RadarGNN: Transformation Invariant Graph Neural Network for Radar-based Perception

Felix Fent, Philipp Bauerschmidt and Markus Lienkamp

Technical University of Munich, Germany

School of Engineering and Design

Institute of Automotive Technology

felix.fent@tum.de

Abstract

A reliable perception has to be robust against challenging environmental conditions. Therefore, recent efforts focused on the use of radar sensors in addition to camera and lidar sensors for perception applications. However, the sparsity of radar point clouds and the poor data availability remain challenging for current perception methods. To address these challenges, a novel graph neural network is proposed that does not just use the information of the points themselves but also the relationships between the points. The model is designed to consider both point features and point-pair features, embedded in the edges of the graph. Furthermore, a general approach for achieving transformation invariance is proposed which is robust against unseen scenarios and also counteracts the limited data availability. The transformation invariance is achieved by an invariant data representation rather than an invariant model architecture, making it applicable to other methods. The proposed RadarGNN model outperforms all previous methods on the RadarScenes dataset. In addition, the effects of different invariances on the object detection and semantic segmentation quality are investigated. The code is made available as open-source software under https://github.com/TUMFTM/RadarGNN.

1. Introduction

Autonomous vehicles rely on an accurate representation and understanding of their environment. To achieve this, even under severe weather conditions, the perception has to be robust against changing environmental conditions. However, current perception systems rely mainly on data from camera or light detection and ranging (lidar) sensors, which are negatively affected by certain environmental conditions [38]. The perception capability of both sensor types is for example reduced by fog or rain and camera sensors

(a) Example scene
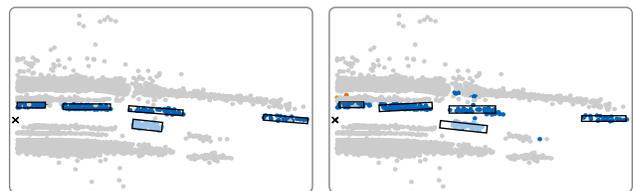
(b) Ground truth   (c) Model prediction

Figure 1. Example scenario a) of the RadarScenes dataset [29] and its corresponding radar point cloud data in the bird's eye view. The annotated ground truth data is shown in b), while the model prediction for object classes and bounding boxes is given in c).

are dependent on an external light source limiting their usability in the dark [38]. As a result of these limitations, research has focused on integrating radio detection and ranging (radar) sensors into perception systems.

Even if radar data is mostly unaffected by adverse environmental conditions [38], the detection quality of radar-based systems cannot yet compete with state-of-the-art image or lidar-based perception methods [6]. While there are several reasons for this discrepancy, the two major challenges of radar-based perception are the limited availability of annotated radar data, and the sparsity of radar point cloud data [26, 35, 39].

Leveraging the sparse information available, a graph neural network (GNN) is proposed that will not just utilize the information encoded in the points but also the relationships between the points. As shown in Fig. 1, an object is

characterized by multiple radar points, which is why the relationship between the points is important to identify and differentiate between objects. In addition, GNNs can operate on unstructured and unordered input data, eliminating the need for data discretization (voxelization) and its associated loss of information [33]. Therefore, all the information of sparse radar point clouds can be used without losing their structural information.

To counteract the limited data availability, a general approach for incorporating invariances into the perception pipeline is proposed. Building upon the success of translation invariant convolution operations, a method is proposed for creating a translation and rotation invariant perception pipeline, leading to better generalization and improved perception quality.

The proposed method was evaluated on the RadarScenes dataset [29] and outperforms all previous methods for bounding box prediction as well as semantic segmentation. In summary, the contributions of this paper are:

- A novel GNN model for radar-based multi-class object detection and semantic segmentation.

- A general approach for transformation invariant object detection and semantic segmentation.

- A new state of the art for object detection and semantic segmentation on the RadarScenes dataset.

2. Related Work

State-of-the-art radar-based object detection methods rely on deep neural networks (DNN) to detect objects within the provided radar point clouds. Even if radar data can also be processed with more conventional methods [25, 28, 31] and object detection can be performed at different data abstraction levels [16, 20, 23], DNNs applied to radar point clouds achieve the best results.

2.1. Radar Datasets

The performance of data-driven perception methods is to a great extend dependent on the underlying dataset. Therefore, the selection of an appropriate dataset is essential for successful model training and meaningful evaluation. However, since most popular perception datasets, such as KITTI [10] or the Waymo Open Dataset [34], do not include annotated radar data, special emphasis is placed on radar-oriented datasets.

Of these, the Dense [3], PixSet [7] and Zendar [17] datasets provide annotated two-dimensional radar data, but the spatial resolution of the deployed radar sensors as well as the extend of the dataset is comparatively small. In contrast, the Oxford Radar RobotCar [1], MulRan [14] and RADIATE [32] datasets utilize high resolution spinning

radar sensors which are not representative of currently deployed automotive radar sensors. The nuScenes [6] dataset includes radar data, but multiple authors [9, 19, 29, 39] have criticized the radar data quality of the nuScenes dataset because of its sparsity, limited feature resolution and errors within the radar domain. In consequence, the RadarScenes [29] dataset is chosen for this work.

The RadarScenes [29] dataset includes point-wise annotated radar data of moving objects assigned to eleven different categories. The dataset comprises the data of four series production automotive radar sensors and contains more than four hours of driving data. The radar points are represented by their spatial coordinates (x, y), target velocities (v_x, v_y), radar cross section (rcs) and a timestamp (t). Currently, most comparative results for radar-based perception are reported based on the RadarScenes dataset, even if it does not provide ground truth bounding boxes and only considers moving objects for ground truth annotations.

2.2. Point Cloud Object Detection

The approaches used to detect objects within point clouds, using deep neural networks, can be divided into three major groups: point-based, grid-based and graph-based methods. In addition to these three general concepts, hybrid methods can be designed by combining different approaches.

Point-based approaches operate on the input point clouds directly, without the need for any preceding data transformations. Therefore, all information and the structural integrity of the point cloud is preserved. Utilizing this method, Schumann *et al.* [27] performed a semantic segmentation on a proprietary radar dataset and later extended their approach to develop an instance segmentation model for radar data [26]. Building upon this, Nobis *et al.* [19] developed a point-based recurrent neural network (RNN) to realize semantic segmentation on nuScenes radar data. Nevertheless, point-based approaches cannot consider individual relationships between points, even if the structure of local groups can be taken into account [33].

Grid-based methods map the point clouds to a structured grid representation by a discretization (voxelization) of the underlying space. Based on this data structure, conventional convolutional neural networks (CNN) can be applied to accomplish different computer vision tasks. Using this approach, Schumann *et al.* [30] developed an autoencoder network to perform a semantic segmentation on radar point clouds, originating from a proprietary dataset. Scheiner *et al.* [26] applied a YOLOv3 [24] detector to a bird's-eye view (BEV) grid representation of radar point clouds and achieved state-of-the-art object detection results on the RadarScenes [29] dataset. However, the preceding data transformation leads to a loss of information and a sparse data representation.

Graph-based methods construct a graph from the input point cloud to operate on and can be categorized into convolutional [15], attentional [37] and message passing [11] neural network types [4]. For graph neural networks, the points are used as nodes within the graph, preserving the structural information of the point cloud, and the relationships between the points are modeled as edges in the graph [5]. This methods was first used by Shi and Rajkumar [33] to implement a GNN for object detection on lidar point clouds. So far, GNNs have only been used once by Svenningsson *et al.* [35] to realize a graph-based object detection on radar point clouds. However, their method was limited to a graph convolutional layer formulation, the effects of invariances where not investigated and their approach was limited to the detection of cars within the nuScenes dataset. Therefore, to the best of our knowledge, GNNs have never previously been used to accomplish a multi-class object detection task on radar point cloud data.

Additionally, hybrid methods can be used to combine several of the above mentioned techniques. Scheiner *et al.* [26] compared multiple different hybrid methods on the RadarScenes [29] dataset and most recently, Palffy *et al.* [21] evaluated a hybrid model architecture on the newly published View-of-Delft [21] dataset. However, most hybrid methods [21, 26, 36] rely on a grid-based approach for the final object detection and are therefore subject to similar disadvantages, as mentioned above.

2.3. Transformation Invariance

The great successes of convolutional neural networks over their predecessors was mainly justified by the translation invariant property of the convolution operation [12, pp. 335–339]. A function is considered translation invariant if its output remains unchanged after a translation of its inputs [5].

Achieving an invariant object detection model is of great interest because the model should be robust to unseen scenarios, where objects can occur in different locations. Moreover, the model should also be applicable to sensors mounted in different positions. To accomplish this property, the following three methods have been used in the literature: data augmentation, invariant data representation and invariant model architectures.

Data augmentation is used to extend the training dataset by modified (e.g. translated or rotated) copies of the original data and relies on the model learning the aspired invariances during the training process [2]. This technique is commonly used to support the generalization of the model and is applied in many of the above mentioned methods [26, 27, 35]. However, this method does not ensure an invariant model after the training process.

Invariant data representation describes the restriction of the input features to those that are invariant to certain transformations [18, p. 23 ff.]. In the Euclidean space, for example, distance is an invariant quantity, which is unaffected by all rigid transformations [18, p. 12]. Only representing the data by such quantities results in an overall invariant data representation. However, limiting the input data to invariant features results in a loss of global context (e.g. absolute vs. relative coordinates).

Invariant model architectures are designed in such a way that the output of the model remains unchanged regardless of a transformation applied to the model input. Such an invariant model architecture was used by Svenningsson *et al.* [35] to design a translation invariant GNN for radar-based object detection. However, incorporating invariances in the model architecture restricts the architectural design to certain (mathematically symmetric) operations [22] and makes the investigation of the effects of different invariances complicated.

3. RadarGNN

This section describes the proposed RadarGNN model for radar-based object detection and semantic segmentation. The object detection pipeline is shown in Fig. 2 and consists of four major steps: data preparation, graph construction, the graph neural network and the detection heads. The main idea behind this method is the achievement of transformation invariances not by using the model architecture itself but by creating an invariant data representation. To achieve this, three things are required: transformation invariant bounding boxes, a graph construction method for invariant input features and a generalization of the GNN layer to consider edge features.

3.1. Data Preparation

The purpose of the data preparation is to create a similar database to that used in previous research and to enable the training of a translation and rotation invariant perception model. This step is important for preserving the comparability of the results and affects not just the model inputs but also the target value determination.

The preparation of the RadarScenes data is based on the implementation of Scheiner *et al.* [26], who serves as a benchmark for this study. Using this approach, the model input is given by an accumulation of radar point clouds within a time period of $500\,\mathrm{ms}$. The resulting point cloud is then cropped to an area of $100\,\mathrm{m}$ times $100\,\mathrm{m}$ in front of the vehicle's rear axis, as shown in [26, Fig. 2]. Since the RadarScenes dataset does not include ground truth bounding boxes, they are created as minimum enclosing rectangles in bird's-eye view, including all points belonging to the same instance. Furthermore, the original eleven instance categories are mapped to five major object classes and finally, the overall dataset is split into training ($64\,\%$), validation ($16\,\%$) and test ($20\,\%$) sets.
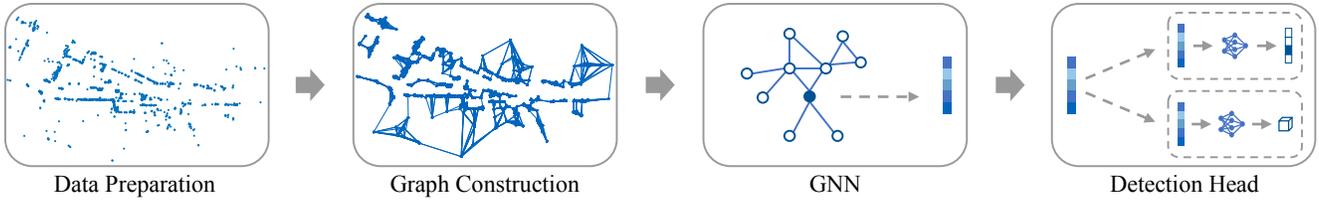
Figure 2. Model overview from point cloud processing on the left, through graph construction and GNN feature extraction, up to the object detection and semantic segmentation on the right.

The investigation of the effects of different invariances on the detection quality of the model requires different bounding box definitions. The existing absolute bounding box definition can be used to train a non-invariant baseline model. An absolute bounding box is defined as a tuple (x, y, w, l, θ) consisting of the box center coordinates x, y, the box dimensions w, l and the yaw angle θ.

The training of a translation invariant model requires a translation invariant bounding box definition. To achieve this, the bounding box is no longer defined by its absolute center coordinates but by the relative position to its associated radar point p_0. Therefore, a translation invariant bounding box is given by a tuple (dx, dy, w, l, θ), with a relative translation dx, dy between the box center and the radar point p_0 it belongs to.

The bounding box definition for the training of a translation and rotation invariant model requires the addition of a second reference point p_{nn}. On this basis, a translation and rotation invariant bounding box is defined as tuple $(d, \varphi, w, l, \theta_{nn})$. Here, the distance d is the distance between the reference point p_0 and the bounding box center. The angle φ represents the angle between the vector from the reference point p_0 to its nearest neighbor p_{nn} and the vector to the bounding box center. Finally, the angle θ_{nn} corresponds to the angle between the directional vector of the bounding box and the vector from the reference point p_0 to its nearest neighbor p_{nn}. A graphical representation of the translation and rotation invariant bounding box definition is given in Fig. 3.

3.2. Graph Construction

The graph construction module transforms the initial radar point cloud into a graph representation and ensures a transformation invariant input data representation. The proposed method comprises of three major steps: the node feature transformation, the edge generation and a matrix transformation.

The node feature transformation maps the original point cloud $\mathcal{P}$, formally defined as finite set $\mathcal{P}$ of $n \in \mathbb{N}$ vectors $p_i \in \mathbb{R}^d$ with $i = 1, ..., n$, to a set of nodes $\mathcal{V} = \{\nu_0, ..., \nu_n | \nu \in \mathbb{R}^{d_\nu}\}$. During this transformation process, the number of points is preserved while the features are transformed. This is important to ensure the invariance of the data representation, while preserving the structure of the original point cloud. The key functionality, however, is the selection of the node features, which determines the invariances of the data representation.

In the non-invariant baseline configuration, the nodes are defined by their absolute spatial coordinates (x, y), their velocity vectors (v_x, v_y), the values of the radar cross section rcs and the associated timestamps t. To achieve a translation invariant data representation, the absolute spatial coordinates are omitted and the encoding of the structural information is subject to the edge generation. Accomplishing a translation and rotation invariant representation further requires the reduction of the velocity information to the Euclidean norm of the velocity vector v. In addition, all nodes hold the information about the connectivity degree c (the number of associated edges). This process shows that the achievement of certain invariances results in a loss of global context, where the edge generation is an attempt at compensating for this.

The edge generation encodes relationships between points in the form of edges and edge features. The set of edges can be formally defined as $\mathcal{E} \subseteq \{(u, \nu) | (u, \nu) \in \mathcal{V}^2, u \neq \nu\}$, where every edge $\varepsilon \in \mathcal{E}$ can be associated with an edge feature vector $e : \mathcal{E} \mapsto \mathbb{R}^{d_\varepsilon}$. The final Graph is then defined by the tuple of nodes and edges $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The edges of the graph are created by a k-nearest neighbors algorithm, where the number of neighbors k is determined empathically and represents a trade-off between performance and computational resources. As a result, every node of the graph is connected to its twenty nearest neighbors.

The edge features are generated in consideration of the desired invariances and with the aim of encoding relationship information between points. However, for the non-invariant baseline configuration, no edge features are generated but all information is contained within the node features. The translation invariant data representation requires the neglection of the absolute coordinates, instead the position relative to the neighboring points dx and dy is encoded in the edge features to preserve the spatial information. To create a rotation and translation invariant data rep-
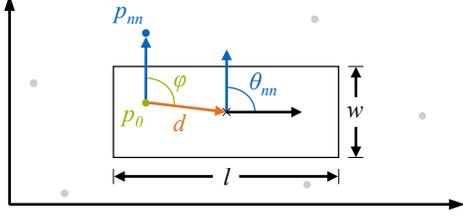
Figure 3. Definition of the translation and rotation invariant bounding box in regard to the radar point p_0 and the reference point p_{nn}. The box is defined by its extend (w, l), position (d, φ) and orientation (θ_{nn}) in the bird's-eye view.

Invariance	Node features	Edge features
-	$x, y, v_x, v_y, rcs, t, \mathrm{c}$	-
Trans.	$v_x, v_y, rcs, t, \mathrm{c}$	dx, dy
Trans. and rot.	v, rcs, t, c	$d, \psi, \gamma_\nu, \gamma_u$

Table 1. Set of node and edge features of the graph for different transformation invariant data representations.

resentation, an enhanced set of point-pair features, inspired by Drost *et al*. [8], is generated. This set consists of the Euclidean distance d between the two points, the relative angle ψ between their velocity vectors and the individual angles between the velocity vector of the points and their connecting line, γ_ν and γ_u. An overview of the various node and edge features for the different data representations is presented in Tab. 1.

The matrix transformation maps the previously generated graph to a more easily processable matrix data representation. Therefore, the graph $\mathcal{G}$ is mapped to a tuple consisting of an adjacency matrix $\mathbf{A}$, a node feature matrix $\mathbf{X}$ and an edge feature matrix $\mathbf{E}$. The resulting tuple $(\mathbf{A}, \mathbf{X}, \mathbf{E})$ represents the actual input of the subsequently defined neural network.

3.3. Graph Neural Network Architecture

The graph neural network consists of two major components and is responsible for the generation of an expressive feature representation for the connected detection heads. The two components are the initial feature embedding and the GNN layers themselves.

The initial feature embedding creates a high-dimensional non-contextual feature representation from the low-dimensional node and edge features. Therefore, a shared multilayer perceptron (MLP) with four layers is used for node feature embedding and one with three layers for edge feature embedding.

A major contribution of this paper is the generalization of the previously proposed graph neural network layer of Svenningsson *et al*. [35], which is based on Shi *et al*. [33].

Their graph convolutional layer function

$$\mathrm{h}_\nu^{l+1} = \zeta(\mathrm{h}_\nu^l, \oplus_{u \in \mathcal{N}_\nu} \xi(x_\nu - x_u, \mathrm{h}_u^l)) + \mathrm{h}_\nu^l, \qquad (1)$$

updates the node features h_ν of node ν by adding the results of the update function ζ to the input node features h_ν^l of the current layer l. The update values are determined by an aggregation $\oplus$ over the node's neighborhood $\mathcal{N}_\nu$ and implemented as a max pooling over the neighbor node features h_u^l weighted by their relative position $x_\nu - x_u$. However, this layer formulation does not allow feature dimension changes across GNN layers and only achieves a translational invariance.

To overcome these limitations we propose a more general message passing neural network (MPNN) layer formulation instead of the graph convolutional (GC) layer formulation of Svenningsson *et al*. [35]. To achieve this, two major changes are made to the update function in Eq. (1). Firstly, the current node features h_u^l are introduced in the embedding function ξ to allow dimensional changes across GNN layers. Secondly, the possibility of using arbitrary edge features is introduced instead of only using the relative position between neighboring nodes. Therefore, the achievement of certain invariances is subject to the edge and node feature generation rather than the layer formulation itself, which allows the analysis of different invariances without changing the network architecture. The resulting update function ζ is given by

$$\mathrm{h}_\nu^{l+1} = \zeta(\mathrm{h}_\nu^l, \oplus_{u \in \mathcal{N}_\nu} \xi(\mathrm{h}_\nu^l, \mathrm{h}_u^l, \mathrm{e}_{\nu,u}^l)) \qquad (2)$$

where the new node features h_ν^{l+1} of node ν after layer l are a function of the original node features h_ν^l and an aggregation $\oplus$ over the neighborhood $\mathcal{N}_\nu$. In this process, the aggregation is conducted over the embedding ξ of the sender node features h_ν^l, the receiver node features h_u^l and the features of the edge $\mathrm{e}_{\nu,u}^l$ connecting them. To obtain a permutation invariant network architecture, the update function ζ and the embedding function ξ are implemented as shared MLPs, and the aggregation function determines the maximum of the embedded features - similar to [22].

The resulting feature space of these layers forms the foundation for the subsequent detection heads to accomplish the desired perception tasks.

3.4. Object Detection and Semantic Segmentation

The proposed RadarGNN model not only performs object detection but also semantic segmentation on the given radar point cloud. This multi-task learning approach is realized by a distinct feature extraction module and individual detection heads, which could also be extended to support additional perception tasks. For our purpose, a semantic

Method	Split	AP_{ped}	AP_{grp}	AP_{tw}	AP_{car}	AP_{trk}	mAP↑
PointPillars [26]	val.	11.2	22.5	38.8	54.2	53.7	36.1
YOLOv3 [26]	val.	**34.4**	55.7	57.4	70.2	61.9	55.9
RadarGNN (ours)	val.	34.0	**58.3**	**66.6**	**72.0**	**70.1**	**60.2**
RadarGNN (ours)	test	33.1	54.8	62.0	70.4	62.0	56.5

Table 2. Object detection results on the RadarScenes dataset for the translation invariant model configuration. The first three rows are evaluated on the validation (val.) set, while the bottom row is evaluated on the test set. The detection quality is given by the average precision (AP) values for the pedestrian (ped), pedestrian group (grp), two-wheeler (tw), car and large vehicle (trk) class with an IOU threshold of 0.3. The mAP represents the mean average precision over all five classes. All benchmark results can be found in [26, Tab. 3].

segmentation and object detection (bounding box prediction) head is used.

The semantic segmentation head consists of a shared MLP, with a final softmax activation function and predicts a confidence score for each class. The final class for every individual point is then determined by the highest confidence score among all classes.

The object detection head is realized by a shared MLP with two consecutive layers and a linear activation function in order not to restrict the output space. The module predicts a bounding box for every point within the given point cloud, which is why a number of suppression schemes have to be applied to receive the final output. Firstly, a background removal is applied to remove all bounding boxes associated with the background class. Secondly, a non-maximum suppression (NMS) is used to remove all overlapping bounding boxes and keep only the one with the highest confidence score. Thirdly, a class-specific threshold is applied to discard all remaining bounding boxes below a certain confidence. Finally, the absolute bounding box representation is restored.

The overall model is trained with a combined loss function, consisting of multiple task-specific loss functions. The semantic segmentation branch uses a class-weighted cross-entropy loss function $\mathcal{L}_{seg}$, whereas the object detection branch utilizes a Huber loss function $\mathcal{L}_{obj}$ [13] with a delta value of one. Additionally, an L2 regularization term $\mathcal{L}_{reg}$ is introduced to prevent the model from overfitting [35]. The overall loss function is given by

$$\mathcal{L} = \alpha \mathcal{L}_{seg} + \beta \mathcal{L}_{obj} + \gamma \mathcal{L}_{reg} \qquad (3)$$

where the weights $\alpha = 1$, $\beta = 0.5$ and $\gamma = 5 \times 10^{-6}$ balance between the different loss terms.

4. Experimental Results

The RadarGNN model is evaluated on the RadarScenes dataset and the obtained results are compared to the current state of the art in radar-based object detection and semantic segmentation. In addition, a detailed analysis of the effects of different invariances on the model's performance is

given. All experiments are conducted on a dedicated benchmark server and within a containerized environment to keep the evaluation environment constant.

4.1. Object Detection

The object detection quality of the RadarGNN model is measured by the mean average precision (mAP) value as defined in [26] and with an intersection over union (IOU) threshold of 0.3. In addition, the class-specific average precision (AP) values are reported for the pedestrian (ped), pedestrian group (grp), two-wheeler (tw), car and large vehicle (trk) classes. The RadarGNN method achieves a mAP value of 60.2% on the RadarScenes validation set, as shown in Tab. 2. The highest AP values are achieved for the two-wheeler, car and large vehicle classes, while the lowest AP value is reported on the pedestrian class. To put that into perspective, the results are compared to the current state of the art on the RadarScenes dataset.

Since all previous results for rotated bounding boxes are only reported on the validation set and the source code of none of the comparison models is publicly available, the RadarGNN model is compared to the literature values accomplished on the validation set. However, the results of our model are also reported on the independent test set in the bottom row of Tab. 2.

The RadarGNN model achieves state-of-the-art results on the RadarScenes dataset and outperforms all previous object detection methods, as shown in Tab. 2. Our graph-based architecture achieves the highest mean average precision (mAP) value and outperforms the hybrid PointPillars as well as the grid-based YOLOv3 method of Scheiner *et al*. [26].

The increased object detection quality has multiple causes, although the utilization of point relationships and the preservation of the structural information are of particular importance. Investigations demonstrated that a higher connectivity (number of edges) results in a better object detection score, but consequently in an increase in computational resources. In addition, the results of Scheiner *et al*. [26] indicate that the detection quality of the grid-based approach is limited in respect of the yaw angle prediction

Method	$F_1 \uparrow$
PointPillars [26]	47.6
YOLOv3 [26]	53.0
LSTM [31]	59.7
PointNet++ [27]	74.3
Recurrent PointNet++ [30]	75.0
RadarGNN (ours)	**77.1**

Table 3. Semantic segmentation results on the RadarScenes test set, given by the macro-averaged F1 score.
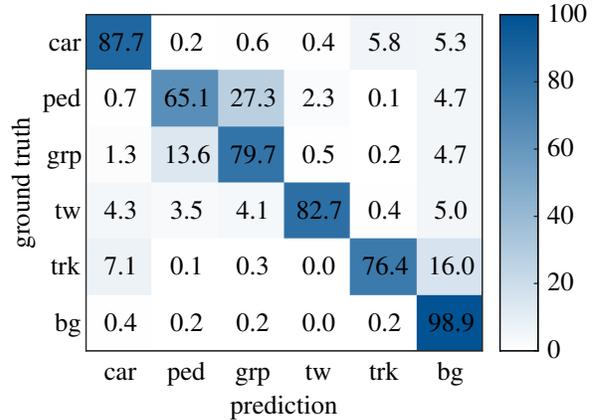

Figure 4. Confusion matrix of the semantic segmentation results on the RadarScenes test set. The matrix represents the ground truth values in contrast to the model prediction for the five objects classes and the background (bg) class.

because of the discretization and associated loss of structural information, which are not observed with the graph-based approach. Nevertheless, the previous YOLOv3 [26] method achieves better results for the pedestrian (ped) class, which is characterized by having very few radar points. Additionally, the introduced invariances greatly influence the detection quality, which is discussed below in Sec. 4.3.

To provide more context to these results, the RadarGNN model is also benchmarked in the official nuScenes detection challenge and achieves a nuScenes detection score (NDS) of 0.059. The proposed method, therefore, outperforms the comparison model of Svenningsson *et al*. [35], which achieved an NDS of 0.034 and has currently the second highest score among all radar-only object detection methods. However, it must be noted that our model was not designed for 3D object detection on the nuScenes dataset but rather for the bird's-eye view (BEV) object detection on the RadarScenes dataset.

4.2. Semantic Segmentation

In addition to the object detection quality, the semantic segmentation quality of the proposed multi-task neural network is evaluated on the RadarScenes dataset and compared to the current state of the art. The segmentation quality is measured by the macro-averaged F1 score, where the RadarGNN model achieves a score of 77.1 % on the RadarScenes test dataset, as shown in Tab. 3.

The proposed method outperforms all previous radar-based multi-task learning approaches and even the dedicated semantic segmentation models of Schumann *et al*. [30]. As shown in Fig. 4, the model is able to differentiate between the different classes, while the highest confusion exists between the pedestrian (ped) and pedestrian group (grp) classes. This result indicates the potential of the graph-based approach for additional computer vision applications on radar point clouds.

On the nuScenes dataset a macro-averaged F1 score of 19.6 % can be achieved on the validation set, which represents the first reported semantic segmentation result with the official nuScenes configuration. Although Nobis *et al*. [19] developed a semantic segmentation method on nuScenes radar data, they used a simplified class configuration and achieved a macro-averaged F1 score of 22.8 %. Using the exact same class configuration as Nobis *et al*. [19], our model achieves a macro-averaged F1 score of 39.4 %.

4.3. Transformation Invariances

The proposed approach to transformation invariance allows a detailed analysis of the effects of different invariances on the perception performance. The conducted study compares the object detection and semantic segmentation quality of three models with a non-invariant, translation invariant as well as translation and rotation invariant configuration.

For the object detection task, the highest mAP value of 56.5 % can be achieved by the translation invariant configuration, as shown in Tab. 4. Hence, the non-invariant as well as the translation and rotation invariant configuration accomplished a lower detection score of 19.4 % and 19.6 %, respectively.

This result could be caused by both the restriction of the input features as well as the differences in the bounding box description (as described in Sec. 3). For this reason, a complementary study, with a translation and rotation invariant bounding box definition for all three configurations, was conducted. The result of this study shows that the more complex bounding box definition leads to an overall lower detection quality but the general trend remains the same. Within this study the non-invariant and translation invariant methods achieved a mAP of 16.4 % and 20.8 %, respectively. Consequently, the input feature restriction can be identified as the root cause of the reduced detection quality. In summary, translation invariance increases the detection quality but the additional rotation invariance, and the accompanying restriction of the input features, negatively affects the detection quality.

Invariance	mAP	F_1
-	19.4	68.1
Translation	**56.5**	**77.1**
Translation and rotation	19.6	76.5

Table 4. Object detection and semantic segmentation results for different invariances on the RadarScenes test set, given by the mean average precision (mAP) and macro-averaged F1 score.

Similar to the object detection task, the highest semantic segmentation score of 77.1 % can be achieved by the translation invariant configuration, as shown in Tab. 4. The non-invariant configuration achieved a macro-averaged F1 score of 68.2 % and the translation and rotation invariant configuration achieved 76.5 %, respectively.

The smaller differences between the semantic segmentation results, in comparison to the object detection results, can be explained by two reasons. Firstly, the bounding box description has no influence on the results, since semantic segmentation requires no bounding boxes. Secondly, directional information (which is lost with the addition of rotational invariance) is less relevant for point-wise classification than it is for the prediction of the bounding box orientation.

However, the object detection and semantic segmentation tasks are not independent but coupled by a combined loss function Eq. (3) and model training. Therefore, the segmentation quality is affected by the object detection performance and vice versa. Since the differences between the segmentation results are small, we conducted a complementary study to further analyze the effects of invariances on the segmentation quality. Within this study, the segmentation branch was trained independently by setting $\beta = 0$. As a result, the non-invariant model achieved an F1 score of 60.5 %, the translation invariant model achieved 66.5 % and the translation and rotation invariant configuration achieved 68.2 %. Consequently, the assumption is made that further invariances increase the segmentation quality and justify the necessary restriction of the input features.

To provide evidence to the claim that the introduction of transformation invariances counteract the effects of limited data availability, we studied the model performance during a reduction of the training data. For this experiment we gradually reduced the amount of training data sequences, but kept the test set constant and monitored the model performance for the three different levels of invariance. The result shows that transformation invariant models are less affected by a limited data availability and the addition of further invariances contribute positively to this effect as shown in Fig. 5.

As a result, the analysis indicates that certain perception tasks benefit differently from specific transformation
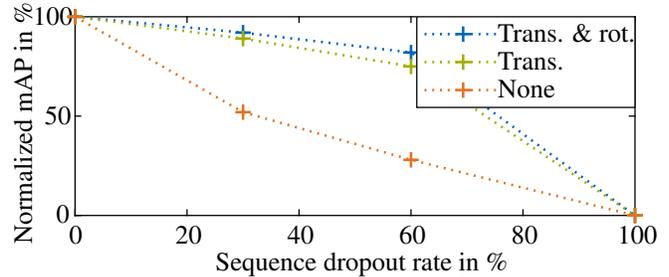

Figure 5. Performance of the RadarGNN model for different invariance levels over the number of training sequences. The performance is normalized to the baseline performance.

invariances. Where semantic segmentation improves with the addition of further invariances, the highest object detection score can be achieved by the translation invariant configuration. Furthermore, additional transformation invariances improve the ability of the model to handle limited data availability.

5. Conclusion

In this paper, we present a graph neural network for both multi-class object detection and semantic segmentation on radar point cloud data. The proposed RadarGNN model uses a generalized message passing neural network layer to consider edge features within its update function and to allow dimensional changes in the GNN. Furthermore, a more generalized approach to achieve transformation invariance is proposed by the creation of an invariant data representation rather than an invariant model architecture. This modification allows the analysis of different invariances without changing the model architecture itself and is transferable to different applications. However, since an invariant data representation always involves a restriction of the input features, a distinct set of point-pair features is proposed to compensate for this during the edge feature generation. The proposed RadarGNN model achieves state-of-the-art results on the RadarScenes dataset for both radar-based object detection and semantic segmentation. In addition, the effects of different invariances on the object detection and semantic segmentation quality is investigated. The incorporation of a sensor fusion concept or the transfer to different sensor modalities is subject to future research.

References

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020. 2

[2] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from train-

ing data. *Advances in neural information processing systems (NeurIPS)*, 33:17605–17616, 2020. 3

[3] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11679–11689, Los Alamitos, California, 2020. IEEE. 2

[4] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. 3

[5] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 3

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11618–11628, Los Alamitos, California, 2020. IEEE. 1, 2

[7] Jean-Luc Deziel, Pierre Merriaux, Francis Tremblay, Dave Lessard, Dominique Plourde, Julien Stanguennec, Pierre Goulet, and Pierre Olivier. Pixset: An opportunity for 3d computer vision to go beyond point clouds with a full-waveform lidar dataset. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2987–2993. IEEE, 2021. 2

[8] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005, San Francisco, CA, 2010. IEEE. 5

[9] Florian Engels, Philipp Heidenreich, Markus Wintermantel, Lukas Stacker, Muhammed Al Kadi, and Abdelhak M. Zoubir. Automotive radar signal processing: Research directions and practical challenges. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):865–878, 2021. 2

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, Providence, RI, 2012. IEEE. 2

[11] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1263–1272, Sydney, NSW. PMLR. 3

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. 3

[13] Peter J. Huber. Robust estimation of a location parameter. In *Breakthroughs in Statistics*, pages 492–518. Springer, New York, NY, 1992. 6

[14] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6246–6253. IEEE, 2020. 2

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 3

[16] Michael Meyer, Georg Kuschk, and Sven Tomforde. Graph convolutional networks for 3d object detection on radar data. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3053–3062. IEEE, 2021. 2

[17] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High resolution radar dataset for semi-supervised learning of dynamic objects. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 450–457, Seattle, WA, 2020. IEEE. 2

[18] Joseph L. Mundy and Andrew Zisserman. *Geometric invariance in computer vision*. MIT Press, Cambridge, Mass., 1992. 3

[19] Felix Nobis, Felix Fent, Johannes Betz, and Markus Lienkamp. Kernel point convolution lstm networks for radar point cloud segmentation. *Applied Sciences*, 11(6):2599, 2021. 2, 7

[20] Andras Palffy, Jiaao Dong, Julian F. P. Kooij, and Dariu M. Gavrila. Cnn based road user detection using the 3d radar cube. *IEEE Robotics and Automation Letters*, 5(2):1263–1270, 2020. 2

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

[22] Charles R. Qi, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, Honolulu, HI, 2017. IEEE. 3, 5

[23] Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Perez. Raw high-definition radar for multi-task learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17000–17009, New Orleans, LA. IEEE. 2

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 2

[25] Nicolas Scheiner, Nils Appenrodt, Jurgen Dickmann, and Bernhard Sick. A multi-stage clustering framework for automotive radar data. In *The 2019 IEEE Intelligent Transportation Systems Conference - ITSC*, pages 2060–2067, Piscataway, NJ, 2019. IEEE. 2

[26] Nicolas Scheiner, Florian Kraus, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Object detection for automotive radar point clouds – a comparison. *AI Perspectives*, 3(1), 2021. 1, 2, 3, 6, 7

[27] Ole Schumann, Markus Hahn, Jurgen Dickmann, and Christian Wohler. Semantic segmentation on radar point clouds. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2179–2186, Piscataway, NJ, 2018. IEEE. 2, 3, 7

[28] Ole Schumann, Markus Hahn, Jurgen Dickmann, and Christian Wohler. Supervised clustering for radar applications: On

the way to radar instance segmentation. In *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, Piscataway, NJ, 2018. IEEE. 2

[29] Ole Schumann, Markus Hahn, Nicolas Scheiner, Fabio Weishaupt, Julius F. Tilly, Jurgen Dickmann, and Christian Wohler. Radarscenes: A real-world radar point cloud data set for automotive applications. In *Proceedings of 2021 24th International Conference on Information Fusion (FUSION)*, pages 1–8, Piscataway, NJ, 2021. IEEE. 1, 2, 3

[30] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wohler, and Jurgen Dickmann. Scene understanding with automotive radar. *IEEE Transactions on Intelligent Vehicles*, 5(2):188–203, 2020. 2, 7

[31] Ole Schumann, Christian Wohler, Markus Hahn, and Jurgen Dickmann. Comparison of random forest and long short-term memory network performances in classification tasks using radar. In *2017 Symposium on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, Piscataway, NJ, 2017. IEEE. 2, 7

[32] Marcel Sheeny, Emanuele de Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception in bad weather. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2021. 2

[33] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1708–1716, Los Alamitos, California, 2020. IEEE. 2, 3, 5

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2451, Los Alamitos, California, 2020. IEEE. 2

[35] Peter Svenningsson, Francesco Fioranelli, and Alexander Yarovoy. Radar-pointgnn: Graph based object recognition for unstructured radar point-cloud data. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–6. IEEE, 2021. 1, 3, 5, 6, 7

[36] Julius F. Tilly, Stefan Haag, Ole Schumann, Fabio Weishaupt, Bharanidhar Duraisamy, Jurgen Dickmann, and Martin Fritzsche. Detection and tracking on automotive radar data with deep learning. In *Proceedings of 2020 23rd International Conference on Information Fusion (FUSION 2020)*, pages 1–7, Piscataway, NJ, 2020. IEEE. 3

[37] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio. Graph attention networks. 3

[38] Keisuke Yoneda, Naoki Suganuma, Ryo Yanase, and Mohammad Aldibaja. Automated driving recognition technologies for adverse weather conditions. *IATSS Research*, 43(4):253–262, 2019. 1

[39] Yi Zhou, Lulu Liu, Haocheng Zhao, Miguel López-Benítez, Limin Yu, and Yutao Yue. Towards deep radar perception for autonomous driving: Datasets, methods, and challenges. *Sensors*, 22(11), 2022. 1, 2