong to the same subgraph and zero otherwise. We calculate the subgraph modularity Q^{sub} following Eq. (9) to determine the additional contribution to the total modularity Q.

The modularity matrix $\mathbf{B}$ always depends on the adjacency matrix $\mathbf{A}^{\mathrm{adj}}$. Since the plain radius graph leads to interconnections of nearby instances, represented in $\mathbf{A}^{\mathrm{adj}}$, we optimize the cluster assignment by including additional information about the vertices. Our global attentive similarity matrix $\mathbf{S}^{\mathrm{glob}}$ directly provides this valuable information since the similarity quantities differentiate between instances. Hence, we calculate the element-wise product of both matrices as:

$$\mathbf{A}_{\mathrm{attn}}^{\mathrm{adj}} = \mathbf{S}^{\mathrm{glob}} \odot \mathbf{A}^{\mathrm{adj}}, \qquad (11)$$

to resolve the issue and derive the final attention-based adjacency matrix $\mathbf{A}_{\mathrm{attn}}^{\mathrm{adj}}$ for the instance assignment. The modularity, which now incorporates the additional instance information, is maximized by the vertex moving method to reach the best possible modularity. The optimization stops if any proposed split for all the subgraphs has a negative or no effect on the modularity. The final partitions represent our instance IDs. The partitioning is class independent and incorporates attention-based instance information.

IV. IMPLEMENTATION DETAILS

We implemented our Radar Instance Transformer in PyTorch [54] and train our network with one Nvidia A100 GPU and a batch size b of 64 over 100 epochs. We utilize the AdamW [45] optimizer with an initial learning rate of 0.001 and drop the learning rate by a factor of 10 after 60 and 80 epochs to train our model. We use the binary cross-entropy loss to learn the local $\mathbf{S}^{\mathrm{loc}}$ and global $\mathbf{S}^{\mathrm{glob}}$ similarity matrix for instance association. For the semantic output, we utilize the focal Tversky loss [1].

The individual scans, which are the input to our network, are sparse radar point clouds with N points and feature dimension D. The RadarScenes [64] dataset comprises two spatial coordinates and the Doppler velocity, whereas the View-of-Delft dataset [52] also includes the elevation information. The radar sensors are often defined as 2+1D and 3+1D radars, respectively. Each sequence consists of $T + 1$ scans, and the batch size b is the number of input sequences. For RadarScenes, we extend the coordinates x_i^C, y_i^C with the coordinate $z_i^C = 0$ to apply the pose transformation mentioned in Sec. III-A. In addition to the position information and the Doppler velocity v_i, the radar sensors provide the radar cross section σ_i, resulting in input vector $\mathbf{x}_i = (x_i^C, y_i^C, z_i^C, \sigma_i, v_i)$.

We first increase the per-point features to $D_1 = 16$ before we apply the intra-attention. We increase the dimension in our SAFE module and set $D_2 = 32$ resulting in an output feature dimension of 48, which is kept for the full-resolution stage $D_{S_1} = 48$ and gradually increased to $D_{S_2} = 96$, $D_{S_3} = 192$, and $D_{S_4} = 384$ during downsampling. We set $N^l = N^d = N^a = 12$, where N^l represents the local area in the transformer layers, N^d the local neighborhood for downsampling, and N^a the local area for the attentive local similarity quantity. The input sequence consists of the current point cloud and $T = 2$ previous scans, which we further elaborate in Sec. V-D. We pad missing previous scans as zeros to always include $T + 1$ scans as input. The padded scans comprise a set of $1,024$ points where we set the features to zero. The padding is adopted until $T + 1$ scans are available. To reduce overfitting during training, we apply data augmentation [84], including Gaussian jittering (variance of 0.01), scaling [$\times 0.95$, $\times 1.05$], shifting ($\pm 0.1\,\mathrm{m}$), and random flipping of the point cloud.

V. EXPERIMENTAL EVALUATION

The main focus of this work is to achieve reliable moving instance segmentation in sparse and noisy radar point clouds. We present our experiments to show the capabilities of our method and to support our key claims, including that our approach achieves state-of-the-art performance in moving instance segmentation without processing multiple scans throughout the whole network. Our sequential attentive feature encoding module improves accuracy by efficiently incorporating temporal information. In addition, the attentive instance transformer head enhances the overall performance of moving instance segmentation by including local and global instance knowledge. Our attention-based graph clustering enhances class-independent instance assignments. The optimized backbone architecture extracts discriminable features from sparse radar point clouds by enriching individual detection with high-level features.

A. Experimental Setup

We perform our evaluation on RadarScenes [64], the only large-scale open-source high-resolution radar dataset [95] including precise point-wise annotations for moving detections

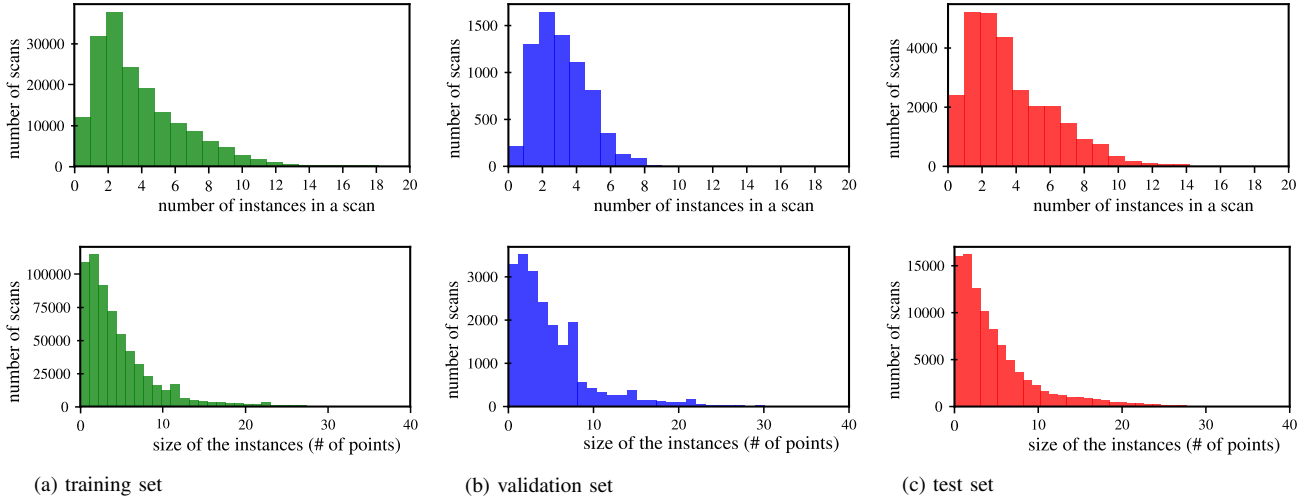(a) training set (b) validation set (c) test set

Fig. 3: Statistics of the moving instance segmentation benchmark based on the RadarScenes dataset. The statistics are conducted individually for the training (a), the validation (b), and the test set (c). The top row illustrates the number of instances within a scan, and the bottom row shows the size of the instances, which correspond to the number of points.

RadarScenes [64]	training set	validation set	test set
number of scans	175,918	7,062	27,849
number of moving instances	630,851	21,143	95,392
avg. number of points per scan	569	559	534

TABLE I: The number of scans, the number of moving instances, and the average number of points per scan in the RadarScenes training, validation, and test set.

for varying scenarios and different weather conditions. Additionally, we evaluate our method on the medium-sized View-of-Delft dataset [52] to illustrate the generalization capabilities of our method. For RadarScenes, we follow Zeller et al. [87] for the training, validation, and test split of the 158 annotated sequences. We utilize the validation set to perform the ablation studies. Furthermore, we assemble the annotations to establish the new task of moving instance segmentation, which we introduce in Sec. V-B. For the View-of-Delft dataset, we keep the original training and validation split. We transfer the bounding box labels into point-wise annotations, including the differentiation between moving and static instances. We utilize the panoptic quality (PQ) [5] to evaluate the combined task of semantic segmentation and instance segmentation. Additionally, we report the segmentation quality (SQ), the recognition quality (RQ), and the intersection over union (IoU). To derive a detailed evaluation, we further differentiate between the static (stat) and moving (mov) classes.

B. Moving Instance Segmentation Benchmark

The RadarScenes [64] dataset consists of 158 annotated sequences split into 130 sequences for training and 28 sequences for testing. We utilize six of the 28 sequences as a validation set (sequences: 6, 42, 58, 85, 99, 122) as previously done [87].

The RadarScenes dataset provides individual point clouds for the four radar sensors. The measurements are from the near-range mode of the 77 GHz automotive radar sensors, which cover detections in a range of up to 100 m. Two sensors

are mounted at $\pm 85°$ and two sensors at $\pm 25°$ with respect to the driving direction. Since the directions in which the sensors point are different, we combine the individual point clouds into one scan based on the relative transformations to obtain information about the surroundings of the vehicle following Zeller et al. [87]. However, at the beginning of the sequence, we recognized that the recordings were asynchronous. To account for that, we only combine the intermediate point clouds of the individual sensors until the next measurement of an already included sensor is available. The reason is that if we include multiple point clouds of one sensor, the pose extraction is more erroneous because each point cloud has corresponding pose information. Since the resulting point clouds are recorded within a short time frame, we do not compensate for additional movement.

We extract the time and the pose information based on the first measurement included within the combined scan in the common coordinate system. The scenes are represented as a sequence of scans where each scan includes up to four radar sensors. The semantic annotations for the moving agents comprise the car, large vehicle, two-wheeler, pedestrian, and pedestrian group class. We transfer the track IDs into instance labels for the individual scans.

We summarize the number of scans for the individual dataset splits in Tab. I. With more than 200,000 scans in total, the large-scale RadarScenes dataset enables data-driven moving instance segmentation. To illustrate the diversity of the dataset, we show in Fig. 3 histograms of the number of instances within a scan and the size of the instances, which is equal to the number of points per instance. The number of instances within a scan displays the diversity of scenes with changing numbers of agents. Furthermore, the small size of the instances illustrates the difficulty of reliably detecting moving agents within sparse radar point clouds. Moreover, 87% of the points with an absolute ego-motion compensated Doppler velocity larger than 0.1 m/s are noise and hence belong to

model	PQ	mIoU	SQ	RQ	PQ^{mov}	IoU^{mov}	SQ^{mov}	RQ^{mov}	PQ^{stat}	IoU^{stat}	SQ^{stat}	RQ^{stat}
Threshold $|v_i| > v_t$ + HDBSCAN [8]	59.7	65.2	88.3	64.6	23.7	35.1	80.9	29.3	95.7	95.2	95.7	100.0
Threshold $|v_i| > v_t$ + mean shift [17]	57.2	65.2	90.4	61.0	18.7	35.1	85.1	22.0	95.7	95.2	95.7	100.0
4DMOS [46] + HDBSCAN [8]	71.3	86.0	93.1	75.0	43.7	73.1	87.3	50.1	98.9	99.0	98.9	100.0
4DMOS [46] + mean shift [17]	79.0	86.0	95.9	81.9	59.1	73.1	92.7	63.7	98.9	99.0	98.9	100.0
Point Voxel Transformer [89] + HDBSCAN [8]	70.3	84.7	91.8	74.6	41.9	70.7	84.9	49.3	98.7	98.8	98.7	100.0
Point Voxel Transformer [89] + mean shift [17]	76.3	84.7	94.6	79.4	53.9	70.7	90.7	59.9	98.7	98.8	98.7	100.0
Stratified Transformer [82] + HDBSCAN [8]	71.2	87.3	92.1	75.5	43.5	75.2	85.2	51.1	98.9	99.0	98.9	100.0
Stratified Transformer [82] + mean shift [17]	78.4	87.3	94.8	81.7	57.9	75.2	90.8	63.7	98.9	99.0	98.9	100.0
Gaussian Radar Transformer [87] + HDBSCAN [8]	73.2	89.1	92.9	77.3	47.3	79.1	86.8	54.5	99.0	99.1	99.0	100.0
Gaussian Radar Transformer [87] + mean shift [17]	81.1	89.1	95.9	84.1	63.3	79.1	92.7	68.2	99.0	99.1	99.0	100.0
DS-Net [30]	62.0	82.2	94.2	64.4	25.9	66.1	90.3	28.7	98.1	98.2	98.1	100.0
Mask3D [62]	80.4	84.3	96.4	83.0	62.0	69.8	94.0	66.0	98.8	98.9	98.8	100.0
Ours	**86.5**	**92.6**	**96.9**	**89.0**	**73.6**	**85.7**	**94.4**	**78.0**	**99.4**	**99.4**	**99.4**	**100.0**

TABLE II: Moving instance segmentation results on the RadarScenes test set in terms of PQ, SQ, RQ, and IoU scores.
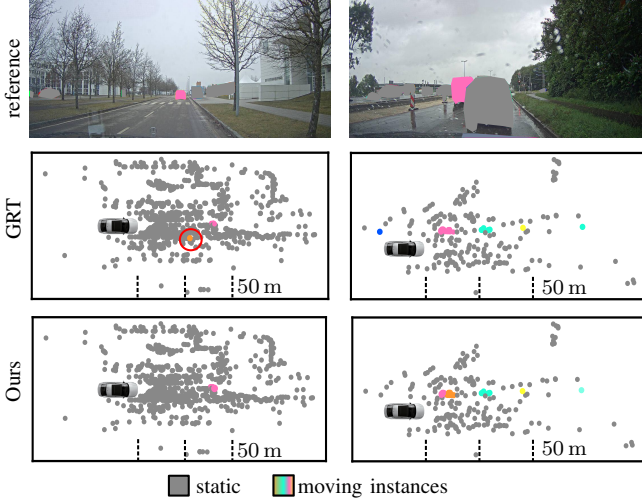

Fig. 4: Qualitative results of our Radar Instance Transformer and the Gaussian Radar Transformer (GRT) [87] on the test set of RadarScenes. The camera images are anonymized and just shown for reference. The colors in the image do not correspond to the predicted instances. The left column is from sequence 14 (fog), and the right is from sequence 93 (rain). In the images of the predictions, each color represents a different instance of moving objects (static is grey). The colors in the images correspond if the object is visible.

Fig. 5: Qualitative results of our Radar Instance Transformer on the validation set of View-of-Delft. The camera images are shown for reference. The ground truth (gt) instance labels and the corresponding instance predictions are projected into the reference image. The ground truth and the predictions are cropped for better visibility.

the static class. The sparsity and noisiness demonstrate that reliable moving instance segmentation in radar data is more challenging compared to typically used 3D LiDAR data. To enable further research, the benchmark [88] is available at https://doi.org/10.5281/zenodo.10203864.

C. Moving Instance Segmentation

The first experiment evaluates the performance of our approach, and its outcome supports the claim that we achieve state-of-the-art results for moving instance segmentation in sparse and noisy radar point clouds. We compare our Radar Instance Transformer to the recent and high-performing networks with a strong performance on different benchmarks, including moving object [46], semantic [82, 87, 89], instance [62], and panoptic segmentation [30]. We added the two approaches [30, 62] based on the fact that the clustering has to be class-agnostic because all instances belong to the moving class. Additionally, the methods do not include range image representations since RadarScenes does only provide x and y coordinates. To derive panoptic labels, we extend Mask3D [62] with state-of-the-art post-processing for mask

predictions [13]. We extend the semantic and moving object segmentation approaches [46, 82, 87, 89] with commonly used clustering algorithms, namely HDBSCAN [8] and mean shift [17], to group points into instances. Furthermore, we add a threshold-based baseline that first identifies moving detections based on the ego-motion compensated Doppler velocities where threshold $v_t = 0.92\,m/s$ and utilizes the clustering algorithms to group the detected point into moving instances. To reliably detect the most vulnerable road users, the pedestrians, we have to identify instances that comprise just a single detection. Therefore, we optimize the hyperparameters and set the minimum cluster size for HDBSCAN to 1. For mean shift, we optimize the bandwidth b^{ms} based on the overall performance on the validation set, resulting in 3.5, 4.5, 5.0, and 4.5 for 4DMOS [46], Stratified Transformer [82], Point Voxel Transformer [89] and Gaussian Radar Transformer [87], respectively. To reliably detect larger objects, we additionally implement an offset prediction head [31] to support the clustering and enhance accuracy for point-based methods. The offset prediction head [31] combines two fully connected layers, batch normalization [33], and a rectified linear unit (ReLU) [49] to regress offset to the instance centers. We concatenate the features of the respective backbone

model	PQ$^{\mathrm{mov}}$	SQ$^{\mathrm{mov}}$	RQ$^{\mathrm{mov}}$	IoU$^{\mathrm{mov}}$
Stratified Transformer [82]	22.8	78.1	29.3	55.6
Gaussian Radar Transformer [87]	26.0	**79.4**	32.7	62.3
Ours	**26.9**	79.1	**34.0**	**63.3**

TABLE III: Moving instance segmentation results on the View-of-Delft [52] validation set in terms of PQ$^{\mathrm{mov}}$, SQ$^{\mathrm{mov}}$, RQ$^{\mathrm{mov}}$, and IoU$^{\mathrm{mov}}$ scores.

$\mathbf{X}^b$ and the coordinates $\mathbf{P}^b$ to include fine-grained position information. The offsets $\mathbf{O} \in \mathbb{R}^{N \times 3}$ point from the point coordinates $\mathbf{P} \in \mathbb{R}^{N \times 3}$ to the instance centers $C \in \mathbb{R}^{N \times 3}$. We derive the center based on the ground truth instance IDs and point coordinates. The loss function for offset regression can be expressed as follows:

$$L^{\mathrm{off}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{o}_i - (\mathbf{c}_i - \mathbf{p}_i)\|_1, \qquad (12)$$

where N is the number of points in the current point cloud.

Our Radar Instance Transformer outperforms the existing methods, especially in terms of PQ and IoU, as displayed in Tab. II. The threshold-based baseline further illustrates the difficulties of the task and underlines the necessity of advanced approaches. The learning-based approaches improve IoU$^{\mathrm{mov}}$ by more than 30 absolute percentage points. DS-Net utilizes an optimized instance clustering, but point-based approaches [82, 89] clearly exceed the voxel-based method, which underlines that minimizing discretization artifacts enhances performance. Point Voxel Transformer includes a point-based branch, which helps to reduce the negative effect. Contrary, 4DMOS [46] enhances performance by aggregating scans and prediction smoothing over time. However, the instance assignment is difficult. We assume that the problems result from the fact that all instances belong to the same class, which makes offset prediction and clustering more complex. Nevertheless, algorithms that do not depend on semantic labels are more generally applicable and potentially cover the long-tail classes. The dedicated mask predictions of Mask3D [62] overcome this limitation and improve PQ. However, our Radar Instance Transformer enhances PQ$^{\mathrm{mov}}$ by 10 absolute percentage points compared to Mask3D, which is a strong improvement. The Stratified Transformer performs well without temporal information. Particularly the mean shift algorithm performs well since the restriction of a cluster of one for HDBSCAN makes it difficult to cluster larger objects reliably. However, even in the semantic domain, our Radar Instance Transformer enhances IoU$^{\mathrm{mov}}$ by more than 9 absolute percentage points, which again illustrates the superiority of our method. Moreover, we even surpass the radar-specific method, the Gaussian Radar Transformer [87], by more than 5 absolute percentage points in terms of IoU$^{\mathrm{mov}}$.

To further illustrate the generalization capability of our model, we evaluate the three best-performing methods in terms of IoU$^{\mathrm{mov}}$ on the View-of-Delft validation set. Tab. III shows the superior performance of our method compared to the Gaussian Radar Transformer and Stratified Transformer. To enable a fair comparison, we utilize the IoU since both baselines are developed for semantic segmentation. The Stratified

	PC	PQ$^{\mathrm{mov}}$	SQ$^{\mathrm{mov}}$	RQ$^{\mathrm{mov}}$	IoU$^{\mathrm{mov}}$
$T = 0$	✓	73.8	94.7	77.9	82.2
$T = 1$	✓	74.0	94.8	78.0	82.5
$T = 3$	✓	76.4	95.0	80.4	83.6
$T = 4$	✓	77.3	95.2	81.2	**84.8**
$T = 5$	✓	**77.7**	95.2	**81.6**	84.5
Ours without PC		76.1	**95.3**	79.8	83.3
Ours ($T = 2$)	✓	76.8	95.1	80.8	84.4

TABLE IV: Influence of the number of previous input point clouds and pose compensation (PC) on the RadarScenes validation set.

Transformer achieves an IoU$^{\mathrm{stat}}$ of 98.1, whereas the Gaussian Radar Transformer and our method achieve an IoU$^{\mathrm{stat}}$ of 98.4. Nevertheless, our approach enhances the IoU$^{\mathrm{mov}}$ by 1 absolute percentage point compared to the best-performing radar-specific method and by more than 7 absolute percentage compared to Stratified Transformer. We observe that the relative transformations to align the point clouds are more precise in the RadarScenes dataset. We assume that this might be one reason why the improvements are more limited, as detailed in Sec. V-D. Overall, the approaches perform well in both moving instance segmentation benchmarks, which is a good starting point for future research. One possibility is to predict the semantic labels in a cluster-then-classify approach, which was out of the scope of this paper.

Fig. 4 shows some qualitative results on the RadarScenes test set of our approach and the Gaussian Radar Transformer. Our approach correctly identifies the distant moving instance in scene 14 and does not include a false positive prediction. Furthermore, our Radar Instance Transformer works reliably under versatile scenes, as illustrated in Fig. 6. Our approach reliably identifies moving instances under different and changing scenarios. However, for the intersection scene, the measured Doppler velocities of the agents are small, and the reliable identification of all distant moving instances is difficult. Since the Doppler velocity is the radial velocity of the detections, the tangential movement is not covered. Despite that fact, the network accurately differentiates between moving and static agents. Furthermore, our approach identifies the oncoming car in rainy scene 79 precisely. The scene illustrates the handling of noise, which is best seen for the Doppler velocity vectors for off-road detections belonging to the static class, as depicted in the ground truth. Additionally, our approach identifies multiple moving agents in the urban environment. The different scenarios further illustrate the sparsity and the changing density of the radar scan. Fig. 5 illustrates further qualitative results on the View-of-Delft validation set of our approach. Overall, our approach performs well despite the various difficulties.

D. Ablation Studies on the SAFE Module

The second experiment evaluates our sequential attentive feature encoding and illustrates that our module improves performance by efficiently including valuable temporal information within the features of the current point cloud. For this experiment, we vary the number of input scans for our Radar Instance Transformer and evaluate the pose compensation as
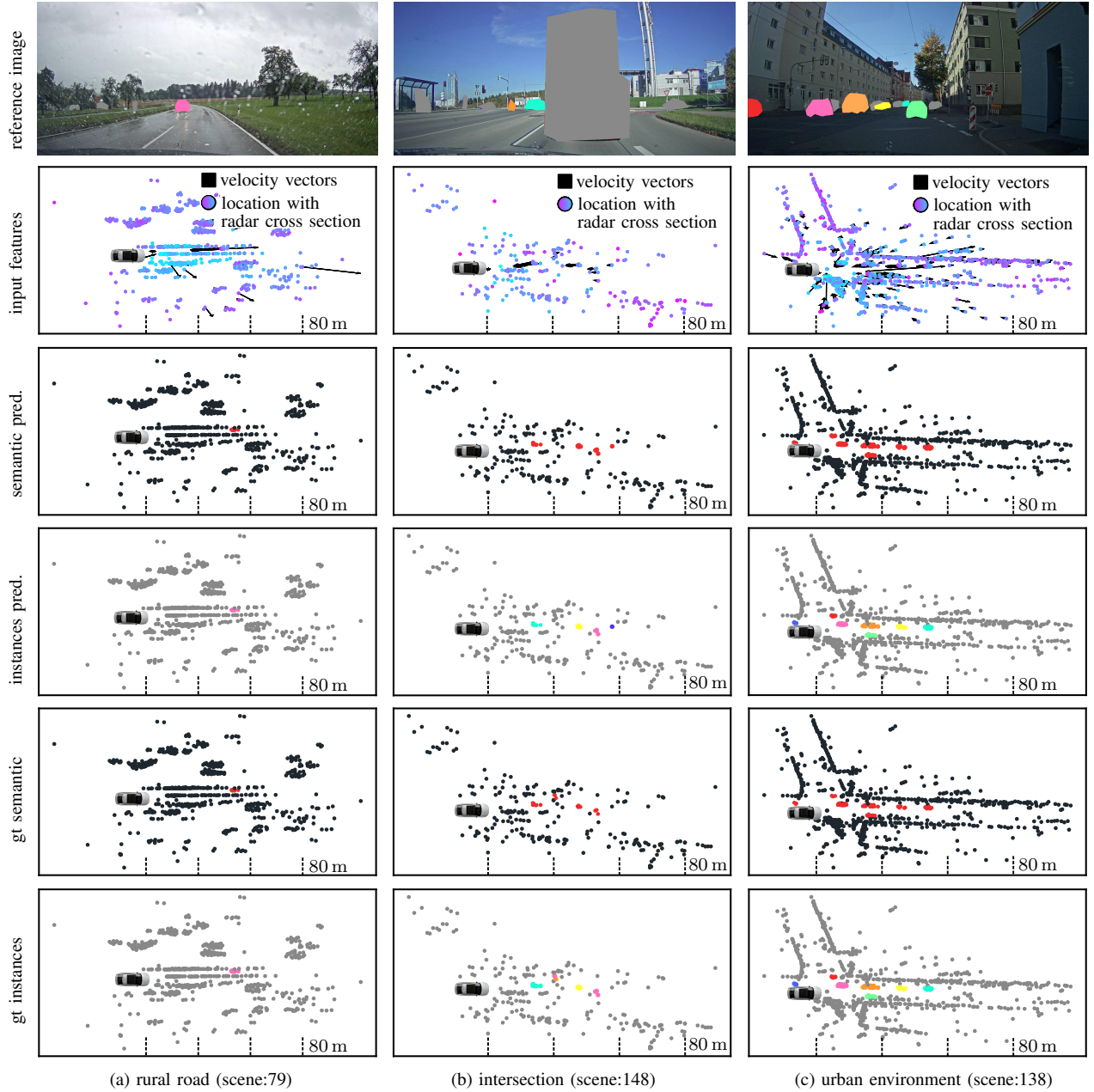
Fig. 6: Qualitative results of our Radar Instance Transformer on the RadarScenes test set. The reference camera images are anonymized. The colors in the illustration of the instance predictions (pred.) and the ground truth (gt) instances have the same color. The colors in the image correspond if the object is visible. The results illustrate the performance within different scenes, including a rural road on a rainy day (a), an intersection with agents crossing (b), and an urban environment with multiple agents (c).

depicted in Tab. IV. The temporal information improves the performance with the best result for five additional previous scans, i.e., $T = 5$. However, the training time, memory consumption, and runtime increase with adding more additional scans. Therefore, we select $T = 2$ as the best trade-off since our method performs well in terms of PQ^{mov} and IoU^{mov} and reduces the inference time. Furthermore, to save resources, the training time decreases by more than a factor of two compared to $T = 4$. The enhancement by temporally enriched features is directly visible by the increase of more than 2 absolute percentage points by including $T = 2$ previous scans instead of none. Furthermore, we argue that recent scans include the

most important information to identify moving instances, leading to a slight decrease in accuracy for $T = 3$. Additionally, the pose compensation does not compensate for the relative movement of instances, leading to false associations over time. Thus, the recent scans contain the most valuable information, and the IoU^{mov} shows just small improvements. To verify that the temporal information supports the identification of static points, including noise, we determine the IoU^{stat} for $T = 0$ and $T = 2$ resulting in 99.4 and 99.5, respectively. We assume that the changing appearance of noise can be identified within the temporal domain, and hence, IoU^{stat} improves. To elaborate on the improvement in detail, separate labels for

#	local similarity $(\mathbf{S}^{\text{loc}})$	global similarity $(\mathbf{S}^{\text{glob}})$	IoU$^{\text{mov}}$	PQ$^{\text{mov}}$
[A]			81.6	-
[B]	✓		83.7	-
[C]		✓	83.2	75.6
[D]	✓	✓	**84.4**	**76.8**
global: with positional encoding			82.6	74.2
local: static included			82.7	74.0

TABLE V: Influence of the different components of our instance transformer head in terms of IoU$^{\text{mov}}$ and PQ$^{\text{mov}}$ on the RadarScenes validation set.

	PQ$^{\text{mov}}$	SQ$^{\text{mov}}$	RQ$^{\text{mov}}$
r=3.0	70.4	94.5	74.5
r=4.0	73.4	95.3	77.0
r=5.0	75.1	**95.4**	78.7
r=6.5	76.6	95.1	80.6
r=7.5	**76.8**	95.1	**80.8**
r=8.0	76.7	95.0	**80.8**
Ours (r=7.0)	**76.8**	95.1	**80.8**

TABLE VI: Ablation study on the graph-based instance assignment on the RadarScenes validation set.

model	IoU$^{\text{mov}}$	PQ$^{\text{mov}}$
Ours + Point Voxel Transformer [89]	80.0	69.0
Ours + Stratified Transformer [82]	82.2	70.1
Ours w/o top level	81.9	73.7
Ours	**84.4**	**76.8**

TABLE VII: Ablation study on the backbone design and comparison to recently best-performing backbones on the RadarScenes validation set.

static and noise are required, which are not available.

In the second step, we remove the pose compensation, which helps to align the previous and the current scan. Here, the decrease in PQ$^{\text{mov}}$ is small. One advantage of radar sensors is the often higher frame rate compared to LiDAR sensors, which leads to smaller transformations between scans. In addition, we expect that the self-attention mechanism adaptively controls the information exchange and helps to extract valuable features, which underlines the advantages of our SAFE module.

E. Ablation Studies on the Instance Transformer Head

The third experiment is presented to illustrate that our instance transformer head incorporates important instance information and enhances the performance of moving instance segmentation. Notably, we do not report the PQ$^{\text{mov}}$ for the models without the global attentive similarity quantity because we cannot perform the partitioning based on the learned weights, and thus we compare the results for this experiment based on the IoU$^{\text{mov}}$. To assess the benefits of the proposed attentive similarity quantities, we remove the respective modules for the global and local computation, as depicted in Tab. V. We first remove the local and global attentive similarity in configuration [A], which reduces the IoU$^{\text{mov}}$ by 2.8 absolute percentage points. In ablation [B], we add the local attention, which already improves the IoU$^{\text{mov}}$ compared to [A]. We assume that the knowledge of which point within the local neighborhood belongs to the same instance helps to differentiate between instances and also to identify moving detections reliably. Compared to [A], the added global attentive similarity in [C] improves the performance. The global context is beneficial for identifying moving instances. Furthermore, false positive detections of moving detection are penalized, which enhances moving object segmentation. The combined local and global attentive similarity [D] combines both properties and further improves PQ$^{\text{mov}}$.

To analyze our design decisions, we add the positional encoding for the global attentive similarity. Since the relative position grows large in magnitude in the global context, as mentioned in Sec. III-C, we assume that the local context is lost, which is essential to differentiate between nearby points and hence does not improve accuracy. Additionally, we exclude the static point from the local similarity in our method, which means that for all static points, the predicted attentive similarity quantity has to be 0. We presume that the focus is clearly on moving instances, which is not the case if we

include the static points and thus decrease PQ$^{\text{mov}}$. In conclusion, the instance transformer head improves performance by exploiting local and global context to extract valuable features from noisy radar point clouds.

F. Ablation Studies on the Instance Assignment

The ablation study presented in this section supports our claim that attention-based graph partitioning enhances instance assignments without class-dependent information. We evaluate the hyperparameter r for the radius graph, as depicted in Tab. VI. Our method achieves good performance for different radii with the best PQ$^{\text{mov}}$ for $r = 7\,\text{m}$. Compared with the bandwidths for mean shift algorithms in Sec. V-C, our approach is able to utilize a larger radius. One reason for a larger radius is that we do not use an offset prediction and still want to reliably detect large moving instances. But this could lead to a combination of instances, especially small instances with nearby larger instances. However, the major advantage of our attention-based graph partitioning is that the global attentive similarity quantity can effectively solve the interconnection between these instances. We additionally evaluated the performance of the most vulnerable road users, pedestrians, which often comprise single detections to show that the interconnection can be reliably solved. However, potential future work might address the changing resolution of radar point clouds based on the distance to the sensor to refine the instance assignment. Within the evaluated range of radii, the PQ$^{\text{mov}}$ for pedestrians remains stable at 84.7, which illustrates the strong capability of our approach. In comparison, clustering often relies on accurate offset predictions and class-specific properties [51] to reliably identify instances of varying size. Hence, our attention-based graph partitioning helps to identify moving instances reliably.

G. Ablation Studies on the Backbone Architecture

Finally, we analyze our method with respect to the ability to extract valuable features by keeping the full resolution of the

model	parameters (M)	mean runtime (ms)
4DMOS [46]	1.8	14.0
Point Voxel Transformer [89]	6.2	47.1
Stratified Transformer [82]	8.0	34.4
DS-Net [30]	50.2	52.3
Mask3d [62]	39.6	85.2
Gaussian Radar Transformer [87]	8.4	24.0
Ours	3.8	31.7

TABLE VIII: The evaluation of the number of parameters and the mean runtime of the models on an Nvidia RTX A6000 GPU based on 1000 randomly sampled point clouds of the RadarScenes validation set.

Stages of our model	parameters (M)	mean runtime (ms)
Pre-processing	-	5.8
SAFE	0.006	2.7
Backbone	3.8	15.7
Instance Transformer Head	0.02	1.2
Graph Partitioning	-	6.3

TABLE IX: The detailed evaluation of the number of parameters and the mean runtime of the individual building blocks of our model on an Nvidia RTX A6000 GPU based on 1000 randomly sampled point clouds of the RadarScenes validation set.

point cloud throughout the network. The results in Tab. VII verify that removing the intermediate transformer blocks on the top levels of our model harms the accuracy. These building blocks are the additional modules compared to the widely-used U-Net [57, 93] architecture. Furthermore, we extend the two best-performing non-radar-specific backbones, which follow the U-Net architecture, with our model-agnostic modules to illustrate the advancements of our approach. The full-resolution processing and the extraction of additional high-level features outperform other methods. Nevertheless, our model-agnostic sequential attentive feature encoding and instance transformer head enhance the performance of the Stratified Transformer [82] and the Point Voxel Transformer [89], which illustrates that these modules help to improve scene interpretation in radar scans.

H. Runtime

Finally, we analyze the runtime and the number of parameters of the different methods. We measured the runtimes on an AMD Threadripper CPU with an Nvidia RTX A6000 GPU. We utilize an optimized farthest point sampling and kNN algorithm in C++ to speed up the inference of the algorithms. Since the point clouds differ in the number of detections, we evaluate 1,000 scans that are randomly selected from the validation set. For our approach, we utilize the best-performing configuration and include the $T = 2$ previous scans.

All results are detailed in Tab. VIII. The mean runtime of our approach is 31.7 ms, which is equal to 31 Hz, and thus faster than the frame rate of 17 Hz of the sensor. The Gaussian Radar Transformer has the lowest runtime of the transformer-based models in our experiments with 24 ms but utilizes twice as many parameters. We argue that the runtime depends on the different specific implementations of the respective attention mechanism. The Stratified Transformer [82] uses a more complex mechanism that induces further latencies compared to the simple vector attention of the Gaussian Radar Transformer. Additionally, we observe that the runtime scales with the number of points. To illustrate that aspect, we calculated the minimum and maximum runtime of our model, which are 23 ms and 221 ms, respectively. For all models, scans without any moving object prediction do not need clustering, which reduces runtime. However, with the increasing number of moving predictions, our global attentive association matrix gets more complex, and therefore, the runtime scales with the number of points and moving predictions. Furthermore, the

runtime scales with the number of T previous scans, which increases to 35.3 ms for $T = 5$. Future research can address these issues to further decrease the runtime. Another option to reduce the runtime is the replacement of the commonly used kNN algorithm. Since the resolution of radar point clouds is dependent on the distance to the sensor, the adjustment of the sampling might address this limitation and additionally improve performance.

To get an in-depth analysis, we additionally report the runtime and parameters of the individual blocks in Tab. IX. We observe that the processing of the intermediate building blocks of our backbone is suboptimal, and additional explicit parallelization can enhance the inference time. Nevertheless, we achieve state-of-the-art performance with an adequate runtime, outperforming all other methods except Gaussian Radar Transformer and DS-Net, while we include temporal information and use fewer parameters.

In summary, our evaluation suggests that our method provides competitive moving instance segmentation in sparse radar point clouds by efficiently including temporal information. The attentive similarity quantities encode valuable information and enable class-agnostic, graph-based instance assignment, outperforming state-of-the-art approaches. Thus, we support all our claims through this experimental evaluation.

VI. CONCLUSION

In this paper, we presented a novel approach for moving instance segmentation in sparse and noisy radar point clouds that shows strong performance and outperforms a large set of baseline methods. Our method efficiently exploits temporal information and overcomes the limitations of passing aggregated scans through the whole network. We utilize the self-attention mechanism throughout the network to extract valuable features and introduce attentive graph-based instance partitioning. This allows us to successfully identify moving agents and enhance the feature extraction by incorporating local and global instance knowledge. We furthermore established a new benchmark based on the RadarScenes dataset, which allows further comparisons with future work, providing comparisons to other methods and supporting all claims made in this paper. The experiments suggest that the different parts of our approach are essential to achieve good performance on moving instance segmentation. Overall, our approach outperforms the state of the art, taking a step forward towards reliable moving instance segmentation and sensor redundancy for autonomous vehicles and robots.

References

[1] N. Abraham and N.M. Khan. A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation. In *Proc. of the IEEE Intl. Symposium on Biomedical Imaging*, 2019.

[2] D. Adolfsson, M. Magnusson, A. Alhashimi, A.J. Lilienthal, and H. Andreasson. Lidar-Level Localization With Radar? The CFEAR Approach to Accurate, Fast, and Robust Large-Scale Radar Odometry in Diverse Environments. *IEEE Trans. on Robotics (TRO)*, 39(2):1476–1495, 2023.

[3] J.L. Ba, J.R. Kiros, and G.E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[4] S. Baur, D. Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger. SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[5] 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.

[6] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

[8] R.J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Proc. of Pacific-Asia. Conf. on Knowledge Discovery and Data Mining*, 2013.

[9] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. Montiel, and J.D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. on Robotics (TRO)*, 37(6):1874–1890, 2021.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-End Object Detection with Transformers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[11] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang. Hierarchical Aggregation for 3D Instance Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[12] 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)*, 6:6529–6536, 2021.

[13] 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.

[14] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu. (AF)2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] 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.

[16] C. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 24:603–619, 2002.

[18] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In *Proc. of the Intl. Symp. on Visual Computing*, 2020.

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

[20] F. Ding, Z. Pan, Y. Deng, J. Deng, and C.X. Lu. Self-Supervised Scene Flow Estimation With 4-D Automotive Radar. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):8233–8240, 2022.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[22] 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.

[23] H. Fan, Y. Yang, and M. Kankanhalli. Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] S. Gasperini, M.A.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.

[25] 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.

[26] M.H. Guo, J. Cai, Z.N. Liu, T.J. Mu, R.R. Martin, and S. Hu. PCT: Point Cloud Transformer. *Computational Visual Media*, 7(2):187–199, 2021.

[27] L. Han, T. Zheng, L. Xu, and L. Fang. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] T. He, C. Shen, and A. van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[29] D. Hendrycks and K. Gimpel. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *arXiv preprint:1606.08415*, 2016.

[30] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu. LiDAR-based 4D Panoptic Segmentation via Dynamic Shifting Network. *arXiv preprint*, arXiv:2203.07186, 2022.

[32] 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.

[33] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arxiv:1502.03167*, 2015.

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

[35] Y. Kittenplon, Y.C. Eldar, and D. Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[36] T. Kreutz, M. Mühlhäuser, and A.S. Guinea. Unsupervised 4D LiDAR Moving Object Segmentation in Stationary Settings With Multivariate Occupancy Time Series. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2023.

[37] L. Kreuzberg, I.E. Zulfikar, S. Mahadevan, F. Engelmann, and B. Leibe. 4D-StOP: Panoptic Segmentation of 4D LiDAR using Spatio-temporal Object Proposal Generation and Aggregation. In *Proc. of the Europ. Conf. on Computer Vision Workshops*, 2022.

[38] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In *Proc. of the Intl. Conf. on Artificial Neural Networks*, 1995.

[40] 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.

[41] Y. Li, S. Si, G. Li, C.J. Hsieh, and S. Bengio. Learnable Fourier Features for Multi-dimensional Spatial Positional Encoding. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[42] 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 (ICCV)*, 2021.

[43] J. Liu, W. Xiong, L. Bai, Y. Xia, T. Huang, W. Ouyang, and B. Zhu. Deep Instance Segmentation with Automotive Radar Detection Points. *IEEE Trans. on Intelligent Vehicles*, 2022.

[44] X. Liu, X. Li, D. Guo, S. Tan, H. Liu, and F. Sun. Embodied multi-agent task planning from ambiguous instruction. In *Proc. of Robotics: Science and Systems (RSS)*, 2022.

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

[46] 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.

[47] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[48] S. Mohapatra, M. Hodaei, S.K. Yogamani, S. Milz, H. Gotzig, M. Simon, H. Rashed, and P. Maeder. LiMoSeg: Real-time Bird's Eye View based LiDAR Motion Segmentation. In *Proc. of the Intl. Conf. of Computer Vision Theory and Applications (VISAPP)*, 2022.

[49] V. Nair and G.E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2010.

[50] M.E. Newman. Modularity and community structure in networks. *Proc. of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[51] A. Palffy, J. Dong, J.F.P. Kooij, and D.M. Gavrila. CNN Based Road User Detection Using the 3D Radar Cube. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1263–1270, 2020.

[52] A. Palffy, E. Pool, S. Baratam, J.F.P. Kooij, and D.M. Gavrila. Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):4961–4968, 2022.

[53] C. Park, Y. Jeong, M. Cho, and J. Park. Fast Point Transformer. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[54] 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.

[55] G. Puy, A. Boulch, and R. Marlet. FLOT: Scene Flow on Point Clouds Guided by Optimal Transport. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[56] 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.

[57] 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.

[58] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

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

[60] R. Razani, R. Cheng, E. Li, E. Taghavi, Y. Ren, and L. Bingbing. GP-S3Net: Graph-based panoptic sparse semantic segmentation network. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[61] R.A. Rosu, P. Schütt, J. Quenzel, and S. Behnke. LatticeNet: Fast point cloud segmentation using permutohedral lattices. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] H. Shi, J. Wei, R. Li, F. Liu, and G. Lin. Weakly Supervised Segmentation on Outdoor 4D point clouds with Temporal Matching and Spatial Graph Propagation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[67] Y. Shi and K. Ma. SAFIT: Segmentation-Aware Scene Flow with Improved Transformer. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[68] 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, 2022.

[69] 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.

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

[71] 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.

[72] 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.

[73] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(10):3349–3364, 2020.

[74] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia. Associatively Segmenting Instances and Semantics in Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[75] Z. Wang, S. Li, H. Howard-Jenkins, V. Prisacariu, and M. Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2020.

[76] Y. Wei, Z. Wang, Y. Rao, J. Lu, and J. Zhou. PV-RAFT: point-voxel correlation fields for scene flow estimation of point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[77] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[78] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeeze-SegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[79] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R.B. Girshick. Early Convolutions Help Transformers See Better. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[80] C. Xie, Y. Xiang, A. Mousavian, and D. Fox. Unseen object instance segmentation for robotic environments. *IEEE Trans. on Robotics (TRO)*, 37(5):1343–1359, 2021.

[81] S. Xie, S. Liu, Z. Chen, and Z. Tu. Attentional shapecontextnet for point cloud recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[82] 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.

[83] 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 (ICCV)*, 2021.

[84] M. Xu, R. Ding, H. Zhao, and X. Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[85] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[86] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.H. Jiang, F.E. Tay, J. Feng, and S. Yan. Tokens-to-Token ViT: Training Vision Transformers From Scratch on ImageNet. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[87] 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.

[88] M. Zeller, V. S. Sandhu, M. Benedikt, J. Behley, M. Heidingsfeld, and C. Stachniss. Dataset for Moving Instance Segmentation Based on Radarscenes. *Zenodo*, https://doi.org/10.5281/zenodo.10203864, 2023.

[89] C. Zhang, H. Wan, X. Shen, and Z. Wu. PVT: Point-voxel transformer for point cloud learning. *Intl. Journal of Intelligent Systems*, 37(12):11985–12008, 2022.

[90] J. Zhang and S. Singh. LOAM: Lidar odometry and mapping in real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.

[91] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[92] 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.

[93] 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 (ICCV)*, 2021.

[94] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y.N. Wu. Multi-Agent Tensor Fusion for Contextual Trajectory Prediction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[95] Zhou, Yi and Liu, Lulu and Zhao, Haocheng and López-Benítez, Miguel and Yu, Limin and Yue, Yutao. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors*, 22(11):4208, 2022.

[96] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Matthias Zeller is a Ph.D. student at CARIAD SE, Mönsheim, Germany, and the Photogrammetry and Robotics Lab at the University of Bonn, headed by Cyrill Stachniss. He received his B.Eng. degree in automotive engineering from the University of Applied Sciences Esslingen, Germany, in 2018 and his M.Sc. degree in electromobility in 2021 from the University of Stuttgart, Germany. His current research focuses on radar-based scene understanding for self-driving vehicles.

Vardeep Singh Sandhu received his B.Tech. degree in Electronics and Communications from GNDU, India, in 2017 and the M.Sc. degree in Geodetic Engineering from the University of Bonn, Germany, in 2022. He is currently working as a Perception Engineer at Motor AI, Germany. His current field of research and development includes the deployment of detection and segmentation networks on edge devices.

Benedikt Mersch is a Ph.D. student at the Photogrammetry and Robotics Lab at the University of Bonn, headed by Cyrill Stachniss. He studied electrical engineering, focusing on control and electric power systems, and received his M.Sc. degree from the Hamburg University of Technology, Germany, in 2020. During his Master's, he spent one semester at the University of Berkeley, California. His current research focuses on SLAM and spatio-temporal LiDAR perception for self-driving cars.

Jens Behley received his M.Sc. degree in 2009 and his Ph.D. degree in 2014, both from the Department of Computer Science at the University of Bonn, Germany. Since 2016 he is a postdoctoral researcher at the Photogrammetry and Robotics Lab at the University of Bonn, Germany. He finished his habilitation in 2023 and is a lecturer at the Photogrammetry and Robotics Lab, University of Bonn, Germany. His area of interest lies in the area of perception for autonomous vehicles, deep learning for semantic interpretation, and LiDAR-based SLAM and localization.

Michael Heidingsfeld received his Dipl.-Ing. degree in mechanical engineering from the Karlsruhe Institute of Technology, Germany, in 2012 and his Ph.D. degree in control engineering from the University of Stuttgart, Germany, in 2018. He is currently with CARIAD SE, working on the development of automated parking functions. His research interests include trajectory planning, motion control, and algorithms for localization and mapping applied to automated and autonomous vehicles.

Cyrill Stachniss is a full professor at the University of Bonn, a Visiting Professor in Engineering at the University of Oxford, and with the Lamarr Institute for Machine Learning and AI. He is the Spokesperson of the DFG Cluster of Excellence PhenoRob at the University of Bonn. Before his appointment in Bonn, he was with the University of Freiburg and ETH Zurich. His research focuses on probabilistic techniques and learning approaches for mobile robotics, perception, and navigation. The main application areas of his research are agricultural robotics, self-driving cars, and service robots. He has co-authored over 300 peer-reviewed publications and has coordinated multiple large-scale research projects on the national and European levels.