CenterPoint Transformer for BEV Object Detection with Automotive Radar

Loveneet Saini
University of Wuppertal, Aptiv
Wuppertal, Germany
loveneet.saini@aptiv.com

Yu Su
Aptiv
Wuppertal, Germany
nanosuyu@gmail.com

Hasan Tercan
University of Wuppertal
Wuppertal, Germany
tercan@uni-wuppertal.de

Tobias Meisen
University of Wuppertal
Wuppertal, Germany
meisen@uni-wuppertal.de

Abstract

Object detection in Bird's Eye View (BEV) has emerged as a prevalent approach in automotive radar perception systems. Recent methods use Feature Pyramid Networks(FPNs) with large yet limited receptive fields to encode object properties. In contrast, Detection Transformers (DETRs), known for their application in image-based object detection, use a global receptive field and object queries with set losses. However, applying DETRs to sparse radar inputs is challenging due to limited object definition, resulting in inferior set matching. This paper addresses such limitations by introducing a novel approach that uses transformers to extract global context information and encode it into the object's center point. This approach aims to provide each object with individualized global context awareness to extract richer feature representations. Our experiments, conducted on the public NuScenes dataset, show a significant increase in mAP for the car category by 23.6% over the best radar-only submission, alongside notable improvements for object detectors on the Aptiv dataset. Our modular architecture allows for easy integration of additional tasks, providing benefits as evidenced by a reduction in the mean L2 error in velocity prediction across different classes.

1. Introduction

Vehicles equipped with perception systems are often engineered to accommodate applications such as automatic emergency braking (AEB) and automatic valet parking (AVP), using a sensor suite comprising camera, radar, and lidar. These complex applications could be derived from a multi-task neural network with object detection, velocity prediction, and segmentation tasks as outputs. Compared to camera and lidar, radar provides accurate Doppler measurements with extended range and an all-weather perception capability. Radar wave reflections from targets are processed by a signal processing chain that estimates parameters such as arrival angle to produce range-azimuth-Doppler (RAD) feature maps for the scene in the radar's field of view (FOV). Although these generated maps are of low resolution compared to the camera images, they provide novel object detection capabilities based on unique target signatures in the radar measurement space. In this work, we develop a model to utilize such maps for object detection and later extend it to additionally support the velocity prediction task.

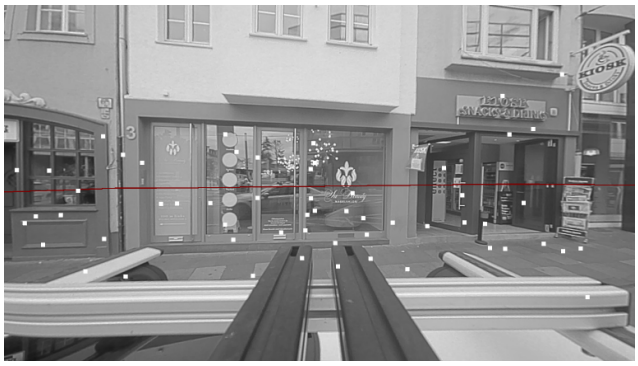

Figure 1. Typical point reflections (white squares) from radar overlaying image from camera: Limited object definition in radar creates poor conditions for driving set matching in DETRs

Radar-based object detection methods extract features from radar measurements to regress bounding boxes and determine class categories for each object. To tackle the inherent challenges associated with the sensor [36][38], recent works use deep learning models based on radar point cloud [23] or Birds Eye View (BEV) maps [20, 35]. Point cloud based methods process the radar measurement on a

per-point basis or by introducing specific locality while the BEV grid input allows for a natural extraction of local patterns. Furthermore, clustering the reflections to generate a region of interest (ROI) and classifying the clusters with deep learning models have also shown encouraging results [22]. While the ROI-based models are simpler to implement, processing the full BEV scene map provides a richer representation.

In the camera vision domain, recent BEV grid based works use key-point based models to process the data for encoding object's properties in the specific keypoints like corners [15] or center [37]. Such methodology of feature extraction is advantageous as it allows to start search and differentiate objects with respect to their characteristic points. Center point based architectures [37] [25] use heat-map regression to identify the points along with specially designed receptive fields like Feature Pyramid Networks (FPNs) [17] for detecting the object and have provided state-of-the-art results [25] for image input modality.

In radar measurement space, where objects are perceived by capturing point reflections of radar waves, the resulting input tensor for the model has a very low resolution with very high dynamic noise [36]. The features of this input tensor are limited to position and Doppler velocity, offering significantly less information compared to the rich color details found in camera images (see Figure 1). Consequently, objects have poor apparency due to the sparse nature of the tensor and the lack of distinct edges or boundaries. However, recent studies [10, 19] on camera images have demonstrated that vision transformer architectures [9] have low reliance on high frequency information, such as textures, for object classification, while preserving spatial information through the network layers. These findings suggest the potential applicability of such architectures to radar data.

The challenges associated with radar input tensor make the transition from Convolutional Neural Networks (CNNs) to transformers not straightforward. As detailed in later sections, simply replacing existing models with an off-the-shelf transformer does not directly improve performance. To this end, in this paper we propose a tailored transformer model for the widely used Range-Azimuth-Doppler (RAD) radar input format. This model features a novel decoder layer that provides a meaningful performance improvement for the object detection task. Such a decoder layer aims to provide each object with an individualized global context by using learnable queries. Specifically for limited object apparency in radar, we bridge the use of a centerpoint-based detection approach and query learning while avoiding extra set matching computations that require well-defined objects such as those available in a camera input. We further extend the model by adding an additional velocity prediction task in parallel with detection to take advantage of the learned richer feature representation.

We evaluate the proposed model on the public NuScenes dataset [3] and a larger and more complex Aptiv dataset [2] for the task of object detection. The contributions of this paper are as follows:

- A transformer model for radar based BEV object detection using Range-Azimuth-Doppler tensor inputs to exploit a global receptive field
- An efficient decoder layer for incorporating individualized global context for each object, eliminating the need for manually defined regions of interest
- An integration of center-point architectures with detection transformers, addressing a significant gap in the existing literature, needed particularly for radar data.

2. Related Work

Recent works on radar based object detection use end-to-end deep learning on either BEV radar image, point clouds, or a combination in a multi-modal input architecture. Depending on the input, different neural architectures and feature extractors are used, e.g. Graph Neural Networks (GNNs) for point clouds or CNNs for input BEV maps.

Point cloud-based methods [23] [21] typically employ pointwise processing using shared MLPs with global pooling operations, or relational feature processing introducing edges between points using a graph data structure. Such approaches exhibit permutation in-variance and can process either the point cloud of the whole scene or a specific ROI. However, these works are inspired by their success on lidar input data. For radar, the low point density and high noise pose a significant challenge in extracting meaningful features with point cloud input processing approaches. Furthermore, such architectures require significant additional processing to create a graph input by computing nearest neighbors for each point.

Methods operating on BEV radar image have been proposed for object detection [20, 35] or object type classification [29]. These works typically use FPN-based [16] architectures with CNNs to implicitly incorporate global context with growing receptive field, albeit limited and slow growing throughout the architecture [18]. Meanwhile, in camera domain, for efficient target detection, these FPNs have been combined with center point based object property extraction [25] to use the techniques of key point estimation for object detection. Such designs have achieved encouraging performances.

Recent research has explored the use of transformers for radar data in classification [6] and segmentation tasks [7, 34], demonstrating their potential to address specific challenges associated with radar data. Furthermore, studies on camera images such as by Ghiasi et al. [10], have shown that the attention-guided global receptive field of transformers differ substantially in tackling object classification task than their CNN counterparts. In particular, trans-

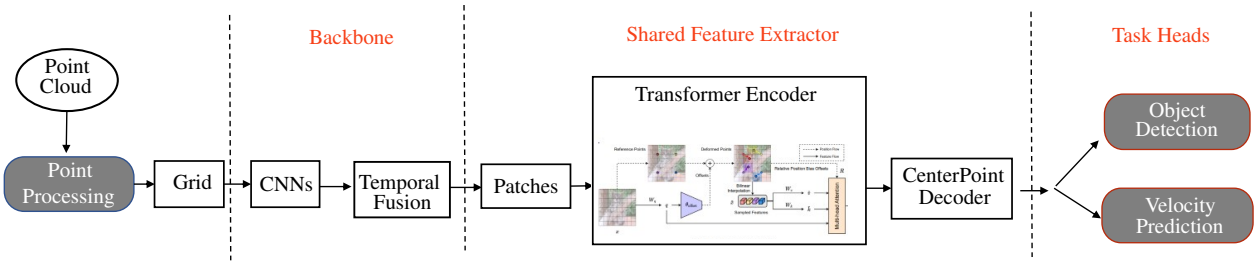

Figure 2. Overall architecture with replaceable shared feature extractor

formers do not rely on rich image features like texture for class differentiation. The dynamic attention weights allow for integrating adaptability to the architecture with respect to a dynamic component of the input such as noise. These findings underscore the suitability of transformers for radar based object detection. Pioneering works in vision, such as DETR [4] and Deformable DETR [40], have provided the methodology of performing object query based detection with set matching approaches. Such approaches presume a clear appearance of the object in the input data, which is valid for camera images, but less so for the sparse radar input. This discrepancy highlights the need for the adaptations presented in this paper.

There have been recent works in the literature using transformers for object detection with radar input, but these are based on unconventional data formats such as the raw analog-to-digital converter (ADC) signal [11] or a time-varying range-Doppler diagram [12]. For our work, we use the conventional RAD format, also present in the NuScenes dataset, which has been shown to provide coherent information of the radar data [8] and to improve the performance of deep learning models compared to other formats [20, 35].

3. Proposed Method

3.1. Overview

Our work proposes a transformer-based model for radar BEV object detection that operates on RAD feature map of the scene in the radar's FOV. We start designing our architecture by using popular FCOS [25] CNN based architecture for camera images as a baseline design template for our radar input. We adapt the original FCOS architecture such that it uses an input of pre-processed received radar reflections with (i) an initial grid processing backbone, (ii) a shared feature extractor neck, and (iii) a center point object detection head. Similar to FCOS, we initially adopt a Feature Pyramid Network (FPN) as the feature extractor neck. The use of FPNs on radar inputs has been previously explored in [20, 27, 35], where the authors have used pyramids for detection with BEV radar image input.

In the next step, to take advantage of the radar-input transformers, we keep the backbone and head fixed and replace the FPN in the neck with a standard off-the-shelf transformer encoder to analyze the performance change. Such a stepwise design philosophy allows us to directly compare the effectiveness of the transformers in the ablation section and provides more insight into how we arrived at our final design. We then motivate the need for our adaptations and update the network by introducing a novel decoder layer (placed after the off-the-shelf encoder), that provides global context from the encoder to each object individually by using object queries. The overall architecture with such a decoder is shown in Figure 2 with the backbone, neck, and head sections. Initially, it is designed to perform enhanced BEV object detection with radar, but we later extend it to additionally support velocity prediction task.

3.1.1 Radar Signal Pre-Processing

Figure 3 shows the standard radar signal processing chain for the received reflections at each point. The radar signal is first processed by a 2D Fast Fourier Transform (FFT). Then, the range-Doppler spectrum resulting from the previous processing step is applied to a CFAR detector, which extracts radar targets. These detections are used to estimate the arrival directions (angles) of the targets. This process is applied to all input radar point clouds to obtain common point features. Finally, the processed point cloud is projected onto a 2D BEV grid, effectively creating an image-like input. For this projection, we use the Pillar Feature Net from the popular PointPillars [14], which divides the cloud space in the form of pillars, which are similar to bins in vertical direction. Similar techniques have been used in most of the literature like [13, 24, 27, 31], hence such block is used here "as is" and further details can be accessed form the original work.

3.1.2 Backbone and Task Head

In the backbone stage, the input BEV grid is processed by convolution layers to introduce spatial awareness of the grid input in the network (cf. Figure 2). Temporal fusion is then performed to accumulate historical information and reduce noise. However, due to the sparse nature of the NuScenes

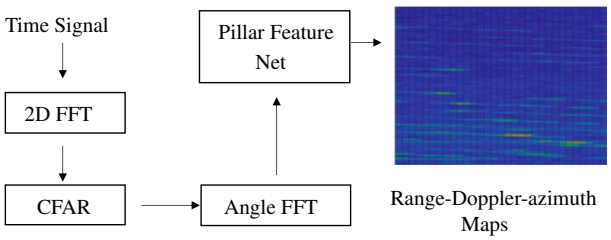

Figure 3. Per point pre-processing chain for input radar reflection

data (sampling at 2Hz), such temporal fusion is skipped for this dataset and used only for the Aptiv dataset. In the detection head, the target processor with non-maximum suppression (NMS) from the Fully Convolutional One-Stage Object Detection (FCOS) network [25] is employed. This uses a centerness score for keypoint estimation with heat map regression to identify and collect the top-K center candidates. As shown in their work, FCOS outperforms all anchor-based methods because it can use as many foreground samples as possible to train the regressor. The identified center candidates are finally regressed to obtain object properties after training using a focal loss based approach [17] as defined in FCOS.

3.1.3 Feature Extractor

An integral part of the architecture shown in Figure 2 is the common feature extractor which is responsible for extracting generalized feature representations to be used by multiple task heads. As explained above, we first integrate a CNN-based FPN to extract features at different resolutions of the input along with max-pooling and upsampling layers. This design facilitates the establishment of a receptive field that adeptly captures global object features of varying sizes. FPNs represent a reliable representative for CNN based designs, given their demonstrated efficacy with radar [27] [20] [35], hence we use this feature extractor later for our ablation study.

Next, we select an encoder from the domain of image-based detection transformers to replace the FPN in the neck. Our choice is the deformable attention encoder from [30] due to two benefits: a linear computational complexity (unlike the quadratic complexity observed in other transformers) and a deformed global receptiveness. Because of the absence of complex color features such as textures and sharpness in radar, we use only single-stage patching from the original work with kernel and stride of size 4. For this single-stage process, we replace the standard convolution with a dynamic convolution [5] featuring three parallel kernels (for K=4) as shown in Table 1, and incorporate learnable positional embeddings for patch generation. This dynamic kernel increases the representational capacity for low-resolution objects with radar inputs. The further operation of the deformable attention is not explained, as the encoder is directly adopted from the cited research without modifications.

Kernel	Stride	Pooling
(K, K)	(K, K)	–
$(K/2, K/2)$	$(K/2, K/2)$	$(2, 2)$
$(K/4, K/4)$	$(K/4, K/4)$	$(4, 4)$

Table 1. Dynamic convolution parameters for creating patches with kernel size K

3.1.4 Need for Adaptation

The difference between an FPN and a transformer can theoretically be traced back to the nature of the receptive field. By design, transformers with access to all pixels can capture global information in early layers compared to CNNs, as shown in [10] for the object type classification (OTC) task. However, a key requirement for transformers seems to be the need for global summarization of the objects exposed by the encoder. This is shown by the role of the [CLS] token for OTC in [9] and also by the localization of an object by a decoder for its detection in DETR [4]. For the architecture in Figure 2 with an off-the-shelf transformer with the same backbone and centerpoint target processor, such a component is missing. With the above reasoning, we find an obvious need to complement the transformer encoder with a decoder.

For images, an object query based decoder with cross attention to the encoder output is proposed in DETR and similar models [4, 40]. Given the well-defined visibility of objects in an image, matching losses complemented with auxiliary losses for each decoder layer are used to drive the learning of the object queries without any initial guesses. As discussed in the literature [36], objects in radar are not well defined (Figure 1), resulting in set matching failure. Therefore, intuitively such data may require more complex and accurate matching criterion for e.g. Mahalanobis distance based matching box loss rather than Euclidean L1 loss, which is currently proposed in DETRs to deal with the data limitations. Instead, in this paper, by combining the advantages of both transformers and center-based networks and avoiding the complexity of set matching computations, we propose an alternative design of decoders for radar use cases.

3.2. Centerpoint Decoder

Figure 4 shows our proposed decoder block which consists of a binary cross-attention (Bi-Attn.) and a context injection (CI) module. As shown in the figure, the three inputs of the N^{th} decoder block are the output of the encoder, M learnable

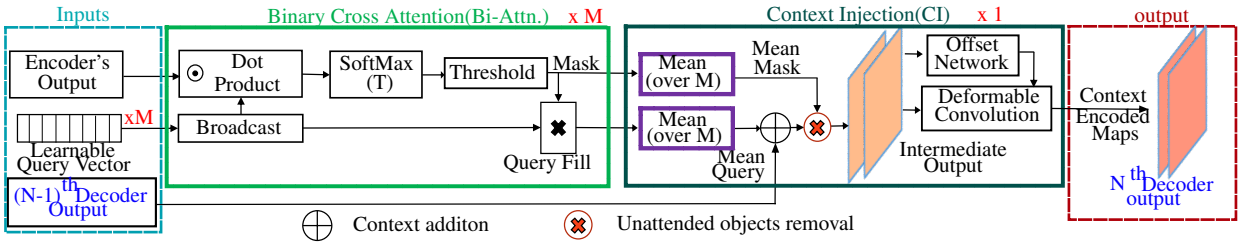

Figure 4. Illustration of the proposed decoder (N[th]block)

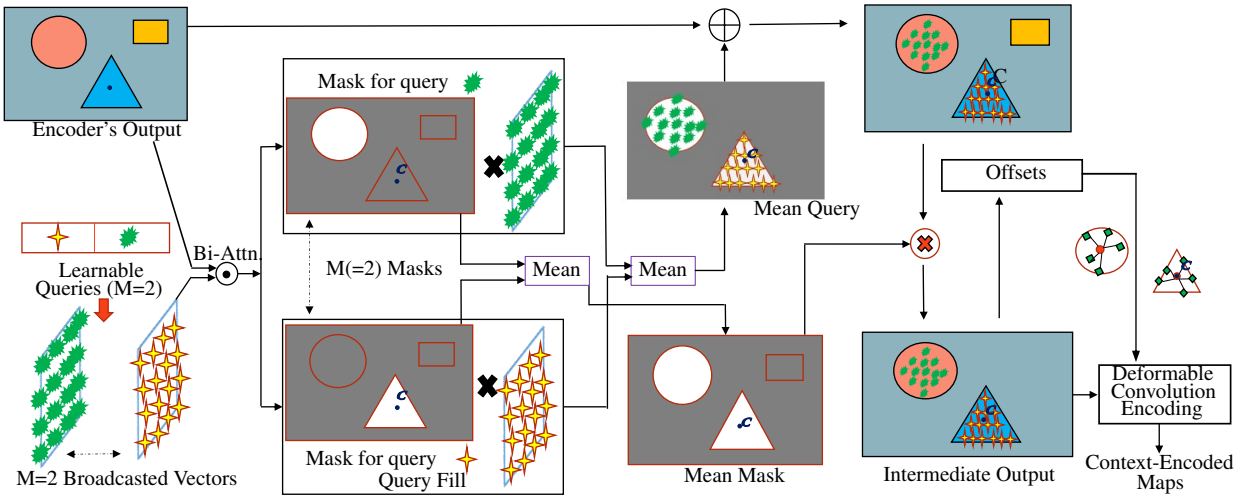

Figure 5. Visualization of the working principle of the (N=1) decoder block

query vectors (parameters), and the output of the previous N-1[th] decoder block. For the first (N=1) decoder block, the input "N-1[th] decoder" is set as the output of the encoder. One of the main goals of the decoder is to encode the global context information (from the encoder) to the center of each object, so that the center point target processor can be used for decoding with radar data. Furthermore, the global context is made available to all points for each object to improve and enable multiple tasks such as velocity prediction, segmentation, etc. In previous works [22, 26], achieving such a task required creating imprecise regions of interest for each object, followed by applying pooling operations on a global scale. In our approach, however, we achieved these goals by using learnable query parameters that bypass the need for explicit ROI construction.

In order to clarify the details of the decoder block design, we describe its working principle in the following. Figure 5 shows the basic principle for M=2 for the first (N=1) decoder block. Essentially, M different learnable queries (like a template) are used to learn an object-level representation for ideally M different objects. First, the binary cross attention module allows each query to examine the entire encoded input (keys) by broadcasting, while using dot product similarity measurement to identify objects for learning. To relax the matching requirement, the softmax is annealed at temperature T to increase the entropy of the distribution of matching locations in the input. Furthermore, a thresholding is performed that maps the distribution to binary values 0 and 1 based on a selected adaptive threshold level (median or mean of the resulting non-uniform distribution) and creates a binary mask (attention maps) corresponding to each query, highlighting the "associated" locations to that object, similar to an ROI (assuming a triangular mask for the star query in Figure 5). The binary attention module designed here differs from conventional cross attention [28], where after the dot product, the obtained attention map is multiplied by "values" to output updated queries. Here, we don't use "values" (Figure 4), but we discretize the map to output a binary mask.

These steps result in M discrete masks corresponding to M queries. Since each query is learnable, it acquires specific information about an object (or part of it) in the scene, which is indicated by a corresponding mask. Therefore, when the brodcasted query is multiplied by its respective discrete mask, as shown by query fill in Figure 4 and 5, it effectively populates the locations where the mask value is 1 with features from the corresponding queries. This operation is performed on all queries before culminating in the

context injection module to generate a "mean query". This query averages all relevant queries for each location.

Next, the "mean query" is added to the output of the last layer (the encoder output for the intial decoder layer) to provide global context to each location within the encoded representation, as shown by the "context addition" operation in Figure 4 and correspondingly in Figure 5. Since the center of the object is one such location, it becomes aware of its global context. In the centerpoint object detection head, when a specific object center is identified via heatmap regression [37] and regressed to its desired attributes, the gradient can be propagated to its associated query for learning. This eliminates the need for initial bipartite query-object matching, as required by DETR's set-matching approach. In addition, the availability of global context for all "associated" locations enriches the feature representations that could be used subsequently by additional task heads.

With the above procedure, all queries should ideally learn about every object present in the scene. However, this is a strong assumption, since not all queries may capture all relevant objects. In addition, individual queries may not correspond to an entire object (as shown by the full triangle with star query in Figure 5), but rather to specific facets of the same object, leading to limited ROI creation. To overcome this limitation, a "mean mask" is created by combining only the masks produced by each query, highlighting only the locations attended by all M queries. Consequently, multiplying this binary "mean mask" by the aggregated output (unattended object removal in Figure 4 and 5) effectively eliminates information about unattended locations by queries from the aggregated feature map to produce "intermediate output". This reasoning also justifies the need to use more than a single decoder block (N) to ensure sufficient flexibility to capture both complete objects (multiple ROIs for the same object) and all objects present in the encoded input.

Finally, to explicitly incorporate global context, the "intermediate output" is fed into a deformable convolution operator, such as the one introduced in [39]. Here, this operator regresses deformable offsets for each location, and then performs convolution based on these offset-defined locations. This method is inspired by the work of Yang et al. [33], which notes that the regressed offsets for the center point can be correlated with the bounding boxes for each object. Since each location in the output of the decoder layer has some awareness of either the whole or a specific facet of the object to which it is linked, the regressed offsets here can be assumed to embody object bounding properties. The goal of providing each object with unique global context awareness is achieved by encoding these boundary properties into the features of each "associated" location through convolution operations with offset locations.

Figure 6 depicts the decoder layer arrangement for N decoder blocks. Multiple decoder blocks are used to ensure that all objects are fully captured from the encoder output. The outputs of each decoder block are interpolated and aggregated to create the final generalized feature representation. Such connections are designed so that the gradient can flow directly to each of the N query vectors of N decoder blocks. Since no auxiliary losses are used, this is an important step to ensure that the gradient is not diminished for all learnable queries, as it would be in a feed-forward style. In addition, cross-connections between blocks are used to facilitate indirect communication between queries, and the first query is set to 0 to output an identity mask.

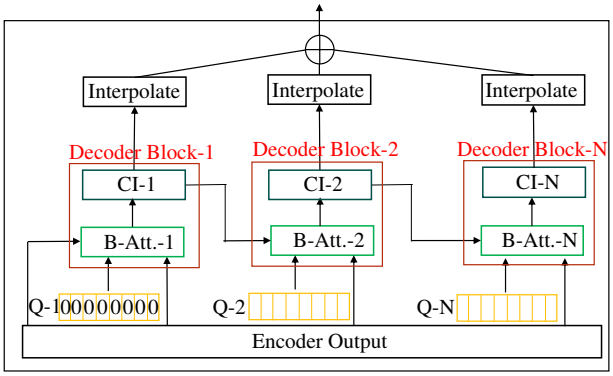

Figure 6. An exemplary CenterPoint Decoder layer illustrating the use of the proposed decoder blocks with binary cross attention (B-Att.) and context injection (CI) modules

4. Experiments

4.1. Dataset

Given the large domain gap in terms of different sensor characteristics between different radar sensors and radar data levels, we tested our model on two different datasets. The NuScenes dataset [3] restricts the number of submissions for its test dataset, leading us to conduct our exhaustive comparison with all relevant works reimplemented from the literature on its official validation dataset, following the test approach in [27]. Furthermore, although our network is capable of detecting object types such as buses and trucks, it's noteworthy that the current state-of-the-art on radar-only detectors for the NuScenes dataset primarily evaluates models based on their performance in detecting the "car" class, as seen in the works [23, 27, 31, 32]. Consequently, in line with these works, our quantitative evaluation on the dataset focuses on the "car" class. This class represents the majority of bounding boxes in the dataset.

The Aptiv dataset [2] features an egocentric setup where a grid of 280 $\times$ 160 cells is projected around the vehicle. The recorded observations on the x-y plane are used as input to our model and are grouped into the classes listed

Object detector	Type	AP4.0 (%) ↑	mAP(%) ↑	rel. mAP ↑
GNN [23]	point-based	24.7	13.7	−47.7%
PointPillars [14]	grid-based	37.0	22.0	−16.0%
RPFA-Net [31]	grid-based	38.3	23.1	−11.8%
KPConvPillars [27]	hybrid	42.2	26.2	baseline
Centerpoint Tf. (ours)	grid-based	**43.4**	**32.4**	**+23.6 %**

Table 2. Quantitative benchmark for class *car* on the NuScenes dataset

in Table 3. "Large vehicles include types such as trucks, buses, tankers, trailers, and vans, while small vehicles include cars, auto-rickshaws, mini-trucks, etc. The dataset includes detections of objects within 40 meters of the radar's field of view (FOV), including complex real-world scenarios involving Large Stationary Vehicles (LVS) and Vehicle Stationary (VS). These classes cover cases such as fully occluded objects that are invisible to both lidar and camera, but detectable in radar data via multipath propagation. However, challenges such as angular resolution limitations make it difficult to separate objects at longer distances, making this dataset very challenging. In our work, a total of 21776 scenes are used for training and 9294 scenes are used for the test set with ground truth semantics derived from annotated lidar point clouds.

Vehicle Moving (VM)
Vehicle Stationary (VS)
Large Vehicle Moving (LVM)
Large Vehicle Stationary (LVS)

Table 3. Class Categories for Aptiv Dataset

4.2. Implementation Details

We trained all the networks presented here using the Adam optimizer with a learning rate of 1×10^{-4} for twenty-nine epochs and a rate of 1×10^{-5} for the last 30^{th} epoch, with a batch size of 1 on an Nvidia 2080 GPU. The models are fitted using the sigmoid focal-based loss [17] to account for class imbalance in the training dataset. For bounding box regression, the smooth L1 loss is used. For the FPN design, the best performance was found with a three-level pyramid designed with max-pooling and upsampling layers. For the off-the-shelf transformer encoder-only network, a single-stage patching using the form of the work [30] with a dynamic kernel size K=4 is used. For the single stage, two layers of encoders are used with each using K=8 offsets along-with a query grouping factor of 4 inspired by the default settings in [30]. For the proposed center-point transformer network, only an additional center-point decoder layer is introduced after the transformer encoder (the same encoder retained from the off-the-shelf network implementation) just before the head, as shown in Figure 2. For each decoder block, the query vector size is set to 64 with a total of M=32 queries, and a temperature of 2.0 with median thresholding is used for the binary cross attention module. In total, including zero query initialization, 1+3 decoder blocks are used in the decoder layer.

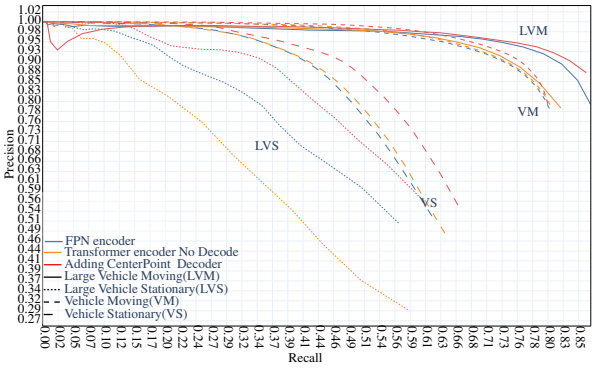

Figure 7. Precision-Recall curves illustrating the detailed performance of the proposed CenterPoint Transformer (TF.) against the FPNs and encoder only network at all operating conditions

4.3. Results

In our comparison of the proposed model with other detectors, we use the quantitative analysis presented in [27], which reimplements and evaluates models from the literature on the NuScenes dataset with radar data as exclusive input. Note that we don't compare our model to the leading model of the NuScenes 2024 (radar modality only) detection leaderboard, RadarDistill [1], because it uses lidar data in addition to radar data to distill knowledge and therefore cannot be used for a fair comparison. Instead, we choose the next best approach from leaderboard, KPConvPillars [27] as our baseline. Also, as described in Section 2, we don't compare our model to radar transformers [11, 12] because they can't be applied to the RAD radar data format of the NuScenes dataset.

Table 2 presents the results of our experiments on NuScenes. When comparing the AP4.0 and the mean average precision (average of AP4.0, AP2.0, AP1.0 and AP0.5

for a car class), they show a 23.3% improvement over the current the state-of-the-art grid based detectors. Furthermore, significant improvement is observed in mAP as compared to AP 4.0% (for a matching threshold of 4m), indicating the model performs significantly better even for smaller matching distances.

4.4. Ablation

For the ablation of our model, we use the Aptiv dataset which has higher number of data samples and complex real world cases, allowing us to provide deeper insights of our work by analyzing detailed precision-recall curves. These curves highlight the efficacy of models at different precison and recall metrics trade-offs. As detailed in Section 3.1, we decompose our overall architecture shown in Figure 2 by replacing its neck with three different variants: (i) CNN-based FPN as feature extractor, (ii) replacing the FPN with an off-the-shelf transformer encoder, and (iii) keeping the encoder but adding the additional centerpoint decoder block (Figure 2).

Figure 7 compares the performance of the proposed model (enhanced with an additional decoder) against the CNN-based FPN and its substitution with a transformer encoder alone across all precision and recall metrics for four vehicular classes. It can be observed that the substitution does not yield any performance improvement, indicating that solely having a global receptive field is not sufficient and needs to be summarized. Furthermore, this substitution results in a performance decline for the LVS (the most challenging) class. These findings are consistent with the arguments presented in Section 3.1.4, which highlight the necessity of incorporating a decoder in transformer-based models.

The curves demonstrate that adding our centerpoint decoder to the existing transformer encoder significantly enhances performance across all classes and operating conditions, as shown by a larger area under the curve (AUC). This enhancement means that any operating point chosen will deliver superior performance compared to CNNs without necessitating trade-offs between precision and recall metrics. Hence, by introducing a modest number of N*M(3*64) learnable query vectors, the centerpoint decoder achieves significant performance improvements across all classes when combined with the transformer encoder for radar data.

Since our model is based on center-point target processing, it has limited capabilities when applied to classes where a center point is not easily definable, such as in the semantic segmentation of background classes. However, unlike DETRs, which primarily output a list of updated object queries (Bi-attn. in Section 3.2), our network generates a feature map encoded with global context. This rich representation allows for the addition of parallel task heads to the detection, enhancing the multitasking capabilities of the model. We illustrate this by extending it to support velocity prediction task. Essentially, we train our network with Aptiv dataset to regress velocity prediction in addition to detection properties. Figure 8 illustrates the performance of our model for velocity prediction compared to the FPN feature extractor baseline. The curves, which represent the L2 mean average error of velocity prediction across different operating conditions (recall), show a significant reduction in velocity prediction error across all classes when replacing CNNs with centerpoint transformers. Most notably, a reduction of about 20% in velocity prediction error is observed for the large moving vehicle category.

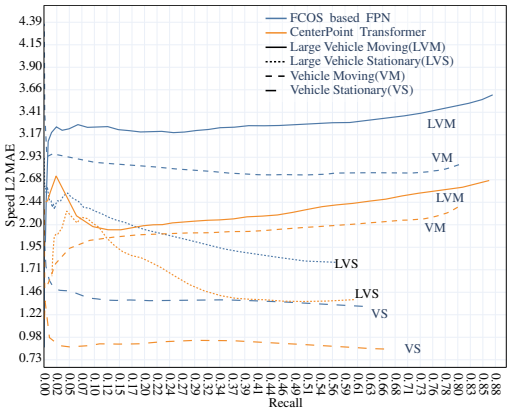

Figure 8. L2 mean error performance comparison curves for the velocity prediction task.

5. Summary

In this work, we introduced a new transformer architecture for detecting objects in Bird's Eye View (BEV) using radar data by exploiting centerpoint based target processing. This architecture aims to incorporate an individualized global context for each object in the scene without the need to explicitly define regions of interest. Our approach, extensively evaluated on two different datasets, demonstrates significant performance improvements over existing radar-only object detection models. Furthermore, our architecture replaces the inefficient set-matching traditionally required for object detection with transformers by using an effective centerpoint representation of objects. This approach not only generates a rich feature representation instead of mere object queries as output, but also facilitates the performance of additional tasks besides object detection. As our evaluation shows, this includes a reduction in the L2 mean error for velocity prediction compared to the baseline FPN feature extractor. Future work will further enhance the multitasking capabilities of the architecture by adding complex tasks such as semantic segmentation and improving the use of centerpoint representations for background classes.

References

[1] Geonho Bang, Kwangjin Choi, Jisong Kim, Dongsuk Kum, and Jun Won Choi. Radardistill: Boosting radar-based object detection performance via knowledge distillation from lidar features. *arXiv preprint arXiv:2403.05061*, 2024. 7

[2] Marco Braun, Moritz Luszek, Jan Siegemund, Kevin Kollek, and Anton Kummert. Quantification of uncertainties in deep learning-based environment perception. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, pages 1–8. IEEE, 2021. 2, 6

[3] 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, 2020. 2, 6

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3, 4

[5] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11030–11039, 2020. 4

[6] Yahia Dalbah, Jean Lahoud, and Hisham Cholakkal. Radarformer: Lightweight and accurate real-time radar object detection model. In *Scandinavian Conference on Image Analysis*, pages 341–358. Springer, 2023. 2

[7] Yahia Dalbah, Jean Lahoud, and Hisham Cholakkal. Transradar: Adaptive-directional transformer for real-time multi-view radar semantic segmentation, 2023. 2

[8] Yahia Dalbah, Jean Lahoud, and Hisham Cholakkal. Transradar: Adaptive-directional transformer for real-time multi-view radar semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 353–362, 2024. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020. *arXiv preprint arXiv:2010.11929*, 2010. 2, 4

[10] Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. What do vision transformers learn? a visual exploration. *arXiv preprint arXiv:2212.06727*, 2022. 2, 4

[11] James Giroux, Martin Bouchard, and Robert Laganiere. T-fftradnet: Object detection with swin vision transformers from raw adc radar signals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4030–4039, 2023. 3, 7

[12] Tiezhen Jiang, Long Zhuang, Qi An, Jianhua Wang, Kai Xiao, and Anqi Wang. T-rodnet: Transformer for vehicular millimeter-wave radar object detection. *IEEE Transactions on Instrumentation and Measurement*, 72:1–12, 2022. 3, 7

[13] Daniel Köhler, Maurice Quach, Michael Ulrich, Frank Meinl, Bastian Bischoff, and Holger Blume. Improved multi-scale grid rendering of point clouds for radar object detection networks. In *2023 26th International Conference on Information Fusion (FUSION)*. IEEE, 2023. 3

[14] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds, 2019. 3, 7

[15] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 4, 7

[18] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 2

[19] Xu Ma, Huan Wang, Can Qin, Kunpeng Li, Xingchen Zhao, Jie Fu, and Yun Fu. A close look at spatial modeling: From attention to convolution. *arXiv preprint arXiv:2212.12552*, 2022. 2

[20] 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. 1, 2, 3, 4

[21] Zhijun Pan, Fangqiang Ding, Hantao Zhong, and Chris Xiaoxuan Lu. Moving object detection and tracking with 4d radar point cloud. *arXiv preprint arXiv:2309.09737*, 2023. 2

[22] Loveneet Saini, Axel Acosta, and Gor Hakobyan. Graph neural networks for object type classification based on automotive radar point clouds and spectra. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 5

[23] 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, 2, 6, 7

[24] Bin Tan, Zhixiong Ma, Xichan Zhu, Sen Li, Lianqing Zheng, Sihan Chen, Libo Huang, and Jie Bai. 3d object detection for multi-frame 4d automotive millimeter-wave radar point cloud. *IEEE Sensors Journal*, 2022. 3

[25] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2, 3, 4

[26] Michael Ulrich, Claudius Gläser, and Fabian Timm. Deepreflecs: Deep learning for automotive object classification with

radar reflections. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–6. IEEE, 2021. 5

[27] Michael Ulrich, Sascha Braun, Daniel Köhler, Daniel Niederlöhner, Florian Faion, Claudius Gläser, and Holger Blume. Improved orientation estimation and detection with hybrid object detection networks for automotive radar, 2022. 3, 4, 6, 7

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 5

[29] Tristan Visentin Visentin, Daniel Rusev Rusev, Bin Yang Yang, Michael Pfeiffer Pfeiffer, Kilian Rambach Rambach, and Kanil Patel Patel. Deep learning-based object classification on automotive radar spectra. 2019. 2

[30] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4794–4803, 2022. 4, 7

[31] Baowei Xu, Xinyu Zhang, Li Wang, Xiaomei Hu, Zhiwei Li, Shuyue Pan, Jun Li, and Yongqiang Deng. Rpfa-net: A 4d radar pillar feature attention network for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3061–3066. IEEE, 2021. 3, 6, 7

[32] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 6

[33] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9657–9666, 2019. 6

[34] Matthias Zeller, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Gaussian radar transformer for semantic segmentation in noisy radar data. *IEEE Robotics and Automation Letters*, 8(1):344–351, 2022. 2

[35] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganiere. Raddet: Range-azimuth-doppler based radar object detection for dynamic road users. In *2021 18th Conference on Robots and Vision (CRV)*, pages 95–102. IEEE, 2021. 1, 2, 3, 4

[36] Taohua Zhou, Mengmeng Yang, Kun Jiang, Henry Wong, and Diange Yang. Mmw radar-based technologies in autonomous driving: A review. *Sensors*, 20(24):7283, 2020. 1, 2, 4

[37] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 6

[38] 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):4208, 2022. 1

[39] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019. 6

[40] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3, 4