2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Radar Velocity Transformer: Single-scan Moving Object Segmentation in Noisy Radar Point Clouds

Matthias Zeller Vardeep S. Sandhu Benedikt Mersch Jens Behley Michael Heidingsfeld Cyrill Stachniss

***Abstract*— The awareness about moving objects in the surroundings of a self-driving vehicle is essential for safe and reliable autonomous navigation. The interpretation of LiDAR and camera data achieves exceptional results but typically requires to accumulate and process temporal sequences of data in order to extract motion information. In contrast, radar sensors, which are already installed in most recent vehicles, can overcome this limitation as they directly provide the Doppler velocity of the detections and, hence incorporate instantaneous motion information within a single measurement. In this paper, we tackle the problem of moving object segmentation in noisy radar point clouds. We also consider differentiating parked from moving cars, to enhance scene understanding. Instead of exploiting temporal dependencies to identify moving objects, we develop a novel transformer-based approach to perform single-scan moving object segmentation in sparse radar scans accurately. The key to our Radar Velocity Transformer is to incorporate the valuable velocity information throughout each module of the network, thereby enabling the precise segmentation of moving and non-moving objects. Additionally, we propose a transformer-based upsampling, which enhances the performance by adaptively combining information and overcoming the limitation of interpolation of sparse point clouds. Finally, we create a new radar moving object segmentation benchmark based on the RadarScenes dataset and compare our approach to other state-of-the-art methods. Our network runs faster than the frame rate of the sensor and shows superior segmentation results using only single-scan radar data.**

I. Introduction

Self-driving vehicles need to distinguish moving from stationary objects to safely navigate in dynamic, real-world environments. To enable redundancy and overcome the shortcomings of individual sensors, the sensor suites of autonomous vehicles are versatile, including cameras, LiDARs, and radars. The widely explored camera and LiDAR sensors utilize temporal sequences of input data to segment moving objects, often neglecting the valuable information of radar data. The Doppler velocity provided by a radar enables the identification of moving objects in single scans and radar sensors work under adverse weather, including rain, fog, and snow where other modalities encounter difficulties. A serious drawback, however, is that the radar scans are largely affected by noise due to multi-path propagation, ego-motion, and sensor noise. The noisy measurements frequently lead to false positives and make threshold-based moving object segmentation [32] unacceptable, as visualized in Fig. 1. We aim at investigating in this paper how the additional sensor information of the Doppler velocity can be exploited by learning-based approaches to enable reliable identification of moving objects in the environment. Furthermore, the radar cross section, which depends on the material properties and the structure of the detection, supports the differentiation of closely connected objects.

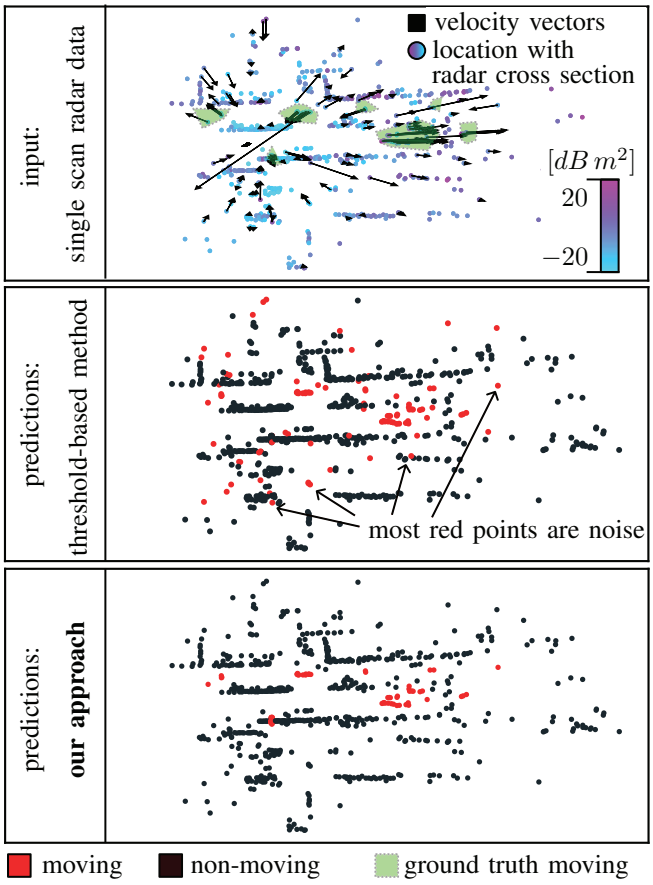

Fig. 1: Our learning-based approach enhance moving object segmentation (middle), from noisy, single-scan radar point clouds (top) compared to a velocity threshold determined on the validation set. Best viewed in color.

We investigate moving object segmentation in radar point clouds. This task requires differentiation between the detection of moving and stationary objects. To accurately differentiate between the two classes, we exploit single radar point clouds, including the valuable Doppler velocity and radar cross section. State-of-the-art methods for moving object segmentation for camera and LiDAR data rely on the elaboration of temporal dependencies in videos [10] or aggregated

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.

residuals between previous scans [18]. The processing of multiple frames induces latency which is unsuitable for a task requiring immediate information about the environment such as collision avoidance. Therefore, we investigate the processing of single, sparse radar point clouds by exploiting additional and valuable radar sensor information to leverage the full potential of radar sensors.

The main contribution of this paper is a novel learning-based approach that accurately predicts moving objects in sparse, single-scan radar point clouds. Our approach, called Radar Velocity Transformer, predicts for each point in the input radar scan the semantic label of moving or non-moving. To classify the individual detection and extract valuable point-wise features, we introduce the velocity encoding in each module of our network. The encoding of the velocity enhances the performance by injecting important information throughout the network. We optimize the feature aggregation in the decoder part by our transformer-based upsampling to adaptively merge features and capture complex local structures in sparse point clouds. Furthermore, we reorganized the RadarScenes [34] dataset providing semantic classes for individual detection, which we transfer into moving and non-moving labels establishing a single-scan benchmark.

In sum, we make two key claims: (i) Our approach is able to accurately perform moving object segmentation in single-scan, noisy radar point clouds and enhance the state of the art in moving object segmentation without exploiting temporal dependencies; (ii) The velocity encoding throughout the network and the transformer-based upsampling are essential to derive highly discriminative features and adaptively aggregate information to enhance accuracy.

II. Related Work

Moving object segmentation in point clouds can be categorized into map-based [17], [24], [31] and map-free approaches [18], [23], [37]. Current advancements focus on the latter to work online and remove the burden of pre-built maps, which is further supported by scene flow estimation [1], [8], [11], [19], [36], [40], [42] and semantic segmentation [12], [16], [44], [46], [47], [52]. To differentiate between the methods, we distinguish between projection-based, voxel-based, point-based, transformer-based, and hybrid methods.

Projection-based methods are introduced to utilize the successful convolutional neural networks (CNNs) on 3D data. For example, Chen et al. [4] first project the LiDAR point clouds into 2D range images and provide the residual images of previous scans as input to SalsaNext [6] to perform moving object segmentation on SemanticKITTI [2]. Kim et al. [18] extend the approach and improve state-of-the-art performance for moving object segmentation in LiDAR data by efficient data augmentation and the attention-based fusion module to combine semantic and motion features. The methods are highly efficient but face back projection artifacts when transferring the 2D predictions to the 3D point cloud which harms the accuracy.

Voxel-based methods keep the 3D information intact and hence reduce the limitations caused by back projection. Mersch et al. [23] adopted the Minkowski engine [5] and propose a receding horizon strategy to incorporate new scans in an online fashion and refine predictions by Bayesian filtering to enhance LiDAR moving object segmentation. However, the voxel-based methods inherently introduced discretization artifacts resulting in an information loss.

Point-based methods [21], [28], [39] allow for keeping the full spatial information of point clouds, which is desirable, especially for sparse point clouds. The pioneering work of Qi et al. [28], called PointNet, utilizes shared multi-layer perceptrons (MLPs) to directly consume point clouds and aggregate nearby information by max pooling functions. FlowNet3D [21], FlowNet3D++ [39], and FLOT [27] follow PointNet++ [29] and introduce dedicated architectures for point-based scene flow estimation in LiDAR point clouds. Schumann et al. [33] adapt the hierarchical grouping of PointNet++ [29] combining features from multiple scales and extending their approach by exploiting strong temporal dependencies by the aggregation of consecutive scans [35] to enhance semantic segmentation of sparse radar point clouds. Fan et al. [10] propose P4Transformer, which combines 4D point-based convolutions with video-level self-attention to merge related local areas spatially and temporally. The point-based method benefits from the transformer-based module since the self-attention mechanism [25], [43], [49] is invariant to permutation and thus inherently suitable to capture strong local and global dependencies and extract valuable features in point clouds.

Transformer-based methods dominate a variety of tasks from natural language processing to point cloud understanding by exploiting the powerful self-attention mechanism [7], [12], [20], [30], [38], [44], [50], [52]. Guo et al. [12] propose offset-attention with an implicit Laplace operator and normalization refinement to reduce the influence of noise and sharpen the attention weights. SAFIT [36] models relations on object and point level via transformers to estimate scene flow. Since the self-attention mechanism is computationally expensive, transformer-based networks benefit from effective sampling strategies to aggregate local features and reduce the computational cost. Besides the wide range of sampling algorithms [15], [41], [48], [49] the common method for downsampling is farthest point sampling following max pooling [29]. For upsampling, trilinear interpolation is usually the method of choice based on an inverse distance weighted average [29]. To further keep fined-grained position information throughout the network, Zhao et al. [52] introduce the trainable position encoding and adapt vector-based attention [51]. Stratified Transformer [44] extends the position encoding and aggregates long-range context by a window-based key-sampling strategy to enhance the accuracy.

In this paper, we follow recent advancements and propose a novel transformer-based moving object segmentation method for sparse and noisy radar data. In contrast to the related work, our newly introduced Radar Velocity Transformer extends the transformer layer and exploits

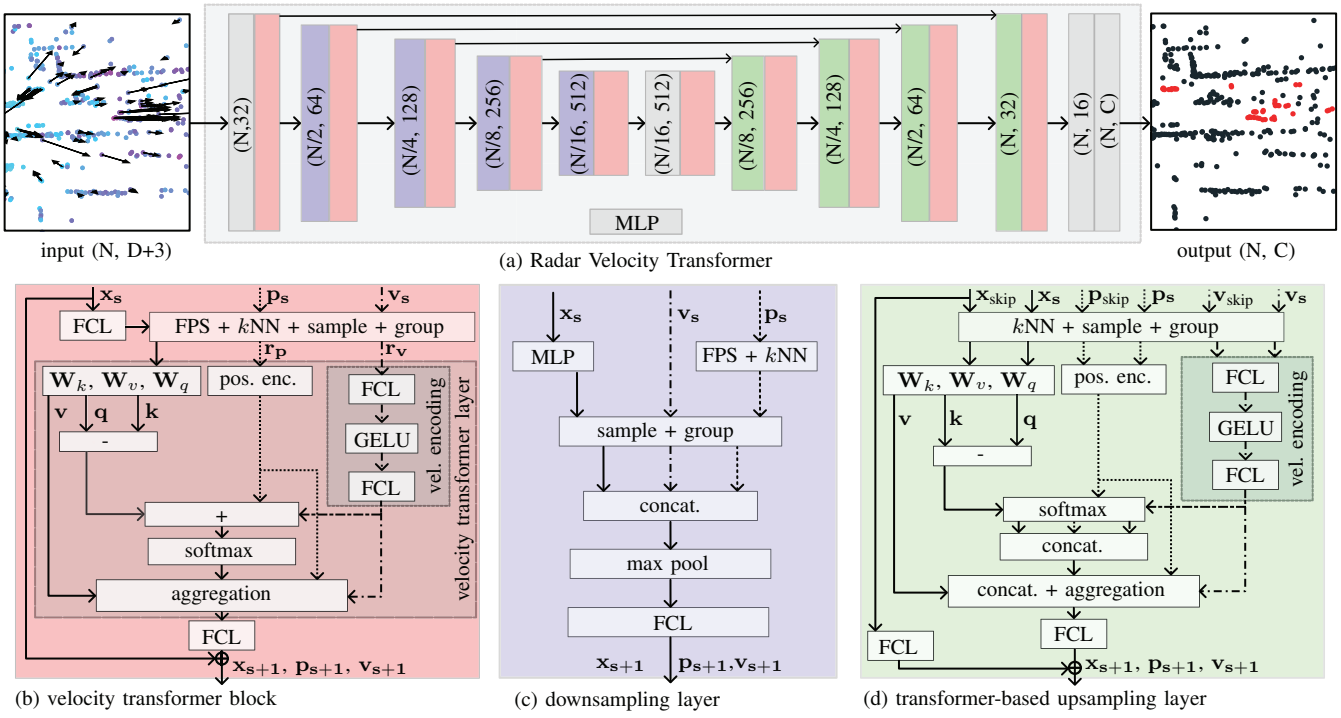

Fig. 2: The detailed design of each module of our Radar Velocity Transformer (a) shows the Radar Velocity Transformer, (b) the velocity transformer block with the velocity transformer layer, (c) the downsampling layer, and (d) the transformer-based upsampling layer. The different colors stand for the different building blocks. The tuples denote the number of points and feature channels in each stage. MLP: multi-layer perceptron, FCL: fully connected layer, vel. encoding: velocity encoding, pos. enc.: positional encoding, FPS: farthest point sampling, kNN: k-nearest neighbor, concat.: concatenation, W: weight matrices

the valuable velocity information throughout the network. Furthermore, our proposed network includes an advanced transformer-based upsampling strategy to capture complex local structures and enhances state-of-the-art performance for moving object segmentation in single-scan radar point clouds.

III. Our Approach

The goal of our approach is to achieve precise moving object segmentation in single-scan, sparse radar point clouds to enhance the environmental perception of autonomous vehicles. Fig. 2 illustrates our Radar Velocity Transformer (RVT), which is a transformer-based framework that builds upon the successful self-attention mechanism [38] and directly processes the input point cloud to omit information loss. We incorporate the valuable Doppler velocity information within each module and use the so-called velocity transformer layer as the central building block of each encoder-decoder stage. Furthermore, we introduce transformer-based upsampling modules to adaptively combine local context information to enable fine-grained feature extraction.

A. Velocity Transformer Layer

In sparse radar point clouds, the information of individual detections can be of great benefit for solving downstream tasks such as moving object segmentation. Therefore, we introduce a velocity transformer layer to enhance feature extraction, as illustrated in Fig. 2 (b).

The inputs are single-scan radar point clouds $\mathcal{P}_s$ with point coordinates $\mathbf{p}_i \in \mathbb{R}^2$, ego-motion compensated Doppler velocity $\mathbf{v}_i \in \mathbb{R}$, and point-wise features $\mathbf{x}_i \in \mathbb{R}^D$ with feature dimension, D.

Since the Doppler velocity provides essential information about the moving and non-moving parts of the environment, we incorporate the information as a central part of our velocity transformer layer. The idea is to support accurate moving object segmentation based on the relative velocity $\mathbf{r}^v_{i,j} \in \mathbb{R}^{N \times N_{\text{vtl}}}$ since this enables the differentiation of nearby points and the identification of moving objects. Hence, we process the relative velocity $\mathbf{r}^v_{i,j} = \mathbf{v}_i - \mathbf{v}_j$ by two fully connected layers and the Gaussian error linear unit (GELU) as an activation function [14] to include the information. The rest of our velocity transformer layer follows standard transformers [51], [52] and relies on the encoded representation of the input features $\mathbf{x}$. The queries $\mathbf{q}$, the keys $\mathbf{k}$, and the values $\mathbf{v}$ are determined by multi-layer perceptrons with the corresponding weight matrices $\mathbf{W}_q \in \mathbb{R}^{D \times D}$, $\mathbf{W}_k \in \mathbb{R}^{D \times D}$ and $\mathbf{W}_v \in \mathbb{R}^{D \times D}$, as follows:

$$\mathbf{q} = \mathbf{W}_q \mathbf{x}, \qquad \mathbf{k} = \mathbf{W}_k \mathbf{x}, \qquad \mathbf{v} = \mathbf{W}_v \mathbf{x}. \tag{1}$$

As relation functions $\mathbf{g}$ for the queries and the keys, we utilize subtraction. For the positional encoding, we adapt the approach of Zhao et al. [52] and process the relative position $\mathbf{r}^p_{i,j} = \mathbf{p}_i - \mathbf{p}_j$ by two fully connected layers and the GELU. To calculate the attention scores $\mathbf{a}_{i,j}$ within local areas, we adapt vector attention [51] to allow for a weighting of individual feature channels. We determine the local areas with N_{vtl} points by farthest point sampling and k-nearest neighbor (kNN) algorithm. To enable fine-grained informa-

tion aggregation, we calculate attention weights based on the sum of the relation of queries and keys $\mathbf{g}(\mathbf{q}_i, \mathbf{k}_j)$, the relative position encoding $\mathbf{r}^p_{i,j}$, and the relative velocity encoding $\mathbf{r}^v_{i,j}$. The final attention weights are determined by the softmax function:

$$\mathbf{a}^{i,j} = \text{softmax}(\mathbf{g}(\mathbf{q}_i, \mathbf{k}_j) + \mathbf{r}^p_{i,j} + \mathbf{r}^v_{i,j}). \tag{2}$$

Additionally, we add the relative velocity encoding to the values and the relative position encoding to derive the combined values $\mathbf{v}^c_{i,j} = \mathbf{v}_{i,j} + \mathbf{r}^p_{i,j} + \mathbf{r}^v_{i,j}$, which include and update the valuable information throughout the network. To derive the weighted features $\mathbf{y}$, we calculate the sum of the element-wise multiplication:

$$\mathbf{y}_j = \sum_{i=1}^{N_{\text{vtl}}} \mathbf{a}_{i,j} \odot \mathbf{v}^c_i, \tag{3}$$

within the local areas. The aggregated features which are enriched by the velocity encoding $\mathbf{y}$ are directly processed by the following module to reduce the computational cost within the velocity transformer layer.

B. Velocity Transformer Block

Our velocity transformer block is a residual block [13], similar to the point transformer block [52], that embeds the velocity transformer layer in the center of two fully connected layers, as depicted in Fig. 2 (b). We add LayerNorm [45] and a GELU activation function for each fully connected layer. The features $\mathbf{x}_i$ are processed by the velocity transformer layer and the linear layers to enrich the information of individual points within local areas. The velocity and position data are utilized to determine the relative encodings but are not further transformed to keep unaltered information throughout the network.

C. Downsampling Layer

The downsampling layer reduces the cardinality of the point cloud $\mathcal{P}_{s+1} \subset \mathcal{P}_s$ after each stage s and has to keep the most relevant information intact. Following Qi et al. [29], we adapt the max pooling operation depicted in Fig. 2 (c). We first process the feature vector by a linear layer. To derive the local areas, we sample and group the points by farthest point sampling and kNN algorithm. The features and the Doppler velocity values are sampled and grouped accordingly. To also induce valuable velocity information in the downsampling, we concatenate the features, the position, and the velocity information. Afterward, we apply max pooling to aggregate the information and process the feature vector by a fully connected layer. We reduce the number of points N_s by a factor of 2 and keep the position and velocity information of the downsampled point cloud to enrich the information in deeper layers.

D. Transformer-based Upsampling Layer

The common upsampling method interpolates the $k = 3$ nearest neighbors based on an inverse distance weighted average [29] and combines these with the features of the skip connection. Especially at the boundaries of moving objects, the straightforward interpolation can result in a combination of features of different classes, which can harm the extraction of discriminative features. Hence, we argue that the upsampling and the aggregation of the features in the decoder part of the network are crucial to enhance accuracy, especially for sparse point clouds.

To adaptively merge the information of the two point clouds, we propose the transformer-based upsampling layer visualized in Fig. 2 (d). The idea is to enable the network to learn to concatenate important information by inter-attention to extract valuable features. The inputs are the output point cloud of the previous velocity transformer block $\mathcal{P}_s$, with the number of points N_s, which has to be upsampled, and the point cloud of the skip connection $\mathcal{P}_{\text{skip}}$ where $N_s \leq N_{\text{skip}}$.

Inspired by our velocity transformer layer, we first encode the features $\mathbf{x}_s$ as keys $\mathbf{k}$, and values $\mathbf{v}$ and the features $\mathbf{x}_{\text{skip}}$ as queries $\mathbf{q}$, following Eq. (1). To determine the relative position and velocity encoding, we calculate the k-nearest neighbors for the point cloud of the skip connection $\mathcal{P}_{\text{skip}}$ within the point cloud $\mathcal{P}_s$.

In the sample and grouping module, we compute the relative position and velocity of the correspondent points of the two point clouds. We determine the encodings by two fully connected layers with the GELU activation function. In contrast to the velocity transformer layer, we calculate individual attention weights for the relation of queries and keys $\mathbf{a}^{qk}_{i,j}$, the relative position encoding $\mathbf{a}^p_{i,j}$, and the relative velocity encoding $\mathbf{a}^v_{i,j}$ to enable fine-grained information aggregation and enhance accuracy. The individual attention weights are determined by the softmax function as follows:

$$\mathbf{a}^{qk}_{i,j} = \text{softmax}(\mathbf{g}(\mathbf{q}_i, \mathbf{k}_j)), \tag{4}$$

$$\mathbf{a}^p_{i,j} = \text{softmax}(\mathbf{r}^p_{i,j}), \tag{5}$$

$$\mathbf{a}^v_{i,j} = \text{softmax}(\mathbf{r}^v_{i,j}). \tag{6}$$

We concatenate the individual attention weights to derive the final attention scores $\mathbf{a}_{i,j} = (\mathbf{a}^{qk}_{i,j}, \mathbf{a}^p_{i,j}, \mathbf{a}^v_{i,j})$. To weight the respective information, the values $\mathbf{v}$ are concatenated with $\mathbf{r}^p_{i,j}$ and the velocity encoding $\mathbf{r}^v_{i,j}$ resulting in the combined values $\mathbf{v}^c_{i,j} = (\mathbf{v}_{i,j}, \mathbf{r}^p_{i,j}, \mathbf{r}^v_{i,j})$.

To derive the weighted features $\mathbf{y}$, we calculate the sum of the element-wise multiplication:

$$\mathbf{y}_j = \sum_{i=1}^{N_{\text{tus}}} \mathbf{a}_{i,j} \odot \mathbf{v}^c_i, \tag{7}$$

within local areas. The aggregated features $\mathbf{y}$ are processed by a fully connected layer to compress the features to the original feature dimension D with a learnable weight matrix $\mathbf{W}_y \in \mathbb{R}^{(D+12)\times D}$:

$$\mathbf{z} = \mathbf{W}_y \mathbf{y}, \tag{8}$$

where $\mathbf{z}$ are the updated features for the upsampled point cloud. The fully connected layer enables the information exchange of the individual parts and reduces the complexity of the succeeding modules.

The final output of the transformer-based upsampling layer is the sum of the features $\mathbf{x}_{\text{skip}}$ and $\mathbf{z}$, which incorporates

Method	Input	IoU
Threshold $\|v_i\| > t$	single-scan	35.1
4DMOS [23]	multiple-scan	73.1
Stratified Transformer [44]	single-scan	74.6
Our Radar Velocity Transformer	single-scan	**81.3**

TABLE I: Moving object segmentation results on the RadarScenes test set in terms of IoU for the moving class. The threshold $t = 0.92\,m/s$ is determined on the validation set and afterwards applied to the test set [32].

the valuable information of both point clouds to derive discriminative features.

E. Network Architecture

We build our network architecture based on the widely-used U-Net [29], [52] with an encoder-decoder architecture including skip connections as illustrated in Fig. 2. The input to the network are the features $\mathbf{x}_i^i$, the position information $\mathbf{p}_i$ with two spatial coordinates x_i^C, y_i^C, and the ego-motion compensated Doppler velocity v_i. The features include the position, the velocity, and additionally, the radar cross section σ_i resulting in a 4-dimensional vector $\mathbf{x}_i^i = (x_i^C, y_i^C, v_i, \sigma_i)$. The input $\mathbf{x}_i^i$ are processed in each stage s resulting in the features $\mathbf{x}_s$. The input is first processed by an MLP before being passed to the first velocity transformer layer. The per-point features are gradually increased within each stage from 32 to 64, 128, 256, and 512. The sampling operations change the cardinality by a factor of 2 resulting in [N, N/2, N/4, N/8, N/16] points for the respective stage. The final output is determined by an MLP with two linear layers to obtain per-point logit values. The individual stages of our architecture each comprise one single velocity transformer block to build an efficient network.

F. Implementation Details

The Radar Velocity Transformer is implemented in PyTorch [26]. We train our model over 50 epochs with AdamW [22] optimizer with an initial learning rate of 0.0005 and a cosine annealing learning rate scheduler [22]. We combine the Lovász loss [3] and the weighted cross-entropy for which we follow the approach by Schumann et al. [35] and set the weights for moving objects to 8.0 and for static ones to 0.5. The local areas for the velocity transformer layer are set to $N_{\text{vtl}} = 16$ and for the transformer-based upsampling to $N_{\text{tus}} = 12$. We train the network with one Nvidia A100 GPU and a batch size of 128. To reduce overfitting, we further apply data augmentation, using jitter, scaling, rotation, and instance augmentation.

IV. Experimental Evaluation

The main focus of this work is an accurate, single-scan moving object segmentation in sparse and noisy radar point clouds. We present our experiments to show the capabilities of our method to reliably segment moving objects. The results of our experiments also support our key claims, which are: Our approach (i) segments moving objects in radar point clouds more precisely compared to state-of-the-art methods and (ii) the velocity encoding and the transformer-based upsampling enhance the accuracy by incorporating valuable information throughout the network and.

A. Experimental Setup

We utilize the RadarScenes [34] dataset, to train and evaluate our model since this dataset is the only open, large-scale radar dataset that includes per-point annotations for moving objects under different weather conditions and driving scenarios. The dataset is split into 130 sequences for training and 28 for validation. To construct a test set, we split the RadarScenes validation set into 6 sequences for validation (6, 42, 58, 85, 99, 122) and the remaining 22 sequences for testing.

The RadarScenes [34] dataset includes four radar sensors. To provide information on the surroundings of the vehicle, we need to merge the point clouds of the individual sensors into one central radar scan. Since the pose information, the measurement time, and the coordinates of the individual detection are given, we merge the data within a common coordinate system.

Following Chen et al. [4], we utilize the intersection over union (IoU) [9], where $IoU = \frac{TP}{TP+FN+FP}$ with the number of true positive (TP), false positive (FP), and false negative (FN) predictions for moving objects to evaluate the methods.

B. Moving Object Segmentation Performance

The first experiment evaluates the performance of our approach and its outcome supports the claim that our approach enhances state-of-the-art moving object detection in sparse and noisy radar point clouds utilizing only single scans.

To compare the results, we select the recently best-performing point-based segmentation method, the Stratified Transformer [44], which utilizes single scans, the 4DMOS network [23] for LiDAR moving object segmentation which does not use the range representation because this is incompatible with the 2D coordinates, and a simple threshold for the velocity determined on the validation set [32]. For specific information on the training regime of the two networks, we refer to the original papers [23], [44].

The Radar Velocity Transformer outperforms the two existing approaches and the learning-based methods are superior compared to the threshold-based method, as displayed in Tab. I. The difference between the learning-based approaches and the threshold-based method illustrates the necessity of advanced models to perform moving object segmentation in noisy radar point clouds. Additionally, the transformer-based methods enhance the performance compared to the voxel-based 4DMOS, which suggests that discretization artifacts lead to information loss that cannot be compensated by additional temporal information of consecutive radar scans. The feature input vector of Stratified Transformer and Radar Velocity Transformer both contain valuable velocity information. However, our Radar Velocity Transformer considerably improves the IoU for moving objects by 6.7 absolute percentage points and performs well under adverse weather conditions, as illustrated in Fig. 3.

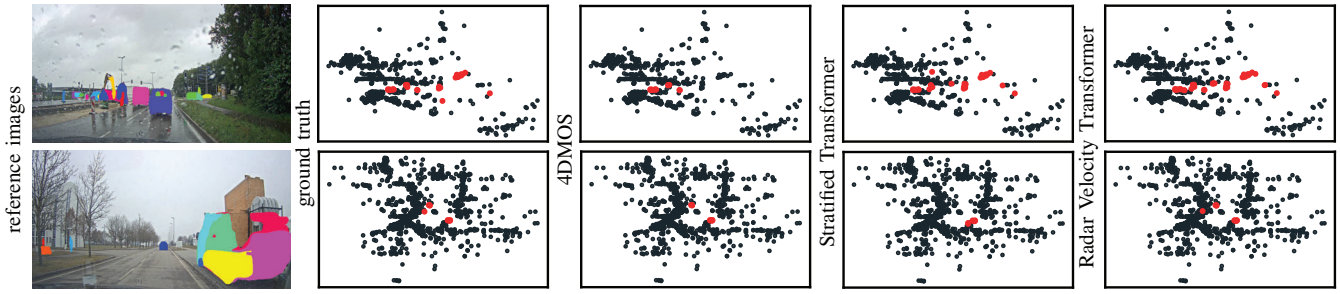

Fig. 3: Qualitative results of 4DMOS [23], Stratified Transformer [44], and our Radar Velocity Transformer on the test set of RadarScenes [34]. Red points indicate moving objects and black points belong to static objects.

#	velocity encoding	transformer-based upsampling	IoU
[A]			73.4
[B]		✓	75.2
[C]	✓		75.6
[D]	✓	✓	**77.4**

TABLE II: Influence of the different components in terms of IoU for moving objects on the RadarScenes validation set.

C. Ablation Study on Network Components

The second experiment, the ablation study on network components, evaluates the influence of the velocity encoding and transformer-based upsampling on the performance to support our second claim that our proposed modules each contribute to the improvements in terms of IoU. The combined results of the ablation study on the validation set are listed in Tab. II.

To assess the benefits of transformer-based upsampling, we replace the module with the commonly used trilinear interpolation based on an inverse distance weighted average [29]. Since the velocity encoding is new and the information of the velocity of the individual detection is present in the feature vector $\mathbf{x}$, we remove the velocity encoding to evaluate the influence on the IoU.

Ablation [A], we replace the upsampling and remove the velocity encoding which leads to a decrease in terms of IoU by 4 absolute percentage points. In ablation [B], we add the transformer-based upsampling, which enables an adaptive feature aggregation of the two point clouds and leads to an improvement of IoU by 1.8 absolute percentage points. In comparison to ablation [A], we add the velocity encoding throughout the network in [C], which enhances the performance. We assume that the velocity encoding is highly valuable since the fine-grained Doppler velocity information may be lost in high-level features of deeper layers. Hence, the specific task of moving object segmentation benefits from the velocity encoding. The final model of our Radar Velocity Transformer, represented in [D], further enhances the IoU by the usage of both, velocity encoding and transformer-based upsampling. We conclude that the additional information of the velocity encoding supports the aggregation of the features for the upsampling and hence leads to the best results.

As an additional experiment, we replaced the concatenation of the transformer-based upsampling with an addition in our final Radar Velocity Transformer. The obtained IoU of 75.3 % indicates that the concatenation leads to a more fine-grained weighting of the individual channels and improves the performance. Additionally, we exploit transformer-based downsampling. However, this does not improve the overall performance and hence we keep the max pooling since it is more efficient and does not mix information, which is suitable for the downsampling of sparse point clouds.

D. Runtime

Finally, we analyze the runtime of our approach and show that our approach runs fast enough to support online processing in the vehicle. We tested our approach on an AMD Ryzon 5 CPU with an Nvidia GTX 1660 GPU. Our implementation includes an optimized farthest point sampling and kNN algorithm in C++ to speed up the inference. Since the point clouds differ in the number of detections, we evaluate 1,000 scans that are randomly selected from the validation set. The mean runtime is 0.012 s, which is equal to 83 Hz, and thus over 4x faster than the frame rate of 17 Hz of the sensor.

V. Conclusion

In this paper, we presented a novel approach to accurately perform single-scan moving object segmentation in the domain of radar data. Our approach encodes the valuable Doppler velocity information throughout the network and optimizes the upsampling operation by adaptively aggregating information to enhance performance. This allows us to successfully differentiate between moving and non-moving objects, which we evaluated on the RadarScenes dataset. The experiments and the comparisons to other approaches support all claims made in this paper and suggest that our architecture achieves superior performance on moving object segmentation in noisy, single-scan point clouds obtained from automotive radars. The sensors used for recording the RadarScenes dataset are series sensors, already implemented in vehicles, which makes our approach available without additional cost. Overall, our approach outperforms the state-of-the-art methods and proposes a new benchmark for radar-based moving object segmentation, which allows further comparisons with future work, taking a step forward towards reliable single-scan moving object segmentation and sensor redundancy for autonomous vehicles.

REFERENCES

[1] 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.

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[3] M. Berman, A.R. Triki, and M.B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] 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.

[5] 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.

[6] 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.

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2019.

[8] 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.

[9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *Intl. Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[10] 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.

[11] X. Gu, Y. Wang, C. Wu, Y. Lee, and P. Wang. HPLFlowNet: Hierarchical Permutohedral Lattice FlowNet for Scene Flow Estimation on Large-Scale Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] 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.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] D. Hendrycks and K. Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint:1606.08415*, 2016.

[15] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] P. Kaul, D. De Martini, M. Gadd, and P. Newman. RSS-Net: weakly-supervised multi-class semantic segmentation with FMCW radar. In *Proc. of the IEEE Vehicles Symposium (IV)*, 2020.

[17] G. Kim and A. Kim. Remove, then revert: Static point cloud map construction using multiresolution range images. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[18] 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.

[19] 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.

[20] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[21] X. Liu, C.R. Qi, and L.J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[23] 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.

[24] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla. Robust Method for Removing Dynamic Objects from Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[25] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier. Attentional pointnet for 3d-object detection in point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2019.

[26] 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.

[27] 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.

[28] 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.

[29] C. Qi, K. 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.

[30] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens. Stand-alone self-attention in vision models. *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[31] P. Ruchti and W. Burgard. Mapping with dynamic-object probabilities calculated from single 3d range scans. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[32] N. Scheiner, F. Kraus, N. Appenrodt, J. Dickmann, and B. Sick. Object detection for automotive radar point clouds–a comparison. *Proc. of AI Perspectives*, 3, 2021.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] W. Wu, Z. Qi, and L. Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] W. Wu, Z.Y. Wang, Z. Li, W. Liu, and L. Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[43] 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.

[44] 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.

[45] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2020.

[46] 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.

[47] 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.

[48] B. Yang, S. Wang, A. Markham, and N. Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction. *Intl. Journal of Computer Vision (IJCV)*, 128(1):53–73, 2020.

[49] 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.

[50] 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.

[51] 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.

[52] 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.