MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Multi-View Radar Detection Transformer with Differentiable Positional Encoding

Yataka, Ryoma; Wang, Pu; Boufounos, Petros T.; Takahashi, Ryuhei

TR2025-027 March 08, 2025

Abstract

The Radar dEtection TRansformer (RETR) has recently been introduced to fuse multi-view millimeter-wave radar heatmaps by leveraging the detection transformer architecture and a geometric learning framework for indoor radar perception. A notable feature of RETR is its tunable positional encoding (TPE), which allows for adjusting the significance of depth positional embedding across multiple views to promote depth- prioritized feature association. However, the TPE ratio is pre- determined, rather than being optimized during the training process. In this paper, we propose a differentiable positional encoding (DiPE) scheme for RETR by automatically adjusting the TPE ratio during the training for enhanced performance and avoiding exhaustive grid value search. DiPE can be applied along with either pre-fixed (e.g., sinusoidal) or learnable positional embeddings, achieved by multiplying dual differentiable masks over the depth and angular positional embedding vectors. Comprehensive evaluations on the open MMVR dataset demonstrate that the proposed DiPE not only simplifies the determination of the TPE ratio but also enhances the overall detection performance.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Multi-View Radar Detection Transformer with Differentiable Positional Encoding

Ryoma Yataka[1,2], Pu (Perry) Wang[2], Petros Boufounos[2], and Ryuhei Takahashi[1]

[1]*Information Technology R&D Center, Mitsubishi Electric Corporation*, Kanagawa 247-8501, Japan
[2]*Mitsubishi Electric Research Laboratories (MERL)*, Cambridge, MA 02139, USA

***Abstract*—The Radar dEtection TRansformer (RETR) has recently been introduced to fuse multi-view millimeter-wave radar heatmaps by leveraging the detection transformer architecture and a geometric learning framework for indoor radar perception. A notable feature of RETR is its tunable positional encoding (TPE), which allows for adjusting the significance of depth positional embedding across multiple views to promote depth-prioritized feature association. However, the TPE ratio is predetermined, rather than being optimized during the training process. In this paper, we propose a differentiable positional encoding (DiPE) scheme for RETR by automatically adjusting the TPE ratio during the training for enhanced performance and avoiding exhaustive grid value search. DiPE can be applied along with either pre-fixed (e.g., sinusoidal) or learnable positional embeddings, achieved by multiplying dual differentiable masks over the depth and angular positional embedding vectors. Comprehensive evaluations on the open MMVR dataset demonstrate that the proposed DiPE not only simplifies the determination of the TPE ratio but also enhances the overall detection performance.**

***Index Terms*—Indoor radar perception, object detection, detection transformer, positional encoding.**

I. Introduction

Compared to cameras and LiDAR, radar provides safer and more reliable perception in low light, harsh weather, and hazardous conditions, all at lower cost. Its applications now extend beyond outdoor automotive sensing [1]–[5] to indoor use cases like elder care, energy management, and navigation [2], [6]–[8]. However, a key limitation of indoor radar perception is coarse semantic features extractable from radar signals. Early efforts use low-resolution point clouds from radar sensors with an angular resolution of 15°, mainly for supporting simple classification tasks [9]–[12]. For more fine-grained downstream tasks such as object detection, pose estimation, and segmentation, recent approaches favor low-level representation formats such as radar heatmaps and even raw analog-to-digital (ADC) data from high-resolution radar sensors such as cascading radar chips with an angular resolution of 1.3° or sub-1-deg [13]–[17].

RF-Pose employs an autoencoder architecture to fuse radar heatmaps from horizontal and vertical planes, directly regressing keypoint heatmaps for pose estimation in the 2D image plane [18]. This autoencoder-based RF-Pose directly incorporates the intrinsic radar-to-camera coordinate transformation as part of the model's inductive bias. RFMask identifies candidate regions on the horizontal plane and assigns a fixed-height to each candidate region, generating 3D bounding boxes (BBoxes) in the 3D radar coordinate space [16].

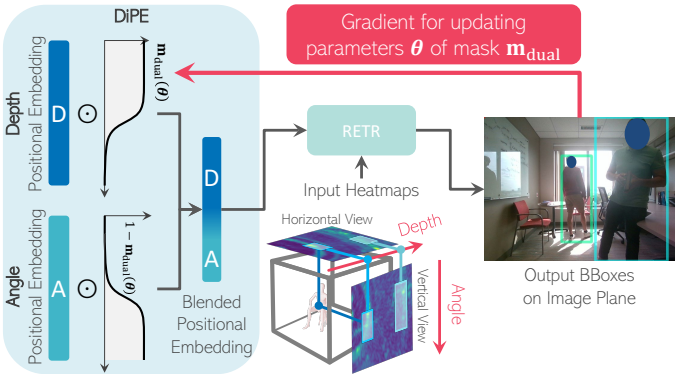


Fig. 1. The DiPE scheme. Depth and angle positional embeddings at a full dimension are multiplied respectively with a differential mask and its complementary. The blended positional embedding is obtained by summing these two embeddings. The overall RETR loss is backpropagated to the learnable mask parameters to dynamically adjust the dimension ratio.

More recently, [19] introduces a radar detection transformer (RETR) by leveraging the detection transformer (DETR) architecture [20] and the attention mechanism [21] to fuse radar features from two views and connect them with learnable object queries, while incorporating learnable radar-to-camera coordinate transformation, a tri-plane loss function on both radar and camera coordinate spaces, and a tunable positional encoding (TPE). Particularly, the TPE is designed to promote higher similarity scores for keys and queries with similar depth position than those far apart in depth by allowing for adjustable dimensions between depth and angular embeddings. However, determining the optimal dimension ratio is challenging and may be dataset-dependent, as it requires exhaustive search and evaluation over a set of grid values in $[0, 1]$.

In this paper, we propose a *Differentiable Positional Encoding (DiPE)* scheme for RETR by automatically adjusting the TPE dimension ratio during training for enhanced performance and avoiding the exhaustive grid search. As illustrated in Fig. 1, this is achieved by introducing a monotonically decreasing differential mask function and its complementary, referred to as the dual masks, and multiplying them respectively over depth and angle embeddings at a full dimension. Besides the learnable RETR weights, the tri-plane loss is backpropagated to the mask parameters to dynamically adjust the dimensions of the depth and angle positional embedding during training. We consider both pre-fixed (e.g., sinusoidal) or learnable positional embeddings for the initial depth and angle positional embeddings.

II. Radar Perception with Multi-view Radar

A. Radar Perception

Consider a pair of horizontal and vertical antenna arrays sending frequency modulated continuous waveform pulses and denote the horizontal (azimuth-depth) radar heatmaps $\mathbf{Y}_{\texttt{hor}} \in \mathbb{R}^{T\times W\times D}$ and the vertical (elevation-depth) radar heatmaps $\mathbf{Y}_{\texttt{ver}} \in \mathbb{R}^{T\times H\times D}$ with a shared depth axis by including T consecutive frames, where W, H and D denote the number of cells of width (azimuth), height (elevation) and depth, respectively. We are interested in detecting objects in the image plane by taking the two radar heatmaps as inputs

$$\mathbf{F}_{\texttt{image}} = \texttt{proj}_{\texttt{image}}\left(\mathcal{T}\left(f\left(\mathbf{Y}_{\texttt{hor}}, \mathbf{Y}_{\texttt{ver}}\right)\right)\right), \tag{1}$$

where $\mathbf{F}_{\texttt{image}}$ denotes predicted bounding boxes (BBoxes) in the image plane, f denotes the 3D object detection module in the radar coordinate system, $\mathcal{T}$ denotes the radar-to-camera coordinate transformation, and $\texttt{proj}_{\texttt{image}}$ denotes the 3D-to-2D image projection.

B. Radar Detection Transformer

In [19], RETR employs the detection transformer with both transformer encoder and decoder modules for f, a learnable geometry-preserving coordinate transformation for $\mathcal{T}$, and a known pinhole camera model for the projection $\texttt{proj}_{\texttt{image}}$.

In the module f, we obtain the feature embeddings: $\mathbf{Z}_{\texttt{hor/ver}} = \texttt{backbone}\left(\mathbf{Y}_{\texttt{hor/ver}}\right)$ with ResNet18 [22]. A transformer encoder expects a sequence of features as input. This is done by mapping the feature maps into a sequence of $2K$ multi-view radar features: $\mathbf{H}^0 = [\mathbf{H}_{\texttt{hor}}, \mathbf{H}_{\texttt{ver}}] \in \mathbb{R}^{C\times 2K}$ where $\mathbf{H}_{\texttt{hor/ver}}$ can be extracted from $\mathbf{Z}_{\texttt{hor/ver}}$ with top-K selection, and C is the number of channels. Then, we obtain the memory with transformer encoder layers: $\mathbf{H}^{l+1} = \texttt{encoder}\left(\mathbf{H}^l\right)$ $(l = 0, 1, \cdots, L_{\texttt{self}} - 1)$. For each decoder layer, it takes N object queries: $\mathbf{Q}^l \in \mathbb{R}^{C\times N}$ as its input, and consists of a self-attention layer, a cross-attention layer and a feed forward network (FFN). It updates all queries through multi-head self-attention: $\bar{\mathbf{Q}}^l = \texttt{decoder}_{\texttt{self}}\left(\mathbf{Q}^l\right)$ and through multi-head cross-attention with the memory: $\mathbf{Q}^{l+1} = \texttt{decoder}_{\texttt{cross}}\left(\bar{\mathbf{Q}}^l, \mathbf{H}^{L_{\texttt{self}}}\right)$ $(l = 0, 1, \cdots, L_{\texttt{cross}} - 1)$. Each decoder embedding $\mathbf{q} \in \mathbf{Q}^{L_{\texttt{cross}}}$ is converted to 3D BBox: $\bar{\mathbf{g}} = \{cx, cy, cz, w, h, d\}^\top$ with FFN, where (cx, cy, cz) is the 3D center and w, h and d are the sizes.

Each 3D BBox $\bar{\mathbf{g}}$ is transformed to 3D camera coordinate with learnable rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$: $\mathbf{g}^i_{\texttt{camera}} = \left\{x^i_{\texttt{camera}}, y^i_{\texttt{camera}}, z^i_{\texttt{camera}}\right\}^\top = \mathcal{T}\left(\mathbf{g}^i_{\texttt{radar}}\right) = \mathbf{R}\mathbf{g}^i_{\texttt{radar}} + \mathbf{t}$ $(i = 1, 2, \cdots, 8)$ for each 3D corner point: $\mathbf{g}^i_{\texttt{radar}} = \left\{x^i_{\texttt{radar}}, y^i_{\texttt{radar}}, z^i_{\texttt{radar}}\right\}^\top$ of the 3D BBox corresponding to $\bar{\mathbf{g}}$. Finally, we obtain the corresponding 2D BBox on the image plane: $\widehat{\mathbf{b}}_{\texttt{image}} = \{cx, cy, w, h\}^\top = \texttt{proj}_{\texttt{image}}\left(\mathbf{G}_{\texttt{camera}}\right)$ where $\mathbf{G}_{\texttt{camera}} = \left\{\mathbf{g}^i_{\texttt{camera}}\right\}_{i=1}^8$.

Since the multi-view radar features lack positional information and the self-attention is permutation-invariant, we supplement $\mathbf{H}^l$ and $\mathbf{Q}^l$ with positional encoding concatenated to the input of each encoder and decoder layers. In RETR, fixed sine/cosine positional encoding along the depth and angular (azimuth or elevation) dimension is used as the specific embedding, and it can be easily extend to other positional encoding approach such as learning based embedding.

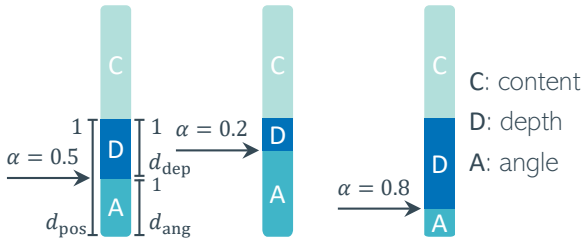


Fig. 2. The factorization of tunable positional encoding (TPE).

C. Tunable Positional Encoding

Positional encoding has a crucial role to give the spatial positional information to each feature (embedding $\mathbf{h} \in \mathbf{H}^l$ or decoder embedding $\mathbf{q} \in \mathbf{Q}^l$) in the transformer. RETR exploited the fact that the two radar views share the depth axis and introduced a tunable positional encoding (TPE) as the inductive bias to prioritize the relative importance of depth and avoid exhaustive correlations between the views. TPE is composed of a depth and an angular (either azimuth or elevation) axes. As such, $\mathbf{p} = \mathbf{d} \oplus \mathbf{a} \in \mathbb{R}^{d_{\texttt{pos}}}$ with $\mathbf{d} \in \mathbb{R}^{d_{\texttt{dep}}}$ representing the depth positional embedding, $\mathbf{a} \in \mathbb{R}^{d_{\texttt{ang}}}$ the angular positional embedding, and the concatenation $\oplus$ similar to conditional DETR [23]. TPE is built on the concatenation between the content embedding $\mathbf{c}$ (token) and positional embedding $\mathbf{p}$, that is $\mathbf{c} \oplus \mathbf{d} \oplus \mathbf{a}$. TPE can promote higher similarity scores with similar depth embeddings than those far apart in depth, especially for the ones from different views, by allowing for adjustable dimensions between depth and angular embeddings as shown in Fig. 2:

$$d_{\texttt{dep}} = \alpha d_{\texttt{pos}},\ d_{\texttt{ang}} = (1-\alpha) d_{\texttt{pos}},\ d_{\texttt{dep}} + d_{\texttt{ang}} = d_{\texttt{pos}}, \tag{2}$$

where α is the tunable dimension ratio and is in the range $[0, 1]$. This α has to be determined by exhaustive pre-experiments. Therefore, automatic determination method is desired for α.

III. Differentiable Positional Encoding

A. Differentiable Mask Function

Assume that we consider a function $h : [a, b] \to \mathbb{R}$, which we desire to be non-zero only in a subset $[c, d] \subseteq [a, b]$. To this end, we can multiply h with a mask m whose values are non-zero only on $[c, d]$, e.g., a rectangular mask $\Pi_{[c,d]}(x) = \mathbf{1}_{[c,d]}$. However, as the gradient of the mask is either zero or non-defined, it is not possible to learn the interval in which it is non-zero by backpropagation. To overcome the limitation, we use a parametric smooth mask $m\left(\cdot, \boldsymbol{\theta}\right)$ which interval of non-zero values is defined by its parameters $\boldsymbol{\theta}$. By using this mask, we can apply the backpropagation to learn the interval on which it is non-zero as the mask is differentiable and learnable. Specifically, we design the mask m parameterized by its offset and its temperature $\boldsymbol{\theta} = \{\mu, \tau\}$, inspired by [24]:

$$m\left(x; \boldsymbol{\theta}\right) = 1 - \frac{1}{1 + \exp\left(-\tau\left(x - \mu\right)\right)},\ \text{s.t. } \mu \geq 0. \tag{3}$$

B. Dual Masking

To automatically determine the parameter α in TPE, we propose a differentiable positional encoding (DiPE) using a mask m. In the TPE, the dimensions $d_{\texttt{dep}}$ and $d_{\texttt{ang}}$ are determined based on the α, and after computing the embeddings for

each axis, the two embeddings are concatenated to generate a positional embedding with dimension $d_{\texttt{pos}}$. In our DiPE, we first generate positional embeddings of dimension $d_{\texttt{pos}}$ for each axis in advance. Then, using the parameters $\boldsymbol{\theta}$, we generate a mask and apply the dual masking:

$$\mathbf{p} = \mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right) \odot \mathbf{d} + \left(\mathbf{1} - \mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right)\right) \odot \mathbf{a}_{\texttt{f}}, \tag{4}$$

where $\mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right) = \left\{m\left(1;\boldsymbol{\theta}\right), ..., m\left(d_{\texttt{pos}};\boldsymbol{\theta}\right)\right\}^{\top}$ is the vector collected with each dimension i, $\mathbf{1}$ is a vector with all elements of 1, $\odot$ represents Hadamard product, and $\texttt{f}$ is an operation that flips the order of the vector's elements: $\mathbf{a}_{\texttt{f}}^{(i)} = \mathbf{a}^{(d_{\texttt{pos}}+1-i)}$. An example of the implementation of Eq. 4 is to use a fixed sine/cosine positional encoding:

$$\mathbf{p}^{(2i)} = m\left(2i\right)\sin\left(\frac{\texttt{p}_{\texttt{dep}}}{T^{\frac{2i}{d_{\texttt{pos}}}}}\right) + \left(1 - m\left(2i\right)\right)\sin\left(\frac{\texttt{p}_{\texttt{ang}}}{T^{\frac{2i}{d_{\texttt{pos}}}}}\right), \tag{5}$$

$$\mathbf{p}^{(2i+1)} = m\left(2i+1\right)\cos\left(\frac{\texttt{p}_{\texttt{dep}}}{T^{\frac{2i}{d_{\texttt{pos}}}}}\right) + \left(1 - m\left(2i+1\right)\right)\cos\left(\frac{\texttt{p}_{\texttt{ang}}}{T^{\frac{2i}{d_{\texttt{pos}}}}}\right),$$

where $i = 0, 1, \cdots, d_{\texttt{pos}}/2 - 1$, $\texttt{p}_{\texttt{dep/ang}}$ is the position index, and $T = 10^4$ is a temperature. The attention weight is based on the dot-product between query ($\texttt{q}$) and key ($\texttt{k}$):

$$\begin{aligned}
&\left(\mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right) \odot \mathbf{d}_{\texttt{q}} + \left(\mathbf{1} - \mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right)\right) \odot \mathbf{a}_{\texttt{f,q}}\right)^{\top} \\
&\qquad \left(\mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right) \odot \mathbf{d}_{\texttt{k}} + \left(\mathbf{1} - \mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right)\right) \odot \mathbf{a}_{\texttt{f,k}}\right) \\
&= \left(\bar{\mathbf{d}}_{\texttt{q}} + \mathbf{a}_{\texttt{f,q}} - \bar{\mathbf{a}}_{\texttt{f,q}}\right)^{\top}\left(\bar{\mathbf{d}}_{\texttt{k}} + \mathbf{a}_{\texttt{f,k}} - \bar{\mathbf{a}}_{\texttt{f,k}}\right) \qquad (6) \\
&= \bar{\mathbf{d}}_{\texttt{q}}^{\top}\bar{\mathbf{d}}_{\texttt{k}} + \bar{\mathbf{a}}_{\texttt{f,q}}^{\top}\bar{\mathbf{a}}_{\texttt{f,k}} + \mathbf{a}_{\texttt{f,q}}^{\top}\bar{\mathbf{d}}_{\texttt{k}} - \bar{\mathbf{a}}_{\texttt{f,q}}^{\top}\bar{\mathbf{d}}_{\texttt{k}} + \bar{\mathbf{d}}_{\texttt{q}}^{\top}\mathbf{a}_{\texttt{f,k}} \\
&\qquad - \bar{\mathbf{d}}_{\texttt{q}}^{\top}\bar{\mathbf{a}}_{\texttt{f,k}} + \mathbf{a}_{\texttt{f,q}}^{\top}\mathbf{a}_{\texttt{f,k}} - \mathbf{a}_{\texttt{f,q}}^{\top}\bar{\mathbf{a}}_{\texttt{f,k}} - \bar{\mathbf{a}}_{\texttt{f,q}}^{\top}\mathbf{a}_{\texttt{f,k}}, \qquad (7)
\end{aligned}$$

where $\bar{\mathbf{x}} = \mathbf{m}_{\texttt{dual}}\left(\boldsymbol{\theta}\right) \odot \mathbf{x}$. Eq. 6 contains blended components according to τ.

Fig. 3 illustrates the DiPE. The $\mathbf{m}_{\texttt{dual}}$ is a monotonically decreasing function with $\boldsymbol{\theta}$, and applying this mask has the effect of attenuating the influence of the latter dimensions of $\mathbf{d}$. Conversely, the $\mathbf{1} - \mathbf{m}_{\texttt{dual}}$ is a monotonically increasing function with $\boldsymbol{\theta}$ that is in a dual relationship, and applying this mask attenuates the influence of the former dimensions of $\mathbf{a}$. Therefore, adding these two vectors together effectively blends the elements of $\mathbf{d}$ and $\mathbf{a}$ using $\boldsymbol{\theta}$, replacing the adjustable dimension with α described in Eq. 2 with the learnable $\boldsymbol{\theta}$.

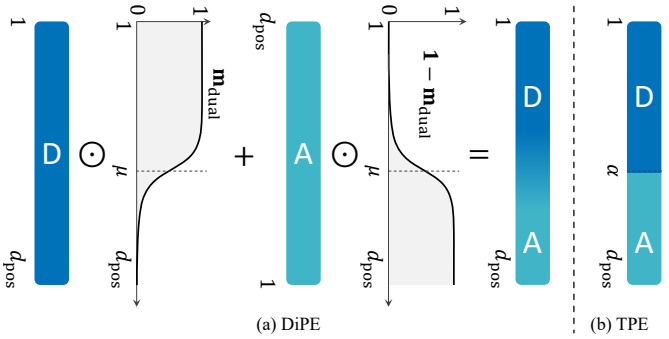
C. Architecture

The simplest way to implement the mask is to use $\boldsymbol{\theta} = \{\mu, \tau\}$ as learnable parameters and flows gradients for each of them. However, since these parameters are constrained within a specific range and may become large (e.g., $\mu \in \left[1, d_{\texttt{pos}}\right]$), it is essential to take these factors into account. Therefore, we apply a sigmoid function and scaling factor $s = d_{\texttt{pos}}$ to unconstrained parameters $\bar{\boldsymbol{\theta}} = \{\bar{\mu}, \bar{\tau}\}$, allowing the mask to effectively operate across each dimension of the embedding:

$$\mu = s \times \texttt{sigmoid}\left(\bar{\mu}\right), \ \tau = \texttt{sigmoid}\left(\bar{\tau}\right), \tag{8}$$

where $\texttt{sigmoid}\left(x\right) = \frac{1}{1+\exp(-x)}$. On the other hand, depending on the initial values of $\bar{\boldsymbol{\theta}}$ and the learning rate, the learning process may either fail to converge if the values are far from optimal, or the values may exhibit little change from their initial values (see Section IV-B). To address this issue, we design a module using a multi-layer perceptron (MLP):

$$\bar{\boldsymbol{\theta}} = \{\bar{\mu}, \bar{\tau}\} = \texttt{MLP}\left(\mathbf{e}\right), \ \text{s.t. } \mathbf{e} \in \mathbb{R}^{d_{\texttt{e}}}, \tag{9}$$



(a) DiPE (b) TPE

Fig. 3. Scheme of DiPE. (a) DiPE blends the depth and the angle ingredients with learnable mask parameters $\boldsymbol{\theta} = \{\mu, \tau\}$; (b) TPE uses the fixed α.

where $\mathbf{e}$ is a learnable parameters for generating $\hat{\boldsymbol{\theta}}$ and is initialized with normal distribution. We set $d_{\texttt{e}} = 32$. We construct $\texttt{MLP}$ via three linear layers and apply leaky ReLU function [25] after the first two layers. As a result, $\boldsymbol{\theta}$ becomes more sensitive in the learning process, making it easier to obtain optimal parameters.

D. Loss

We calculate a matching cost matrix with a loss $\mathcal{L}$ constructed from a classification loss $\mathcal{L}_{\texttt{class}}$ and a tri-plane BBox loss $\mathcal{L}_{\texttt{box}}^{\texttt{tri}}$ which is sum of BBox losses from the three types of planes (horizontal, vertical and image planes):

$$\mathcal{L}_{\texttt{box}}^{\texttt{tri}} = \sum_{p \in \{\texttt{hor},\texttt{ver},\texttt{image}\}} \mathcal{L}_{\texttt{box}}\left(\mathbf{b}_p, \widehat{\mathbf{b}}_p\right), \tag{10}$$

$$\mathcal{L}_{\texttt{box}}(\mathbf{b}_p, \widehat{\mathbf{b}}_p) = \lambda_{\texttt{GIoU}}\mathcal{L}_{\texttt{GIoU}}(\mathbf{b}_p, \widehat{\mathbf{b}}_p) + \lambda_{\texttt{L}_1}\mathcal{L}_{\texttt{L}_1}(\mathbf{b}_p, \widehat{\mathbf{b}}_p), \tag{11}$$

where $\mathbf{b}_p$ is the ground truth, $\widehat{\mathbf{b}}_p$ is the prediction and each λ is the weight coefficient. $\mathcal{L}_{\texttt{GIoU}}$ and $\mathcal{L}_{\texttt{L}_1}$ denote the generalized intersection over union (GIoU) loss [26] and the ℓ_1 loss, respectively. To optimize $\boldsymbol{\theta} = \{\mu, \tau\}$, we need to compute the gradient $\nabla\mathcal{L}\left(\boldsymbol{\theta}\right)$. Our mask m is differentiable for μ and τ, respectively, and the derivatives are:

$$\frac{\mathrm{d}m}{\mathrm{d}\mu} = \frac{\tau\exp\left(-\tau\left(x-\mu\right)\right)}{\left(1+\exp\left(-\tau\left(x-\mu\right)\right)\right)^2}, \tag{12}$$

$$\frac{\mathrm{d}m}{\mathrm{d}\tau} = -\frac{\left(x-\mu\right)\exp\left(-\tau\left(x-\mu\right)\right)}{\left(1+\exp\left(-\tau\left(x-\mu\right)\right)\right)^2}. \tag{13}$$

The gradient $\nabla\mathcal{L}\left(\boldsymbol{\theta}\right)$ can be backpropagated to Eq. 12 and Eq.13 by auto-differentiation, and thus the optimal $\boldsymbol{\theta}^*$ can be determined by learning.

IV. Experiments

A. Experimental Setup

We use the indoor radar dataset: MMVR [17] as same as [19]. MMVR includes multi-view radar heatmaps collected from 25 human subjects across 6 rooms over a span of 9 days. In our implementation, we use data from Protocol 2 (P2) which includes 237.9K frames capturing multiple subjects. For the training-validation-test split, we follow the data split S1 as defined in MMVR.

We implemented two positional encodings: sine/cosine encoding (`Sinusoid`) and learnable embedding (`Learned`), and we consider RFMask [16], DETR [20] and RETR [19] as the

TABLE I
DETECTION RESULTS ON MMVR. RFMASK DOES NOT HAS PE. FOR RETR WITH TPE, `Trained` $\boldsymbol{\theta}$ CORRESPONDS TO A RATIO α.

Model	PE	Trained $\boldsymbol{\theta} = \{\mu, \tau\}$	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
RFMask	-	-	31.37	61.50	27.48	33.23	38.41
DETR	Sinusoid	-	29.38	62.31	25.35	31.32	43.06
DETR	Learned	-	29.05	62.84	23.85	31.11	42.73
RETR with TPE	Sinusoid	0.60 / -	46.75	83.80	46.06	42.19	57.39
RETR with TPE	Learned	0.60 / -	46.71	82.27	45.09	41.61	56.22
RETR with DiPE (Ours)	Sinusoid	0.90 / 0.94	47.09	84.15	46.14	44.43	59.18
RETR with DiPE (Ours)	Learned	0.67 / 0.32	47.75	83.72	46.31	42.11	56.37

TABLE II
ABLATION STUDY OF MASK MODULE ON MMVR.

Mask Parameters	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Scalar	44.93	83.31	43.19	42.11	55.52
MLP	47.09	84.15	46.14	44.43	59.18

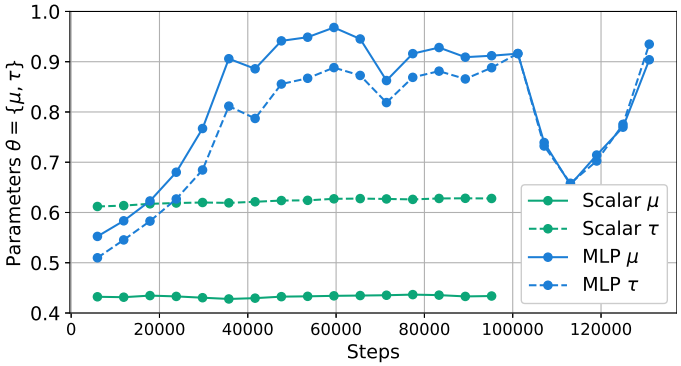


Fig. 4. Comparison of training curves on mask parameters. MLP is highly flexible and can be easier to obtain optimal parameters.

baseline methods. For DETR and RETR, we set the embedding dimension to $d_{\text{pos}} = 256$, and the ratio at $\alpha = 0.6$ which was determined in [19]. Other hyper-parameters are same as [19] including learning rate, early stopping, etc.

To evaluate the performance of object detection, we adopt average precision at two IoU thresholds of 0.5 (AP_{50}) and 0.75 (AP_{75}) and its mean (AP) over thresholds $[0.5 : 0.05 : 0.95]$ as the metrics. We also consider average recall when it is restricted to making only one detection (AR_1) or up to 10 detections (AR_{10}) per input.

B. Results

Table I shows the comparison of the detection models. "PE" and "`Trained` $\boldsymbol{\theta} = \{\mu, \tau\}$" denote the type of positional encoding and the parameters $\boldsymbol{\theta}$ obtained after training, respectively. For RETR with TPE, the value of α from Eq. 2 is shown as μ, while for Ours, μ (Note that we show the value before scaling in Eq. 8) and τ are displayed. The results show that Ours provides a finer optimal value, which is not always consistent with the α used in [19] because of the blending with τ. The main advantage of Ours is the ability to easily optimize and simplify the determination for the $\boldsymbol{\theta}$, which also contributes to improved detection performance.

Table II shows the comparison of the results when $\bar{\boldsymbol{\theta}}$ is directly used as a parameters (Scalar) and when Eq. 9 is used

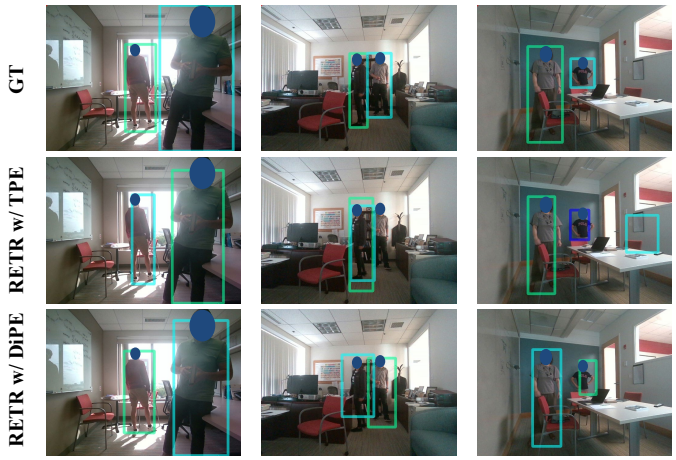


Fig. 5. Visualization of the object detection. GT denotes the ground truth, and each row represents a different room environment.

(MLP). We used RETR with DiPE and Sinusoid as the base-method. It is evident that using MLP leads to an improvement in detection performances. Fig. 4 compares the training curves of Scalar and MLP. Note that the final number of steps differs due to the use of early stopping. It can be observed that Scalar shows minimal change from the initial value, with slight fluctuations. On the other hand, MLP includes significant fluctuations, indicating that it finds an optimal value regardless of the initial values. This fact demonstrates that using the designed architecture effectively activates our method.

We show the visualization in Fig. 5. Compared to RETR with TPE, RETR with DiPE demonstrates an improved ability to correct subtle positional discrepancies of BBoxes and effectively reduces the occurrence of transient false positives. This improvement is attributed to its more efficient use of the depth axis from the heatmaps, allowing for a more accurate representation of the 3D positions derived from the two views.

V. CONCLUSION

To realize the indoor monitoring, we focused on radar perception using multi-view radar. Our approach is based on the RETR and aimed to obtain an optimal ratio with learning which is an issue of TPE. To achieve this, we introduced a dual masking and proposed the DiPE with the masks. This can not only determine the ratio of each axis, but also show the improvement of the detection performance. Comprehensive evaluations on the open MMVR dataset demonstrate that the DiPE not only simplifies the determination of the TPE ratio but also enhances the overall detection performance.

REFERENCES

[1] S. Sun, A. P. Petropulu, and H. V. Poor, "MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 98–117, 2020.

[2] A. Pandharipande, C.-H. Cheng, J. Dauwels, S. Z. Gurbuz, J. Ibanez-Guzman, G. Li, A. Piazzoni, P. Wang, and A. Santra, "Sensing and machine learning for automotive perception: A review," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 11 097–11 115, 2023.

[3] R. Yataka, P. Wang, P. Boufounos, and R. Takahashi, "Radar perception with scalable connective temporal relations for autonomous driving," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 13 266–13 270.

[4] P. Li, P. Wang, K. Berntorp, and H. Liu, "Exploiting temporal relations on radar perception for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 050–17 059.

[5] R. Yataka, P. Wang, P. Boufounos, and R. Takahashi, "SIRA: Scalable inter-frame relation and association for radar perception," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 15 024–15 034.

[6] M. Amin, *Radar for Indoor Monitoring: Detection, Classification, and Assessment*. CRC Press, 2017.

[7] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, "RF-based 3D skeletons," in *Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 267–281.

[8] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16–28, 2019.

[9] M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi, "Learning sleep stages from radio signals: A conditional adversarial architecture," in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 4100–4109.

[10] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 032–10 044, 2020.

[11] P. Gong, C. Wang, and L. Zhang, "MMPoint-GNN: Graph neural network with dynamic edges for human activity recognition through a millimeter-wave radar," in *International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–7.

[12] F. Jin, A. Sengupta, and S. Cao, "mmFall: Fall detection using 4-D mmWave radar and a hybrid variational RNN autoencoder," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 2, pp. 1245–1257, 2022.

[13] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, A. Bourdoux, W. De Neve, and T. Dhaene, "Indoor person identification using a low-power FMCW radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 7, pp. 3941–3952, 2018.

[14] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang, "Capturing human pose using mmWave radar," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[15] S. Yang and Y. Kim, "Single 24-GHz FMCW radar-based indoor device-free human localization and posture sensing with cnn," *IEEE Sensors Journal*, vol. 23, no. 3, pp. 3059–3068, 2023.

[16] Z. Wu, D. Zhang, C. Xie, C. Yu, J. Chen, Y. Hu, and Y. Chen, "RFMask: A simple baseline for human silhouette segmentation with radio signals," *IEEE Transactions on Multimedia*, vol. 25, pp. 4730–4741, 2023.

[17] M. M. Rahman, R. Yataka, S. Kato, P. Wang, P. Li, A. Cardace, and P. Boufounos, "MMVR: Millimeter-wave multi-view radar dataset and benchmark for indoor perception," in *Computer Vision – ECCV 2024*. Cham: Springer Nature Switzerland, 2025, pp. 306–322.

[18] M. Zhao, T. Li, M. A. Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7356–7365.

[19] R. Yataka, A. Cardace, P. P. Wang, P. Boufounos, and R. Takahashi, "RETR: Multi-view radar detection transformer for indoor perception," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 213—-229.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional DETR for fast training convergence," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3631–3640.

[24] D. W. Romero and N. Zeghidour, "DNArch: Learning convolutional neural architectures by backpropagation," *arXiv, 2302.05400, cs.LG*, 2023.

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning (ICML)*, vol. 30, 2013, p. 3.

[26] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.