Graph Convolutional Networks for 3D Object Detection on Radar Data

Michael Meyer
Cruise
michael.meyer@getcruise.com

Georg Kuschk
Cruise
georg.kuschk@getcruise.com

Sven Tomforde
Kiel University
st@informatik.uni-kiel.de

Abstract

Despite its advantages as an inexpensive, weather-robust and long-range sensor which additionally provides velocity information, radar sensors still lead a shadowy existence compared to lidar and camera when it comes to fulfilling the requirements of fully autonomous driving. In this work, we focus on fully leveraging raw radar tensor data instead of building up on human-biased point clouds which are the typical result of traditional radar signal processing techniques. Utilizing a graph neural network on the raw radar tensor we gain a significant improvement of $+10\%$ in average precision over a grid-based convolutional baseline network. The performance of both networks is evaluated on a real world dataset with dense city traffic scenarios, diverse object orientations and distances as well as occlusions up to visually fully occluded objects. Our proposed network increases the maximum range for state-of-the-art full-3D object detection on radar data from previously 20m to 100m.

1. Introduction

One of the main tasks that autonomous vehicles need to perform is the perception of their surroundings. Cameras, lidars, and radars are common sensors used to perform a variety of different sub tasks, including free space detection [33], scene classification [27], and detection of other vehicles [20] and pedestrians [22].

The perception needs to be reliable and robust under all weather conditions in order to guarantee safety for passengers and other road users. One way to improve perception reliability is to perform the same task with various different sensors.

Since the publication of the KITTI dataset [13] a lot of research has focused on cameras and lidars, but more recently there have been also an increasing number of publications about perception algorithms for radar data. Compared to camera and lidar, radar data is more invariant to weather changes making it an important sensor for enabling autonomous driving under severe weather conditions, such as heavy rain or snow. Additionally, radar sensors are able to capture velocity information instantly and they have a far measurement range [39]. Therefore, using radar data for object detection additionally, can improve both redundancy and robustness in the overall autonomous driving stack.

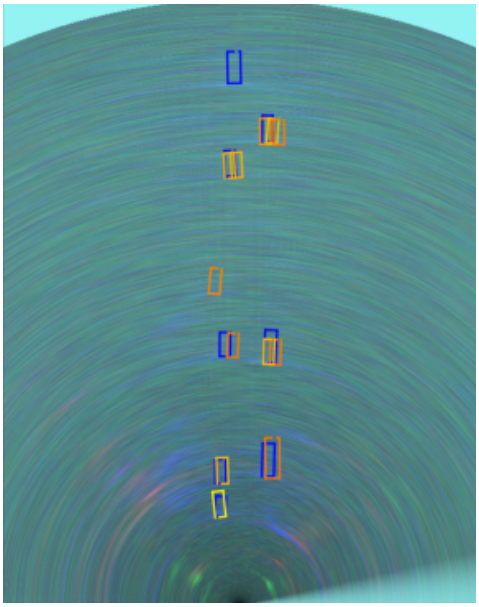


Figure 1: Detections of one test frame overlayed on the radar tensor data. Here the detections are only visualized in 2D but they are actually 3D boxes. Blue rectangles are ground truth and predictions are visualized in a shade of yellow to red depending on the score, yellow being highest. Note that the maximum range corresponds to 102 m.

In this work we investigate if graph neural networks (see Subsection 3.1) are beneficial for 3D object detection on radar data. The two main hypotheses that are investigated in this paper are the following:

1. Using the information of the Cartesian distances between points of data in the radar signal can be used in graph neural networks to improve the performance of object detection tasks on radar data.

2. Isotropic graph convolutional networks (see Subsection 3.1) are beneficial for radar data, because the radar signal often is not located in one cell/pixel, but it usually fades into neighboring cells.

Our main contributions are as follows:

- We present a network for 3D object detection on radar data only, which reaches state-of-the-art performance, while being evaluated on a much more complex dataset.
- We are the first to ever evaluate 3D object detection on radar data with a distance above 50m and up to 100m.
- Using graph neural networks on radar data, we were able to boost the object detection performance by 10% showing that it is a suitable method for aggregating information in low-level radar data.

The remainder of this paper is organized as follows: Section 2 gives an overview of related work. Section 3 briefly introduces graph neural networks and FMCW radar processing. Afterwards, in Section 4 the dataset, networks and evaluation metrics used in this work are described. Section 5 presents the results of the object detection experiments, which are discussed in Section 6. Finally, 7 summarizes this article and gives an outlook on future work.

2. Related Work

Due to the seminal publication of the KITTI dataset [13] 3D object detection in automotive context got a huge boost, leading to thorough investigation on state-of-the-art methods for lidar-based object detection, resulting in mature approaches such as [20, 36, 47] which were especially enabled by the breakthrough of voxel-based as well as point cloud based feature extraction backbones in deep neural networks [30, 31].

Camera-only approaches for 3D object detection have gained attraction recently [8, 37, 46], but are falling short to lidar based approaches in terms of distance accuracy.

The obvious fusion of lidar and camera data combines the geometrical precision of lidar together with dense visual cues rich of context and is heavily investigated as well [19, 29, 38, 41].

When it comes to adverse weather conditions, long-range detection and measuring velocities, camera and lidar sensors are at a disadvantage and it is up to radar sensors to fill this perception gap. Early work on applying deep learning techniques to radar data for better generalizing perception algorithms was done in the area of object detection based on occupancy grid maps [24], radar point cloud segmentation [34], radar point cloud based object detection in conjunction with camera data [26] or without [2].

Radar point cloud data however is very challenging as even with the latest high-resolution radar sensors the density is by a factor of 10 to 100 lower / sparser than current lidar systems.

To that end, this work among others is investigating the underlying raw data instead of the point cloud, which itself is a result of classical radar signal processing on the raw data. This raw radar data (radar tensors or radar spectra) usually form a 2D (range, velocity) or 3D (range, azimuth, velocity) regular grid of complex-valued or real-valued energy reflection responses.

2.1. Radar Object Detection

State-of-the-art work in the area of regressing 2D or 3D objects (including the object orientation and dimension) on radar tensors is rather scarce so far:

The authors of [4] are using a Faster R-CNN [32] style network to estimate the 3D location (distance and azimuth/elevation angle) of a real-world single point target (corner reflector) per scene (naturally excluding dimension and orientation).

The work of [25] collapsed the 3D radar tensor (range, azimuth, velocity) in each dimension separately and runs CNN-based feature extraction on each of the three resulting 2D inputs, concatenating the resulting feature maps and ingesting them in a 2D object detection head.

In [9] the authors employ a ResNet-style backbone ([15]), upsampling layers and anchor-based proposals per spatial grid cell. This is closely related to our baseline network architecture (see Section 4.1).
In terms of combining radar tensor data with camera data at an early stage and feeding it into one object detection network research was done as well by [21], [14] and [17] investigating different ways to fuse the rather different data representations.

2.2. Graph Neural Networks for Object Detection

Even though graph neural networks (GNNs) are a relatively new direction in research, they have been rapidly adopted for object detection. In [11] spatial relationships between 3D proposals are used in a graph in order to consider the whole scene structure for the final box predictions. Similarly, the authors of [7] use relative positions of the proposals embedded in a graph to form an attention map. Furthermore, they use a U-Net based on graph convolutions to aggregate features.

In [45] lidar point clouds are discretized into pillars and then graphs are constructed from k-nearest neighbor pillars. Even though graph neural networks have been surpassing state-of-the-art results in many areas, they have never been applied to radar data yet.

2.3. Datasets

There are a few publicly available radar datasets [44, 3, 28, 6], but none containing raw radar data with accurately labeled ground truth objects. Thus, we had to create such a dataset as described in Section 4.2.

3. Theory

3.1. Graph Neural Networks

Graph Neural Networks generalize the concept of standard convolutional neural networks to non-Euclidean domains [5]. The input for GNNs are graphs consisting of nodes and edges, where edges represent the relationship between nodes. In standard CNNs this relationship can only be used as features of nodes, whereas in GNNs it can be used to propagate information through the graph guided by the edge information. Essentially, a GNN leverages the graph connectivity to learn relationships between nodes. Through an iterative process that depends on the graph structure, the GNN transforms the input node features and edge features into output feature vectors. These output features are invariant to the input order of nodes [1].

There are many possible ways to define convolution layers on graphs [43]. In this paper the Graph Convolutional Networks (GCN) operator of Kipf and Welling will be used [18]. The C dimensional node features of N nodes in the graph can be summarized in a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times C}$ which is propagated through a GCN layer with a layer-specific trainable weight matrix $\mathbf{\Theta} \in \mathbb{R}^{C \times C'}$ by the following rule:

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X} \mathbf{\Theta} \tag{1}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix $\mathbf{A}$ with inserted self-connections and

$$\hat{D}_{ii} = \sum_j \hat{A}_{ij} \tag{2}$$

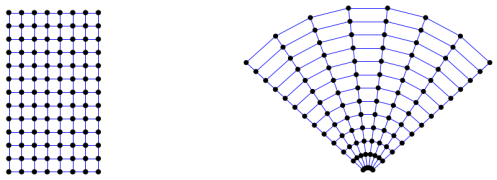
is its diagonal degree matrix. The node-wise formulation for the above calculation is given by:

$$\mathbf{x}'_i = \left(\sum_j \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j \right) \mathbf{\Theta} \tag{3}$$

$$\hat{d}_i = \sum_j e_{j,i} \tag{4}$$

where $e_{j,i}$ denotes the edge weight from source node i to target node j and $\mathbf{x}_i$ is the i-th row vector in $\mathbf{X}$.

Originally, the GCN convolution was used for node classification and its authors argued that it is especially useful when the adjacency matrix contains information not present in the data [18]. That this data can be used as edge weights is the one of the main differences between the GCN operator and standard 2D convolutions.



(a) Illustration how radar data is represented in polar tensor form.

(b) Polar representation of range/azimuth angle transformed to Cartesian space.

Figure 2: Illustration of radar data. Each black circle represents a specific range-azimuth pair and has a 256-dimensional feature vector representing various radial velocities.

Another major difference can be understood by analyzing Equation 1. The first part of the equation

$$\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X} \tag{5}$$

corresponds to a weighted sum of feature vectors of the previous layer and the result is a $N \times C$ matrix, which is then multiplied with the trainable $C \times C'$ weight matrix $\mathbf{\Theta}$. The weighting of the feature vectors is done through the edges and edge weights. For one/each output feature channel there is exactly one weight for each input feature channel. This *isotropic* property is a substantial difference to 2D convolutions, where there is one trainable weight per pixel (or in this case connected node) per input feature channel. In other words, isotropic GCNs treat every "edge direction" equally [10], whereas standard 2D convolutions differentiate between left/right/up/down pixel.

3.2. FMCW Radar

In this work a frequency modulated continuous wave (FMCW) radar is used. FMCW radars emit a signal with a linear modulated frequency and record the signal that is reflected from objects in the scene. The received signal is mixed with the transmitted signal to obtain the frequency difference between the two. This beat frequency is a function of the range of the objects. Multiple receiver channels are used to infer azimuth angles (beams) of objects through digital beamforming. The signal pulse is repeated multiple times and from the phase shift between adjacent pulses, caused by radial motion occurring between pulses within a range resolution cell, can be used to compute the Doppler radial velocity in that cell. So, many radar sensors use a data tensor with dimensions range gates, beams, and Doppler channels to detect interesting objects. In classical radar processing, a constant false alarm rate (CFAR) algorithm is often used to extract a point cloud from this tensor.

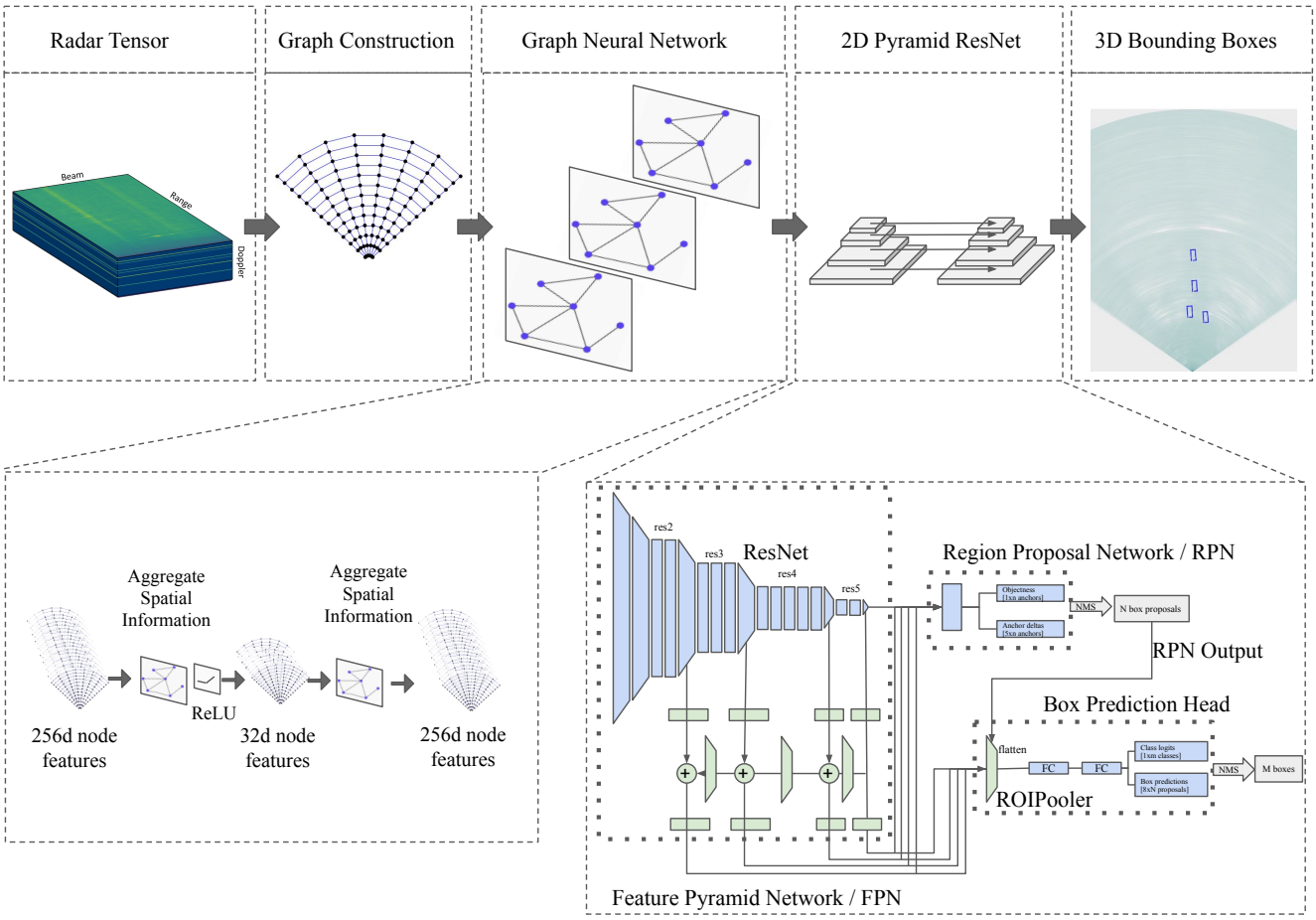


Figure 3: Overview of our network architecture

For a more thorough introduction to radar fundamentals and classical signal processing we refer to [39, 40].

Due to the digital beamforming radar sensors (like lidar sensors) measure the data in polar coordinates with a specific resolution in range and azimuth (see Figure 2b). Therefore, the signal of objects covering multiple bins in azimuth varies with range, because an object covers more and more azimuth bins the closer it is to the sensor. Recent publications performing object detection on range-beam-Doppler tensors use bilinear interpolation to transform the polar data into Cartesian space within the network [25, 9].

4. Methods and Experiments

In this work, we train three variants of a network performing 3D object detection on radar data. The baseline variant does not contain any graph convolutions whereas the other two variants contain graph convolution layers. The 3D box predictions of these networks (x, y, z, l, w, h, yaw) are evaluated both in 3D and in projected birds-eye-view (2D).

4.1. Network architecture

Almost all neural networks performing object detection on radar data act on the data in a birds-eye-view representation, which means that ideally the data should be invariant to shifts in range and azimuth. This is clearly not the case for polar radar data. One approach is to train the network with more data, so it learns to use different filters for different ranges. Even if this works, it has the shortcoming that the network needs to first implicitly infer the range and then use the correct filter to detect objects, making the task for the neural network much harder. This approach could be compared to detecting objects in camera images at different scales, without a specific mechanism. In most cases, it seems favorable to align variations in the data, rather than training the network to learn it implicitly. Feature pyramid networks [23] for example, introduce a mechanism to detect objects at different scales and spatial transformers [16] tackle the problem of aligning variations of data caused by rotations.

Transforming the polar data to the Cartesian space (through e.g. bilinear interpolation) would make objects in the data more or less invariant to shifts in range and beam dimension. However, potentially a lot of information is lost in this transformation and it has been shown that transforming the range-beam-Doppler tensor into Cartesian space at the beginning of the network leads to a significant decrease in object detection performance [25].

If, in contrast, the first layers in the network act on the polar radar data, the information about the various different Cartesian distances between the cells for different ranges is not available to the network (see Figure 2). To feed this information into the network without using bilinear information, graph convolutions are utilized. In this work, the performance of three different networks is evaluated.

As a baseline for the experiments, we use a network based on a pyramid ResNet [15], which uses the polar range-beam-Doppler tensor as input and after two convolutional layers converts the feature maps to Cartesian space through bilinear interpolation. This baseline network is referred to as Radar Tensor Network (*RT-Net*).

The RT-Net is compared to a network where the layers before the polar-Cartesian transformation are replaced with graph convolutions (see Fig. 3). The graph is constructed from the radar tensor in the following manner: each range-beam cell forms a node with a C-dimensional node feature, where C corresponds to the number of Doppler channels. Each node has edges to other nodes which are either

- in the same beam and in neighboring range cells, or
- in the same range gate and in neighboring beams.

For each edge the Euclidean distance r is determined between connected nodes i, j in Cartesian space and serves as parameter for the edge weight $e_{j,i}$:

$$e_{j,i} = \frac{1}{1+r} \tag{6}$$

This graph is used as input to two consecutive GCN convolution layers (implemented in [12]) before the output is transformed back into a three-dimensional tensor and processed further with the pyramid ResNet. This whole network is called Graph Tensor Radar Network (GTR-Net) and the comparison between GTR-Net and RT-Net is used to evaluate how suitable graph convolutions are in aggregating information and encoding features from radar tensor data.

The influence of the edge weights is evaluated by training and evaluating the GTR-Net twice - once with all edge weights set to 1, and once with edge weights calculated from the Euclidean distance in Cartesian space.

Each of the three networks is trained until the validation loss was increasing and the average precision on the validation dataset was degrading.

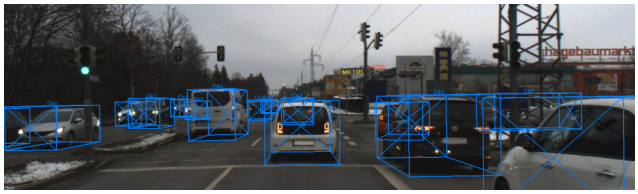


Figure 4: Exemplary street scene (camera image) corresponding to a frame of a range-beam-Doppler tensor which was used for training. Note that camera data was not used for object detection and that data annotations also contain (visually) fully occluded objects.

Number of scenes	25
Number of frames	2010
Number of objects	16169
Median number of objs/frame	7.0 (min/max = 1..41)
Occlusion level (none, partial, fully)	34.8% / 58.2% / 7.0%
Median distance of objects	33.7m (max = 102m)

Table 1: Dataset properties

4.2. Dataset

Due to a large domain gap in terms of different sensor characteristics between radar sensors and different radar data levels, we could not build upon existing radar datasets like [3, 6, 28, 44], because either the raw radar data or accurately labeled ground truth data was missing. To create a dataset we equipped a test vehicle with a radar (Astyx HiRes), lidar (Ouster OS-1) and camera (Point Grey Blackfly), placing them in front-looking direction and maximizing the overlap of the commonly observed area. Extrinsic calibration and timestamp synchronization was taken care of to have accurately calibrated 6DoF poses (error $\leq$ 0.2 deg) and temporally synchronized multi-sensor data (error $\leq$ 5ms).

Camera and lidar data were used for manual annotation of 3D ground truth objects (cars). In case of fully occluded objects (not visible by lidar nor camera) but visible in radar data via multi-path propagation, the objects properties determined in previous frames were associated with the location determined by the radar measurements. We downsampled the synchronized multi-sensor data to 0.25 Hz to reduce temporal correlation between adjacent frames.

The data was captured in inner city complex traffic and junction scenarios, trying to balance the orientation distribution of the objects to not have a highway-style biased dataset.

For training and evaluating our networks we use a fix $0.7/0.15/0.15$ train/val/test split.

Difficulty level	Object properties
Easy	0..40m and occ $\leq$ none
Moderate	0..70m and occ $\leq$ partial
Hard	0..102m and occ $\leq$ full

Table 2: Evaluation categories used

Further statistics of our dataset are listed in Table 1 and an exemplary street scene is depicted in Figure 4.

4.3. Evaluation Metrics

To measure the detection performance we provide the average precision (AP) for each experiment, denoting the integral of the precision-recall (PR) curve for varying thresholds of the networks score values.
To have some further high-level insights we further define three different difficulty categories of objects according to Table 2.

Related work in radar-based object detection [2, 9, 25] typically use an IoU threshold of either 0.3 or 0.5 evaluated on either 2D groundplane projection or in 3D.
In this work for the 3D evaluation an IoU threshold of 0.3 is used. Assuming that each dimension is contributing equally to the IoU, this threshold nearly corresponds to a threshold of 0.5 in 2D, because $0.3^{2/3} \approx 0.5$. Nevertheless, the IoU threshold has a great influence on the AP and different down-stream tasks might have different requirements on how accurate the detections need to overlap with objects. Hence, the AP is also evaluated for a range of different IoU thresholds.

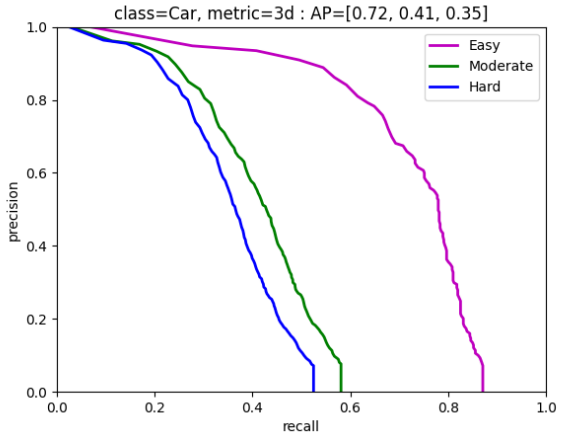
5. Results

The average precision values for the test dataset are displayed in Table 3 and show a significant performance improvement of the graph convolution network over the baseline network. However, there is only a marginal difference in performance between using edge weights based on Cartesian distances or using identical edge weights for all edges. The best performing network is GTR-Net with identical edge weights for all edges. Compared to the baseline, the performance of this network improved by $(+12.3\%, +10.7\%, +9.8\%)$ for the difficulties *Easy*, *Moderate*, *Hard*, respectively. The GTR-Net with edge weights computed from Cartesian distances between the nodes showed an improvement of $(+11.5\%, +9.8\%, +8.9)\%$ compared to the RT-Net.

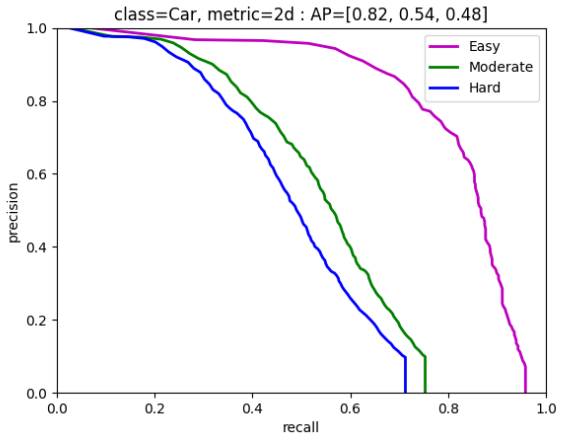
The best performing network - GTR-Net with identical edge weights - is evaluated both in 2D and 3D. Precision-recall curves for the 3D and 2D evaluation are given in Figure 5a and Figure 5b and we further evaluate the AP over varying IoU thresholds in Figure 6a and Figure 6b.

Method	Easy	Moderate	Hard
RT-Net (baseline)	59.3%	30.0%	25.6%
GTR-Net with all identical edge weights	71.6%	40.7%	35.4%
GTR-Net with Cartesian edge weights	70.8%	39.8%	34.5%

Table 3: Average precision values of our baseline method using regular spatially gridded convolutions and graph neural network (for 3D evaluation and IoU threshold of 0.3).



(a) 3D evaluation



(b) 2D evaluation

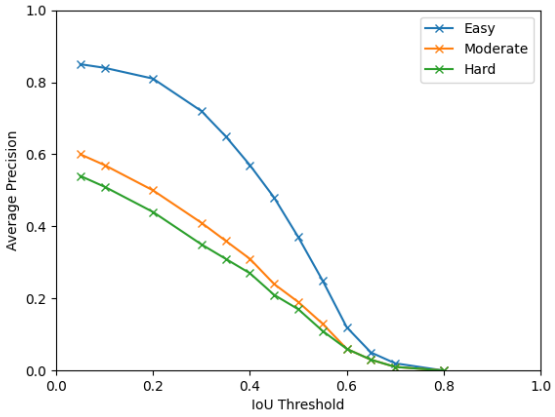
Figure 5: PR curve for the GTR-Net with identical edge weights and IoU threshold of 0.3.

5.1. Comparison with State-of-the-Art

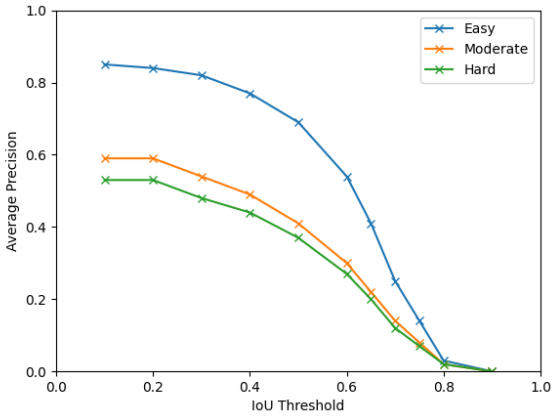
Considering how many researchers work in the area of autonomous driving, surprisingly little research has focused

Publication	Data	2D/3D	Scenarios	Max. Range [m]	IoU	AP [%]
RAD [25]	radar tensor	2D	Highway	46.8	0.5	86.8 ± 0.3
Dong *et al.* [9]	radar tensor	2D	Highway	30	0.3	77.3
Pointillism [2]	radar pointcloud	3D	All	20	0.5	67
GTR-Net (**Ours**)	radar tensor	2D	All	40 70 102	0.5	69.3 41.4 37.1
GTR-Net (**Ours**)	radar tensor	3D	All	40 70 102	0.3	71.6 40.7 35.4

Table 4: Comparison with state-of-the-art radar-based object detection methods. Please note the varying evaluation parameters and the underlying different datasets.



(a) 3D evaluation



(b) 2D evaluation

Figure 6: AP over IoU threshold

on 3D object detection on radar data. There have been a few publications about 2D birds-eye-view object detection on radar tensor data (see Table 4), especially noteworthy in this context is the work of Major *et al.* [25].

The average precision values of these publications are not directly comparable with the results of this work, because the datasets differ and the datasets are not publicly available, so it was impossible to evaluate our network on their data. Re-implementation of these non open source methods from scratch and evaluating them on our dataset was considered, but dismissed, because different code maturity levels would also prevent a fair comparison.

Nevertheless, each of these publications had one or multiple shortcomings. Particularly, all publications only evaluated their performance with a maximum range below 50m. Some other deficiencies are:

- RAD [25] was trained (and evaluated) only on highway data in 2D. Object detection on highway data is much easier, because objects do not have a high variance in position and orientation.
- Dong *et al.* [9] only evaluated their 2D detection performance on highway data up to 30m.
- Pointillism [2] was trained with ground truth, which was labeled with 16-layer lidar data only and evaluate only up to a maximum distance of 20m.

Other publications in the field of radar-only object detection often use non-standardized evaluation metrics. Wang *et al.* [42], for example, only use cameras for generating ground truth and, therefore, cannot use the standard IoU based AP metric and define their own metric.

This paper is the first to perform 3D object detection on a range above 20 meters. At the same time, the GTR-Net is trained and evaluated on difficult traffic scenes instead of highway-only data. All in all, it is the first work demon-

strating that radar can potentially become (one of) the main sensor(s) for perception in autonomous vehicles.

6. Discussion

6.1. 3D Object Detection on Radar Data

Undeniably, the performance of 3D object detection on lidar data is still superior to radar in close range and good weather conditions. For example, in [35] the authors achieve a performance of (90.3%, 81.4%, 76.8%) with lidar-only for the KITTI benchmark dataset, categories (*Easy*, *Moderate*, *Hard*). Nevertheless, recent advances in using deep neural networks on radar data show that there is great potential to mature radar sensors as an L4/L5 automotive sensor, complementing lidar and camera sensors.

Prior works showed that object detection in 2D can be successfully performed on radar data [25]. The evaluation of object detection in 3D presented in this work (GTR-Net) with an IoU threshold of 0.3 yields similar AP values to the evaluation in 2D with an IoU threshold of 0.5 (see Tab. 4). This is an indicator that the network did indeed learn to detect objects in the z-dimension (at least with similar detection capabilities as in the other two dimensions). This shows that radar sensors are in fact capable of 3D object detection - important for non-planar road surface environments (hilly streets and highway ramps).

6.2. Graph Neural Networks

The experiment of the GTR-Net with edge weightings based on Cartesian distances does not perform better than the one with identical edge weights for all nodes. This implies that increasing the weights of spatially close nodes, which are primarily nodes from neighboring range gates, does not help to improve the performance. This would be expected if nodes with equal distances in neighboring beams are more important than nodes in neighboring range gates. Arguably the performance could be improved further by using two dimensional edge attributes and differentiating between a Cartesian distance in range- and beam- direction. Thereby the nodes in neighboring beams could be weighted differently than nodes in neighboring ranges. Investigating if this leads to an improvement could answer the question if one direction is more important for information aggregation. Further work is needed into this direction.

The question remains why graph convolutional neural networks perform so much better than regular CNNs when the distance information of the data is not even included. This question could be answered by the difference of a standard 2D convolution layer and a graph convolution. The isotropic graph convolution used in this work aggregates information in one node by weighting the features of connected nodes via edge weights but only using one trainable weight per feature channel for all connected nodes.

This difference could explain why the GTR-Net was much better at aggregating relevant information leading to a much better performance. Due to the nature of radar signals and the resulting non-ideal point spread function, the signal is not localized exactly in one range-beam pixel, but it might be affecting neighboring pixels, too. The isotropic GCN seems to be better at aggregating information under these circumstances.

7. Conclusion

Object detection is one of the most crucial tasks in autonomous driving to further predict trajectories of non-static objects. Based on lidar data it, already gives good results but to design fail-safe systems it is desirable to be able to perform this task on other sensor modalities also, e.g. radar data. Therefore, in this work we have presented a network for 3D object detection purely on radar data. Compare to current state-of-the-art, the presented object detection on radar data goes beyond distances of 50m range. Through utilizing graph neural networks for information aggregation and extraction of features from the radar tensor, we improved the object detection performance by $+10\%$ and achieved state-of-the-art performance. We demonstrated that, due to their properties, graph neural networks are very compatible with radar data.

In future work, we will investigate if the performance can be further improved by using graph convolutions with separate edge attributes in range and in beam direction. Additionally, we will evaluate the performance on pedestrians and study how well the network is able to predict the velocity of objects.

References

[1] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. Computing graph neural networks: A survey from algorithms to accelerators. *arXiv e-prints*, pages arXiv–2010, 2020. 3

[2] Kshitiz Bansal, Keshav Rungta, Siyuan Zhu, and Dinesh Bharadia. Pointillism: Accurate 3d bounding box estimation with multi-radars. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, SenSys '20, page 340353, New York, NY, USA, 2020. Association for Computing Machinery. 2, 6, 7

[3] 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. 3, 5

[4] Daniel Brodeski, Igal Bilik, and Raja Giryes. Deep radar detector. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2019. 2

[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 of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3, 5

[7] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z. Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[9] Xu Dong, Pengluo Wang, Pengyue Zhang, and Langechuan Liu. Probabilistic oriented object detection in automotive radar. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 458–467, 2020. 2, 4, 6, 7

[10] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020. 3

[11] Mingtao Feng, Syed Zulqarnain Gilani, Yaonan Wang, Liang Zhang, and Ajmal Mian. Relation graph network for 3d object detection in point clouds. *IEEE Transactions on Image Processing*, 30:92–107, 2021. 2

[12] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 5

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1, 2

[14] Christopher Grimm, Tai Fei, Ernst Warsitz, Ridha Farhoud, Tobias Breddermann, and Reinhold Haeb-Umbach. Warping of radar data into camera image for cross-modal supervision in automotive applications. *arXiv preprint arXiv:2012.12809*, 2020. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 4

[17] Jinhyeong Kim, Youngseok Kim, and Dongsuk Kum. Low-level sensor fusion network for 3d vehicle detection using radar range-azimuth heatmap and monocular image. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 3

[19] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Lake Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[20] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *Preprint*, abs/1812.05784, 2018. 1, 2

[21] Teck-Yian Lim, Amin Ansari, Bence Major, Daniel Fontijne, Michael Hamilton, Radhika Gowaikar, and Sundar Subramanian. Radar and camera early fusion for vehicle detection in advanced driver assistance systems. In *Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems*, 2019. 2

[22] Chunze Lin, Jiwen Lu, Gang Wang, and Jie Zhou. Graininess-aware deep feature learning for pedestrian detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1

[23] Tsung-Yi Lin, Piotr Dollr, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 4

[24] Jakob Lombacher, Kilian Laudt, Markus Hahn, Juergen Dickmann, and Christian Wohler. Semantic radar grids. pages 1170–1175, 06 2017. 2

[25] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subramanian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 4, 5, 6, 7, 8

[26] Michael Meyer and Georg Kuschk. Deep Learning Based 3D Object Detection for Automotive Radar and Camera. *European Radar Conference (EuRAD)*, 2019. 2

[27] Michael Meyer, Georg Kuschk, and Sven Tomforde. Complex-valued convolutional neural networks for automotive scene classification based on range-beam-doppler tensors. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020. 1

[28] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. Carrada dataset: Camera and automotive radar with range-angle-doppler annotations. *arXiv preprint arXiv:2005.01456*, 2020. 3, 5

[29] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 2

[30] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[31] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99, 2015. 2

[33] Tobias Scheck, Adarsh Mallandur, Christian Wiede, and Gangolf Hirtz. Where to drive: free space detection with one fisheye camera. *Twelfth International Conference on Machine Vision (ICMV 2019)*, Jan 2020. 1

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

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 8

[36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. *arXiv preprint arXiv:1812.04244*, 2018. 2

[37] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 2

[38] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019. 2

[39] Merrill I Skolnik. *Radar handbook*. McGraw-Hill Education, 2008. 1, 4

[40] Petre Stoica, Jian Li, and Yao Xie. On probing signal design for mimo radar. *IEEE Transactions on Signal Processing*, 55(8):4151–4161, 2007. 4

[41] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 2

[42] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Radar object detection using cross-modal supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 504–513, January 2021. 7

[43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. 3

[44] Zhi Yan, Li Sun, Tomas Krajnik, and Yassine Ruichek. EU long-term dataset with multiple sensors for autonomous driving. *CoRR*, abs/1909.03330, 2019. 3, 5

[45] Junbo Yin, J. Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11492–11501, 2020. 2

[46] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *arXiv preprint arXiv:2006.11275*, 2020. 2

[47] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2