RETR: Multi-View Radar Detection Transformer for Indoor Perception

Ryoma Yataka[1,3,*,†], **Adriano Cardace**[2,*,‡], **Pu (Perry) Wang**[1,*],
Petros Boufounos[1], **Ryuhei Takahashi**[3]
[1]Mitsubishi Electric Research Laboratories (MERL), USA
[2]Department of Computer Science and Engineering, University of Bologna, Italy
[3]Information Technology R&D Center (ITC), Mitsubishi Electric Corporation, Japan

Abstract

Indoor radar perception has seen rising interest due to affordable costs driven by emerging automotive imaging radar developments and the benefits of reduced privacy concerns and reliability under hazardous conditions (e.g., fire and smoke). However, existing radar perception pipelines fail to account for distinctive characteristics of the multi-view radar setting. In this paper, we propose Radar dEtection TRansformer (**RETR**), an extension of the popular DETR architecture, tailored for multi-view radar perception. RETR inherits the advantages of DETR, eliminating the need for hand-crafted components for object detection and segmentation in the image plane. More importantly, RETR incorporates carefully designed modifications such as 1) depth-prioritized feature similarity via a tunable positional encoding (TPE); 2) a tri-plane loss from both radar and camera coordinates; and 3) a learnable radar-to-camera transformation via reparameterization, to account for the unique multi-view radar setting. Evaluated on two indoor radar perception datasets, our approach outperforms existing state-of-the-art methods by a margin of $15.38+$ AP for object detection and $11.77+$ IoU for instance segmentation, respectively.

1 Introduction

Perception information encompasses the processes and technologies to detect, interpret, and understand their surroundings. Complementary to the mainstream camera and LiDAR sensors, radar can enhance the safety and resilience of perception under low light, adversarial weather (e.g., rain, snow, dust), and hazardous conditions (e.g., smoke, fire) at affordable device and maintenance cost. An emerging application of radar perception is indoor sensing and monitoring for elderly care, building energy management, and indoor navigation [7]. A notable limitation of indoor radar perception is the low semantic features from radar signals.

Earlier efforts use radar detection points [42, 30] to support simple classification tasks such as fall detection and activity recognition over a limited number of patterns. To support challenging perception tasks such as object detection, pose estimation, and segmentation, lower-level radar signal representation such as radar heatmaps is more preferred. Along this line, the earliest work is RF-Pose [43] using a convolution-based autoencoder network to fuse features from the two radar views and regress keypoints for 2D image-plane pose estimation. It is later extended to 3D human pose estimation [44]. It is noted that RF-Pose is not publicly accessible. More recently, RFMask [38]

[*] Equal contribution.
[†] The work was done as a visiting scientist from ITC in Mitsubishi Electric Corporation.
[‡] The work was done during his internship at MERL.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

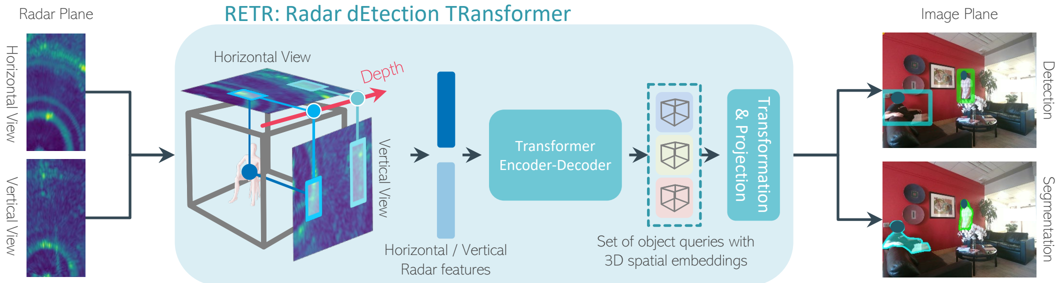

Figure 1: By taking horizontal-view and vertical-view radar heatmaps as inputs, RETR introduces a depth-prioritizing positional encoding (exploit the shared depth between the two radar views) into transformer self-attention and cross-attention modules and outputs a set of 3D-embedding object queries to support image-plane object detection and segmentation via a calibrated or learnable radar-to-camera coordinate transformation and 3D-to-2D pinhole camera projection.

borrows the Faster R-CNN framework [27] by proposing candidate regions only in the horizontal radar heatmap via a region proposal network (RPN). A corresponding proposal in the vertical radar heatmap is automatically determined using a *fixed-height* candidate region at the same depth as the horizontal proposal. The combined horizontal and vertical proposals are then projected into the image plane for bounding box (BBox) estimation. In addition, RFMask calculates the BBox loss only over the 2D horizontal radar view and disregards features from the vertical radar heatmap for BBox estimation.

In this paper, we exploit features from both horizontal and vertical radar views for object estimation and segmentation and introduce **Radar dEtection TRansformer (RETR)** (Fig. 1). RETR extends the popular Detection Transformer (DETR) [3], which effectively eliminates the need for hand-crafted components such as non-maximum suppression and proposal/anchor generation, to the multi-view radar perception. More importantly, RETR incorporates carefully designed modifications to exploit the unique multi-view radar setting such as shared depth dimension and the transformation between the radar and camera coordinate systems. Our contributions are summarized below:

1. **Extending DETR for Multi-View Radar Perception:** 1) Encoder: we associate features from both radar views by applying self-attention over the pooled multi-view radar tokens, eliminating the need for a cumbersome association scheme. We introduce a top-K feature selection to allow only K features from each view to keep the complexity low. 2) Decoder: the DETR decoder provides a natural way to associate the same object query to corresponding features from the two radar views via cross-attention. As such, the object query is able to learn 3D spatial embedding of objects in the radar coordinate (see Fig. 1).

2. **Tunable Positional Encoding**: To enhance feature association across the two radar views, we further exploit the fact that the two radar views share the depth dimension and introduce a tunable positional encoding (TPE) as an inductive bias. TPE imposes constraints in the attention map to prioritize the relative importance of depth dimension and avoid exhaustive correlations between radar views.

3. **Tri-Plane Loss from Both 3D Radar Coordinate and 2D Image Plane**: we enforce the output queries of the DETR decoder to directly predict 3D BBoxes in the radar coordinate system and convert them into the 2D image plane. We introduce a tri-plane loss that combines the BBox loss in the 3D radar plane and that in the 2D image plane, to calculate the global set-prediction loss.

4. **Learnable Radar-to-Camera Coordinate Transformation**: We employ a calibrated radar-to-camera coordinate transformation via a calibration process and a learnable coordinate transformation via reparameterization by preserving the orthonormal (i.e., 3D special orthogonal group $\mathcal{SO}(3)$) structure of the rotation matrix.

We demonstrate the effectiveness of our contributions through evaluations on two open datasets: the HIBER dataset [38] and the MMVR dataset [26].

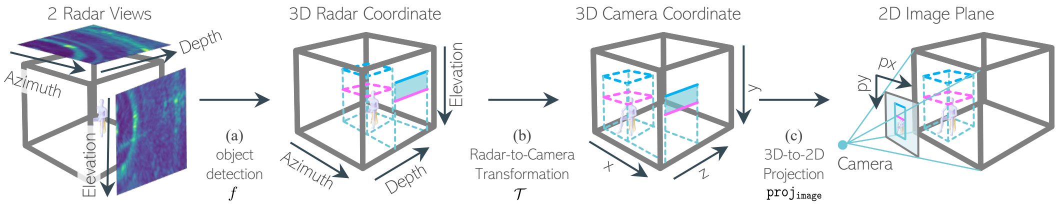

Figure 2: Indoor radar perception pipeline: (a) multi-radar views are utilized to estimate 3D BBoxes in the radar coordinate system; (b) the 3D BBoxes are then transformed into the 3D camera coordinate system by a radar-to-camera transformation; and (c) the transformed 3D BBoxes are projected onto the image plane for final object detection. **Blue line** denotes a fixed-height regional proposal in RFMask, while **Magenta line** denotes an object query with learnble height in RETR.

2 Related Work

Radar-based Object Detection and Segmentation: Indoor radar perception tasks include object detection (BBoxes), pose estimation (keypoints), and instance segmentation (human masks) [1, 35, 43, 44, 20, 23, 45, 38, 46], and radar datasets in different data formats were reported in [35, 31, 30, 39, 2, 40, 14, 38, 26]. Particularly, radar heatmap-based approaches have gained attention not only in indoor perception [43, 44, 14, 38, 26] but also for automotive radar perception [19, 24, 32, 11, 5], due to richer semantic features compared to those extracted from sparse radar point clouds [31, 30, 39, 2, 15, 40, 41]. RF-Pose [43] predicts human poses on the image plane using a convolution autoencoder-based architecture. With the HIBER dataset [38], RFMask considers proposal-based object detection and instance segmentation. More recently, MMVR [26] has been openly released to accelerate advancements in indoor radar perception.

Image-based Object Detection and Segmentation with DETR: Since the introduction of DETR for 2D image-plane object detection, subsequent studies have been developed based on its framework [21, 47, 4, 17, 37, 18, 10, 25], largely due to DETR's ability to eliminate the need for hand-designed components such as non-maximum suppression (NMS). In [21], Conditional DETR decomposes the roles of content and positional embeddings in the transformer decoder, improving not only prediction accuracy but also training convergence speed. More recently, [25] has proposed Rank-DETR as a rank-oriented architectural design, guaranteeing lower false positives and false negatives in prediction.

3 Preliminary

Generation of Radar Heatmaps: Conceptually, let us consider a pair of (virtual) horizontal and vertical antenna arrays with $N_{\texttt{ant}}$ elements for each array, sending a set of frequency modulated continuous waveform (FMCW) pulses for object detection [26, 38, 34]. The two 1D arrays generate one horizontal radar view in the azimuth-depth $(x - z)$ domain and one vertical radar view in the elevation-depth $(y - z)$ domain,

$$y_{\texttt{hor}}\left(t, x, z\right) = \sum_{k=1}^{K_p} \sum_{m=1}^{M} s_{k,m,t} e^{j2\pi \frac{d_m(x,z)}{\lambda_k}}, \quad y_{\texttt{ver}}\left(t, y, z\right) = \sum_{k=1}^{K_p} \sum_{m=1}^{M} s_{k,m,t} e^{j2\pi \frac{d_m(y,z)}{\lambda_k}}, \tag{1}$$

where $s_{k,m,t}$ denotes the k-th sample of FMCW sweep on the m-th antenna at time t, λ_k is the wavelength of the k-th sample, $d_m\left(x, z\right)$ denotes the round-trip distance from the m-th array element to a position (x, z), and K_p and M denote the number of samples and the number of array antennas, respectively. Usually, the azimuth x is in an interval of $x \in \mathcal{X} = [x_{\text{min}} : \Delta x : x_{\text{max}}]$ and the elevation y and the depth z are similarly defined. At a particular time t, we have the horizontal radar heatmap $\mathbf{Y}_{\texttt{hor}}(t) = \{|y_{\texttt{hor}}\left(t, x, z\right)|\}_{x\in\mathcal{X}}^{z\in\mathcal{Z}} \in \mathcal{R}^{W\times D}$ and the vertical radar heatmap $\mathbf{Y}_{\texttt{ver}}(t) = \{|y_{\texttt{ver}}\left(t, y, z\right)|\}_{y\in\mathcal{Y}}^{z\in\mathcal{Z}} \in \mathcal{R}^{H\times D}$ with a shared depth axis. The multi-view radar testbeds in HIBER [38] and MMVR [26] utilize advanced MIMO-FMCW radar systems. We defer the MIMO-FMCW radar heatmap generation to Appendix D.

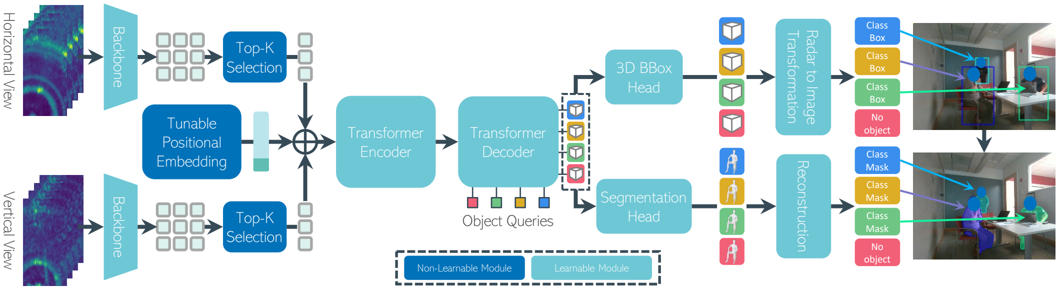

Figure 3: The RETR architecture: 1) **Encoder**: Top-K features selection and tunable positional encoding to assist feature association across the two radar views; 2) **Decoder**: TPE is also used to assist the association between object queries and multi-view radar features; 3) **3D BBox Head**: Object queries are enforced to estimate 3D objects in the radar coordinate and projected to 3 planes for supervision via a coordinate transformation; 4) **Segmentation Head**: The same queries are used to predict binary pixels within each predicted BBox in the image plane.

Indoor Radar Perception: By taking T consecutive multi-view radar heatmaps ($\boldsymbol{Y}_{\texttt{hor}} \in \mathcal{R}^{T\times W\times D}$ and $\boldsymbol{Y}_{\texttt{ver}} \in \mathcal{R}^{T\times H\times D}$) as the input, we are interested in detecting objects on the image plane:

$$\boldsymbol{F}_{\texttt{image}} = \texttt{proj}_{\texttt{image}}\left(\mathcal{T}\left(f\left(\boldsymbol{Y}_{\texttt{hor}}, \boldsymbol{Y}_{\texttt{ver}}\right)\right)\right), \tag{2}$$

where $\boldsymbol{F}_{\texttt{image}}$ denotes predicted BBoxes for object detection and pixel-level masks for instance segmentation. Using the BBox as an example in Fig. 2, our pipeline includes the following steps: 1) Fig. 2 (a): By taking the two radar views over T consecutive frames $(\boldsymbol{Y}_{\texttt{hor}}, \boldsymbol{Y}_{\texttt{ver}})$ as input, the end-to-end object detection module f outputs a set of parameters describing 3D BBoxes in the radar coordinate system; 2) Fig. 2 (b): The radar-to-camera 3D coordinate transformation $\mathcal{T}$ converts the predicted 3D BBoxes at the output of f to corresponding 3D BBoxes in the 3D camera coordinate system; 3) Fig. 2 (c): The 3D-to-2D projection $\texttt{proj}_{\texttt{image}}$ projects the 3D BBox in the camera coordinate system into corresponding 2D image plane normally with a known pinhole camera model.

4 RETR: Radar Detection Transformer

We first present the RETR architecture and then highlight radar-oriented modifications. We defer the discussion on Segmentation to Appendix B.

4.1 RETR Architecture

We present the RETR architecture in Fig. 3, introducing its major modules in a left-to-right order. Refer to Appendix A for the detailed architecture.

Backbone: Given $\boldsymbol{Y}_{\texttt{hor}} \in \mathcal{R}^{T\times W\times D}$ and $\boldsymbol{Y}_{\texttt{ver}} \in \mathcal{R}^{T\times H\times D}$, a shared backbone network (e.g., ResNet [8]) generates separate horizontal-view and vertical-view radar feature maps: $\boldsymbol{Z}_{\texttt{hor}} = \texttt{backbone}\left(\boldsymbol{Y}_{\texttt{hor}}\right) \in \mathbb{R}^{C\times\frac{W}{s}\times\frac{D}{s}}$ and $\boldsymbol{Z}_{\texttt{ver}} = \texttt{backbone}\left(\boldsymbol{Y}_{\texttt{ver}}\right) \in \mathbb{R}^{C\times\frac{H}{s}\times\frac{D}{s}}$, where C and s represent the number of channels and downsampling ratio over the spatial dimension, respectively.

Tokenization: A transformer-based encoder expects a sequence of tokens as input. This is done by mapping the feature maps into a sequence of P *multi-view radar tokens* $\boldsymbol{H} = \{\boldsymbol{H}_{\texttt{hor}}, \boldsymbol{H}_{\texttt{ver}}\} \in \mathbb{R}^{C\times P}$: $\boldsymbol{Z}_{\texttt{hor}} \rightarrow \boldsymbol{H}_{\texttt{hor}} \in \mathbb{R}^{C\times P_{\texttt{hor}}}$ and $\boldsymbol{Z}_{\texttt{ver}} \rightarrow \boldsymbol{H}_{\texttt{ver}} \in \mathbb{R}^{C\times P_{\texttt{ver}}}$, where $P = P_{\texttt{hor}} + P_{\texttt{ver}}$. We defer the tokenization discussion to Section 4.2.

Encoder as Cross-View Radar Feature Association: The transformer encoder provides a simple yet effective method for associating radar features from both horizontal and vertical views by applying self-attention over the pool of P multi-view radar tokens $\boldsymbol{H} = \{\boldsymbol{H}_{\texttt{hor}}, \boldsymbol{H}_{\texttt{ver}}\} \in \mathbb{R}^{C\times P}$, eliminating the need for cumbersome association schemes. Specifically, the l-th ($l = 0, \cdots, L_{\texttt{self}} - 1$) encoder layer updates the multi-view radar tokens through multi-head self-attention $\texttt{Att}_{\texttt{self}}$:

$$\boldsymbol{H}^{l+1} = \bar{\boldsymbol{H}}^{l} + \texttt{FFN}\left(\bar{\boldsymbol{H}}^{l}\right), \quad \bar{\boldsymbol{H}}^{l} = \boldsymbol{H}^{l} + \texttt{Att}_{\texttt{self}}\left(\texttt{Que}\left(\boldsymbol{H}^{l}\right), \texttt{Key}\left(\boldsymbol{H}^{l}\right), \texttt{Val}\left(\boldsymbol{H}^{l}\right)\right), \tag{3}$$

where FFN denotes feed-forward networks, $L_{\texttt{self}}$ is the number of encoder layers, and Que, Key and Val are projections to derive the multi-head query, key and value embedding from $\boldsymbol{H}$, respectively. For the first (0-th) layer, we have $\boldsymbol{H}^0 = \boldsymbol{H}$. Note that we omit the description of "Layer norm" and "multi-head index" in Eq. 3 for clarity.

Additionally, since the multi-view radar tokens lack positional information and the self-attention is permutation-invariant, we supplement $\boldsymbol{H}^l$ with positional embedding added (or attached) to the input of each encoder layer. Refer to Section 4.3 for a tunable positional encoding.

Decoder to Associate Object Queries with Multi-View Radar Features: The decoder provides a natural way to associate the same object query with features from the two radar views via cross-attention. For each decoder layer, it takes N object queries $\boldsymbol{Q}^l = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N\} \in \mathbb{R}^{C \times N}$ as its input, and consists of a self-attention layer, a cross-attention layer and a FFN. Specifically for the l-th $(l = 0, 1, \cdots, L_{\texttt{cross}} - 1)$ decoder layer, it first updates all queries through multi-head self-attention:

$$\bar{\boldsymbol{Q}}^l = \boldsymbol{Q}^l + \texttt{Att}_{\texttt{self}}\left(\texttt{Que}\left(\boldsymbol{Q}^l\right), \texttt{Key}\left(\boldsymbol{Q}^l\right), \texttt{Val}\left(\boldsymbol{Q}^l\right)\right), \tag{4}$$

where Que, Key and Val are the projections with different parameterization from those in the self-attention layer (Eq. 3). Then, the decoder layer further updates the object queries $\bar{\boldsymbol{Q}}^l$ of Eq. 4 via multi-head cross-attention with the multi-view radar tokens $\boldsymbol{H}^{L_{\texttt{self}}}$ from the encoder output:

$$\boldsymbol{Q}^{l+1} = \tilde{\boldsymbol{Q}}^l + \texttt{FFN}\left(\tilde{\boldsymbol{Q}}^l\right), \quad \tilde{\boldsymbol{Q}}^l = \bar{\boldsymbol{Q}}^l + \texttt{Att}_{\texttt{cross}}\left(\texttt{Que}\left(\bar{\boldsymbol{Q}}^l\right), \texttt{Key}\left(\boldsymbol{H}^{L_{\texttt{self}}}\right), \texttt{Val}\left(\boldsymbol{H}^{L_{\texttt{self}}}\right)\right), \tag{5}$$

where both $\bar{\boldsymbol{Q}}^l$ and $\boldsymbol{H}^{L_{\texttt{self}}}$ are supplemented with positional embedding. Finally, the decoder outputs N enhanced object queries $\boldsymbol{Q}^{L_{\texttt{cross}}}$ for downstream tasks.

Mapping from 3D Radar Coordinate to 2D Image Plane: Given the N enhanced object queries $\boldsymbol{Q}^{L_{\texttt{cross}}}$, RETR directly estimates 3D BBoxes in the radar coordinate:

$$\bar{\boldsymbol{g}} = \{cx, cy, cz, w, h, d\}^\top = \texttt{sigmoid}\left(\texttt{FFN}\left(\boldsymbol{q}\right)\right), \quad \boldsymbol{q} \in \boldsymbol{Q}^{L_{\texttt{cross}}} \tag{6}$$

where $\bar{\boldsymbol{g}}$ describes the 3D BBox center and respective widths along the 3D axes, and sigmoid normalizes the 3D BBox prediction to $[0, 1]$. Then, as shown in Fig. 2 (b), we apply a radar-to-camera transformation $\mathcal{T}$ to convert the predicted 3D BBoxes to ones in the 3D camera coordinate as

$$\boldsymbol{g}^i_{\texttt{camera}} = \left\{x^i_{\texttt{camera}}, y^i_{\texttt{camera}}, z^i_{\texttt{camera}}\right\}^\top = \mathcal{T}\left(\boldsymbol{g}^i_{\texttt{radar}}\right) = \boldsymbol{R}\boldsymbol{g}^i_{\texttt{radar}} + \boldsymbol{t}, \quad i = 1, 2, \cdots, 8, \tag{7}$$

where $\boldsymbol{R}$ is a 3D rotation matrix, $\boldsymbol{t} \in \mathbb{R}^3$ is the 3D translation vector, and $\boldsymbol{g}^i_{\texttt{radar}}$ is i-th corner of the 3D BBox corresponding to $\bar{\boldsymbol{g}}$. Subsequently in Fig. 2 (c), we project the 3D BBoxes $\boldsymbol{g}^i_{\texttt{camera}}$ onto the 2D image plane via a 3D-to-2D projection. From the projected 2D corners, one can calculate the 2D BBox center and width and height in the image plane as

$$\boldsymbol{b}_{\texttt{init}} = \{cx, cy, w, h\}^\top = \texttt{proj}_{\texttt{image}}\left(\boldsymbol{G}_{\texttt{camera}}\right). \tag{8}$$

The final BBox estimation $\widehat{\boldsymbol{b}}_{\texttt{image}}$ in the image plane is obtained by adding an offset head $\texttt{FFN} : \mathbb{R}^{10} \to \mathbb{R}^4$ to compensate for the spatial downsampling and normalizing it to the interval $[0, 1]$:

$$\widehat{\boldsymbol{b}}_{\texttt{image}} = \texttt{sigmoid}\left(\boldsymbol{b}_{\texttt{init}} + \texttt{FFN}\left(\boldsymbol{b}_{\texttt{init}} \oplus \bar{\boldsymbol{g}}\right)\right). \tag{9}$$

4.2 Top-K Feature Selection as Tokenization

In DETR, the tokenization simply collapses the spatial dimensions of the feature map into a single dimension, resulting in $P_{\texttt{hor}} = WD/s^2$ and $P_{\texttt{ver}} = HD/s^2$ tokens for the horizontal and vertical radar feature maps, respectively. As a result, we have $P = (W + H)D/s^2$ multi-view radar tokens. It is known that the complexity of transformers grows quadratically with respect to the token length P. Here, we introduce a customized Top-K feature selection as tokenization, maintaining a low complexity for the RETR encoder and decoder: $\boldsymbol{H}_{\texttt{hor}} = \texttt{Selector}\left(\boldsymbol{Z}_{\texttt{hor}}\right) \in \mathbb{R}^{C \times K}$ and $\boldsymbol{H}_{\texttt{ver}} = \texttt{Selector}\left(\boldsymbol{Z}_{\texttt{ver}}\right) \in \mathbb{R}^{C \times K}$, where $K \ll \min\{WD/s^2, HD/s^2\}$. In this case, we shrink the multi-view radar tokens from $P = (W + H)D/s^2$ to $P = 2K$. For each radar frame, we consistently select the Top-K strongest features, which may originate from varying spatial locations depending on the specific radar frame. Consequently, the gradient propagates back through the selected K features to the backbone weights, irrelevant to their spatial locations.

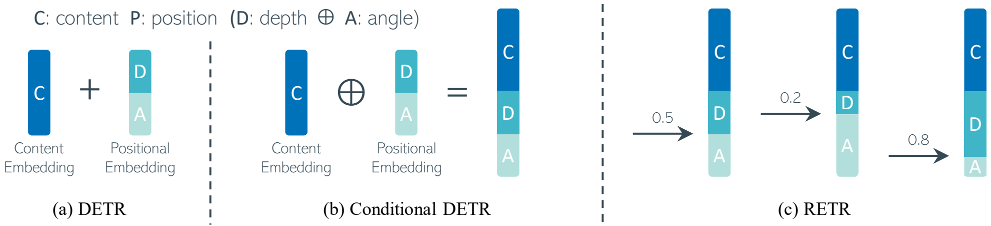

Figure 4: Schemes of positional encoding: (a) the sum operation in the original DETR; (b) the concatenation in Conditional DETR; and (c) TPE in RETR that allows for adjustable dimensions between depth and angular embeddings and promotes higher similarity scores for keys and queries with similar depth embeddings than those far apart in depth.

4.3 TPE: Tunable Positional Encoding

The TPE is built on the top of the concatenation operation between the content embedding $\boldsymbol{c}$ (either feature embedding $\boldsymbol{h}$ at the encoder or decoder embedding $\boldsymbol{q}$ at the decoder) and positional embedding $\boldsymbol{p}$ in the conditional DETR [21] (see Fig. 4 (b)):

$$\left(\boldsymbol{c}_{\texttt{que}} \oplus \boldsymbol{p}_{\texttt{que}}\right)^{\top} \left(\boldsymbol{c}_{\texttt{key}} \oplus \boldsymbol{p}_{\texttt{key}}\right) = \boldsymbol{c}_{\texttt{que}}^{\top}\boldsymbol{c}_{\texttt{key}} + \boldsymbol{p}_{\texttt{que}}^{\top}\boldsymbol{p}_{\texttt{key}}, \tag{10}$$

where $\oplus$ denotes concatenation, rather than the sum in DETR [3] (see Fig. 4 (a)):

$$\left(\boldsymbol{c}_{\texttt{que}} + \boldsymbol{p}_{\texttt{que}}\right)^{\top} \left(\boldsymbol{c}_{\texttt{key}} + \boldsymbol{p}_{\texttt{key}}\right) = \boldsymbol{c}_{\texttt{que}}^{\top}\boldsymbol{c}_{\texttt{key}} + \boldsymbol{c}_{\texttt{que}}^{\top}\boldsymbol{p}_{\texttt{key}} + \boldsymbol{p}_{\texttt{que}}^{\top}\boldsymbol{c}_{\texttt{key}} + \boldsymbol{p}_{\texttt{que}}^{\top}\boldsymbol{p}_{\texttt{key}}. \tag{11}$$

It is seen that Eq. 10 eliminates the cross terms between the content and positional embeddings in Eq. 11 and, allowing content/positional embeddings focus on their respective attention weights, contributes to faster training convergence [21].

In our case, the positional embedding is composed of a depth (y) axis and an angular (either azimuth x or elevation z) axis. As such, $\boldsymbol{p} = \boldsymbol{d} \oplus \boldsymbol{a}$ with $\boldsymbol{d}$ representing the depth positional embedding and $\boldsymbol{a}$ the angular positional embedding. Then expanding Eq. 10 with $\boldsymbol{p} = \boldsymbol{d} \oplus \boldsymbol{a}$ leads to

$$\left(\boldsymbol{c}_{\texttt{que}} \oplus \boldsymbol{d}_{\texttt{que}} \oplus \boldsymbol{a}_{\texttt{que}}\right)^{\top} \left(\boldsymbol{c}_{\texttt{key}} \oplus \boldsymbol{d}_{\texttt{key}} \oplus \boldsymbol{a}_{\texttt{key}}\right) = \boldsymbol{c}_{\texttt{que}}^{\top}\boldsymbol{c}_{\texttt{key}} + \boldsymbol{d}_{\texttt{que}}^{\top}\boldsymbol{d}_{\texttt{key}} + \boldsymbol{a}_{\texttt{que}}^{\top}\boldsymbol{a}_{\texttt{key}}. \tag{12}$$

In Eq. 12, we have the following observations:

1. $\boldsymbol{c}_{\texttt{que}}^{\top}\boldsymbol{c}_{\texttt{key}}$ reflects how similar the features in the key and query may appear;
2. Depth similarity $\boldsymbol{d}_{\texttt{que}}^{\top}\boldsymbol{d}_{\texttt{key}}$ remains consistent regardless of whether the key and query originate from the same radar view or different radar views;
3. Angular similarity $\boldsymbol{a}_{\texttt{que}}^{\top}\boldsymbol{a}_{\texttt{key}}$ can be a self-angular similarity (azimuth-to-azimuth or elevation-to-elevation) when the key and query are from the same radar view, or a cross-angular similarity (azimuth-to-elevation or elevation-to-azimuth) for different radar views.

Motivated by the above observations, we can promote higher similarity scores for keys and queries 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:

$$d_{\texttt{dep}} = \alpha d_{\texttt{pos}}, \quad d_{\texttt{ang}} = (1-\alpha) d_{\texttt{pos}} \quad \rightarrow \quad d_{\texttt{dep}} + d_{\texttt{ang}} = d_{\texttt{pos}}, \tag{13}$$

where the tunable dimension ratio α is in the interval $[0, 1]$. As illustrated in Fig. 4 (c), when $\alpha = 0.5$, the positional embedding is equivalent to that used in conditional DETR. When α approaches 0, the depth positional embedding is minimized, making the depth similarity $\boldsymbol{d}_{\texttt{que}}^{\top}\boldsymbol{d}_{\texttt{key}}$ negligible in Eq. 12. Conversely, as α approaches 1, the depth positional embedding dimension increases, and so does the importance of the depth similarity in Eq. 12.

We implement our TPE with a fixed sine/cosine positional encoding along the depth and angular (azimuth or elevation) dimension. For an even depth/angular positional dimension, we have

$$\boldsymbol{d}_{2i} = \sin(\mathrm{p}_{\texttt{dep}}/\tau^{2i/d_{\texttt{dep}}}), \quad \boldsymbol{d}_{2i+1} = \cos(\mathrm{p}_{\texttt{dep}}/\tau^{2i/d_{\texttt{dep}}}), \quad i = 0, 1, \cdots, d_{\texttt{dep}}/2 - 1, \tag{14}$$

$$\boldsymbol{a}_{2i} = \sin(\mathrm{p}_{\texttt{ang}}/\tau^{2i/d_{\texttt{ang}}}), \quad \boldsymbol{a}_{2i+1} = \cos(\mathrm{p}_{\texttt{ang}}/\tau^{2i/d_{\texttt{ang}}}), \quad i = 0, 1, \cdots, d_{\texttt{ang}}/2 - 1, \tag{15}$$

where $\mathrm{p}_{\texttt{dep/ang}}$ and $d_{\texttt{dep/ang}}$ are the position index and dimension for the depth and angular axes, respectively, i is the (even/odd) element index, and $\tau = 10000$ is a temperature. By adjusting the ratio α in Eq. 12, we change the dimensions of the depth $\boldsymbol{d}$ in Eq. 14 and the angular $\boldsymbol{a}$ in Eq. 15, while keeping the total positional dimension of $\boldsymbol{p} = \boldsymbol{d} \oplus \boldsymbol{a}$ constant. We show the visualization of TPE in Appendix C.

4.4 Tri-Plane Set-Prediction Loss

DETR calculates a matching cost matrix with each element constructed from 1) a classification cost $\mathcal{L}_{\texttt{class}}$ and 2) a BBox loss between one of N predictions $\widehat{\boldsymbol{b}}$ and one of ground truth BBoxes $\boldsymbol{b}$ (including the "no object" class). The BBox loss is a weighted combination of the generalized intersection over union (GIoU) loss $\mathcal{L}_{\texttt{GIoU}}$ [28] and the ℓ_1 loss $\mathcal{L}_{\mathrm{L}_1}$:

$$\mathcal{L}_{\texttt{box}}(\boldsymbol{b}, \widehat{\boldsymbol{b}}) = \lambda_{\texttt{GIoU}} \mathcal{L}_{\texttt{GIoU}}(\boldsymbol{b}, \widehat{\boldsymbol{b}}) + \lambda_{\mathrm{L}_1} \mathcal{L}_{\mathrm{L}_1}(\boldsymbol{b}, \widehat{\boldsymbol{b}}), \tag{16}$$

where λ_* denotes the weight. Over the permutation set $\mathfrak{S}_N$ between N predictions and ground truth objects, the Hungarian algorithm [12] is applied with the matching cost matrix to find the optimal assignment $\sigma^* \in \mathfrak{S}_N$ of predictions to ground truth. Given σ^* , the loss is computed only for the matched pairs and is referred to as the set-prediction loss.

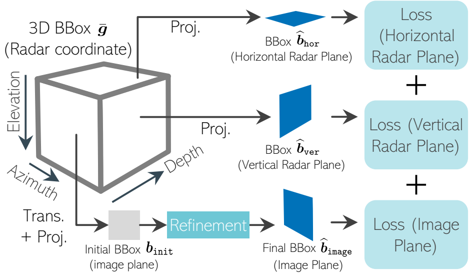
3D BBox $\bar{\boldsymbol{g}}$
(Radar coordinate)
Elevation
Azimuth
Depth
Proj.
BBox $\widehat{\boldsymbol{b}}_{\texttt{hor}}$
(Horizontal Radar Plane)
Loss (Horizontal Radar Plane)
+
Proj.
BBox $\widehat{\boldsymbol{b}}_{\texttt{ver}}$
(Vertical Radar Plane)
Loss (Vertical Radar Plane)
+
Trans. + Proj.
Initial BBox $\boldsymbol{b}_{\texttt{init}}$
(image plane)
Refinement
Final BBox $\widehat{\boldsymbol{b}}_{\texttt{image}}$
(Image Plane)
Loss (Image Plane)

Figure 5: Tri-Plane BBox loss.

Since RETR predicts 3D BBoxes $\bar{\boldsymbol{g}}$ in the 3D radar coordinate and maps them into the 2D image plane, we propose to enhance the above Hungarian match cost matrix using a *Tri-Plane BBox Loss* from both the radar coordinate and image plane. This is illustrated in Fig. 5, where a 3D BBox $\bar{\boldsymbol{g}}$ in the radar coordinate is projected onto 1) the 2D horizontal radar plane as $\widehat{\boldsymbol{b}}_{\texttt{hor}} = \texttt{proj}_{\texttt{hor}}(\bar{\boldsymbol{g}})$ (the top branch); 2) the 2D vertical radar plane as $\widehat{\boldsymbol{b}}_{\texttt{ver}} = \texttt{proj}_{\texttt{ver}}(\bar{\boldsymbol{g}})$ (the middle branch); and 3) the 2D image plane as $\widehat{\boldsymbol{b}}_{\texttt{image}}$ of Eq. 9 (the bottom branch). The tri-plane BBox loss $\mathcal{L}^{\texttt{tri}}_{\texttt{box}}$ sums up 2D BBox losses over all three planes using Eq. 16:

$$\mathcal{L}^{\texttt{tri}}_{\texttt{box}} = \mathcal{L}_{\texttt{box}}\left(\boldsymbol{b}_{\texttt{hor}}, \widehat{\boldsymbol{b}}_{\texttt{hor}}\right) + \mathcal{L}_{\texttt{box}}\left(\boldsymbol{b}_{\texttt{ver}}, \widehat{\boldsymbol{b}}_{\texttt{ver}}\right) + \mathcal{L}_{\texttt{box}}\left(\boldsymbol{b}_{\texttt{image}}, \widehat{\boldsymbol{b}}_{\texttt{image}}\right). \tag{17}$$

RETR finds the optimal assignment $\sigma^*_{\texttt{tri}}$ using the matching cost with 1) the original classification cost $\mathcal{L}_{\texttt{class}}$ and 2) the tri-plane BBox loss $\mathcal{L}^{\texttt{tri}}_{\texttt{box}}$. The resulting set-prediction loss using $\sigma^*_{\texttt{tri}}$ is referred to as the tri-plane set-prediction loss.

4.5 Learnable Radar-to-Camera Coordinate Transformation

The rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ in the radar-to-camera transformation of Eq. 7 can be calibrated in advance. However, this calibration process may be accurate only for a limited interval of depth and angles. Instead of relying on the calibrated transformation, we introduce a learnable transformation via a reparameterization on $\boldsymbol{R}$ while keeping it orthonormal. To this end, we need to ensure that the learnable $\widehat{\boldsymbol{R}}$ resides in the 3D special orthogonal group $\mathcal{SO}(3)$. Considering that $\mathcal{SO}(3)$ is a special case of a Lie group, one of the differentiable manifolds, we can firstly map a 3D vector $\boldsymbol{\omega} = \{\omega_x, \omega_y, \omega_z\}^\top \in \mathbb{R}^3$ to Lie algebra $\mathfrak{so}(3)$ using the projection $[\cdot] : \mathbb{R}^3 \to \mathfrak{so}(3)$. And then we apply the exponential map $\texttt{exp} : \mathfrak{so}(3) \to \mathcal{SO}(3)$ that maps $[\boldsymbol{\omega}]$ into the nearest point in $\mathcal{SO}(3)$ such that the resulting $\texttt{exp}([\boldsymbol{\omega}])$ resides on $\mathcal{SO}(3)$ and satisfies the orthonormal structure [13, 33]. This leads to the following reparameterization of $\widehat{\boldsymbol{R}}$ in terms of $\boldsymbol{\omega}$:

$$\widehat{\boldsymbol{R}} \approx \texttt{exp}([\boldsymbol{\omega}]) = \boldsymbol{I} + \frac{\sin\phi}{\phi}[\boldsymbol{\omega}] + \frac{1-\cos\phi}{\phi^2}[\boldsymbol{\omega}]^2, \text{ s.t. } [\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{18}$$

where $\phi = \|\boldsymbol{\omega}\|$ is the ℓ_2 norm, With the above reparameterization, the learnable radar-to-camera coordinate transformation in Eq. 7 reduces to learn the vector $\boldsymbol{\omega}$ and the translation vector $\boldsymbol{t}$.

Table 1: Main results of object detection in the image plane under "P2S1" of MMVR. The top section shows results from conventional models, while the bottom section presents RETR results.

Model	Dim	Input	BBox Loss	AP	AP$_{50}$	AP$_{75}$	AR$_{1}$	AR$_{10}$
RFMask	2D	H, V	$\mathtt{H}+\mathtt{I}$	31.37	61.50	27.48	33.23	38.41
DETR	2D	H	$\mathtt{H}+\mathtt{I}$	29.38	62.31	25.35	31.32	43.06
DETR (Top-K)	2D	H, V	$\mathtt{H}+\mathtt{I}$	39.71	82.74	33.29	38.98	52.81
RETR (TPE@Dec.)	3D	H, V	$\mathtt{H}+\mathtt{V}+\mathtt{I}$	45.94	81.99	44.04	42.03	57.38
RETR	3D	H, V	$\mathtt{H}+\mathtt{V}+\mathtt{I}$	46.75	83.80	46.06	42.19	57.39

Table 2: Main results of object detection in the image plane under "WALK" of HIBER. The notation follows the same format as Table 1.

Model	Dim	Input	BBox Loss	AP	AP$_{50}$	AP$_{75}$	AR$_{1}$	AR$_{10}$
RFMask	2D	H, V	$\mathtt{H}+\mathtt{I}$	17.77	52.46	6.78	32.71	32.71
DETR	2D	H	$\mathtt{H}+\mathtt{I}$	14.45	47.33	4.25	28.64	28.64
DETR (Top-K)	2D	H, V	$\mathtt{H}+\mathtt{I}$	14.35	48.94	5.50	28.78	28.78
RETR (TPE@Dec.)	3D	H, V	$\mathtt{H}+\mathtt{V}+\mathtt{I}$	20.18	52.53	7.32	32.91	32.91
RETR	3D	H, V	$\mathtt{H}+\mathtt{V}+\mathtt{I}$	22.09	59.83	10.99	35.16	35.16

5 Experiments

5.1 Setup

Datasets: We evaluate performance over two open indoor radar perception datasets: MMVR[4] [26] and HIBER[5] [38]. MMVR includes multi-view radar heatmaps collected from over 20 human subjects across 6 rooms over a span of 9 days. In our implementation, we utilize data from **Protocol 2** (P2) which includes 237.9K data frames capturing both single and multiple human subjects in diverse activities such as walking, sitting, stretching, and writing on the board. For the training-validation-test split, we follow the data split **S1** as defined in MMVR.

HIBER, partially released, includes multi-view radar heatmaps from 10 human subjects in a single room but from different angles with two data splits: 1) "WALK", consisting of $73.5K$ data frames with one subject (Section 5.2); and 2) "MULTI", consisting of $70.8K$ radar frames with multiple (2) human subjects walking in the room (Appendix G). More dataset details can be found in Appendix E.

Implementation: We consider RFMask [38] and DETR [3] as baseline methods. Since RFMask and DETR originally compute the BBox loss only in the 2D horizontal ($\mathtt{H}$) radar plane and the 2D image ($\mathtt{I}$) plane, respectively, we enhance both methods with a unified bi-plane BBox loss ($\mathtt{H}+\mathtt{I}$). We also introduce a DETR variant with top-K feature selection, allowing it to take features from both horizontal ($\mathtt{H}$) and vertical ($\mathtt{V}$) heatmaps as input. For RETR, we set $K = 256$ for the top-K selection, the positional embedding dimension to $d_{\text{pos}} = 256$, and a tunable dimension ratio at $\alpha = 0.6$. We include one variant that only employs the TPE at the decoder (TPE@Dec.). More hyper-parameter settings can be found in Appendix E.

Metrics: For object detection, we adopt average precision (AP) at two IoU thresholds of 0.5 ($\mathtt{AP}_{50}$) and 0.75 ($\mathtt{AP}_{75}$) and its mean ($\mathtt{AP}$) over thresholds $[0.5 : 0.05 : 0.95]$. We also consider average recall (AR) when it is restricted to making only one detection ($\mathtt{AR}_{1}$) or up to 10 detections ($\mathtt{AR}_{10}$) per image. For segmentation, we report the average $\mathtt{IoU}$ value between the predictive and ground truth masks. Detailed metric definitions can be found in Appendix F.

5.2 Main Results

MMVR: Table 1 shows the main results on the MMVR dataset under "P2S1". Compared with RFMask, DETR with a single horizontal radar view does not show performance improvement. By

[4] https://zenodo.org/records/12611978

[5] https://github.com/Intelligent-Perception-Lab/HIBER

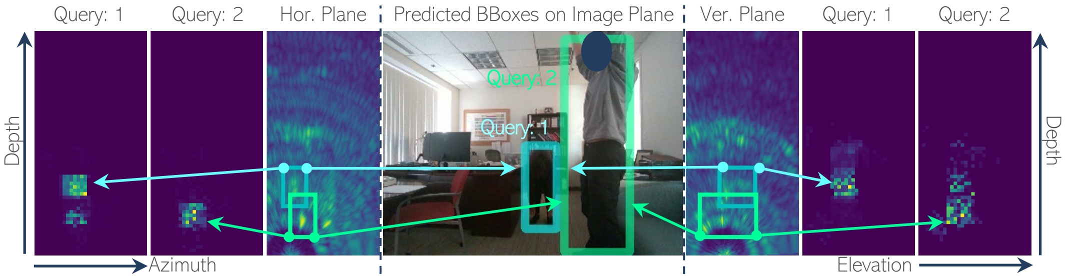

Figure 6: Visualization of cross-attention map between predicted BBoxes and multi-view radar features. BBoxes with the same color correspond to the same subject.

Ratio α	0.2	0.4	0.6	0.8
AP	44.35	45.16	46.75	43.81
AR1	42.04	41.60	42.19	41.89

(a) A dimension **ratio** $\alpha = 0.6$ of TPE achieves the best.

LT	AP	AR_1
-	42.97	40.20
✓	46.75	42.19

(b) **Learnable Transformation** (LT) may replace calibration.

Loss	AP	AR_1
H + I	42.18	39.39
H + V + I	46.75	42.19

(c) **Tri-plane loss** enhances object detection.

Table 3: **Ablation studies** under "P2S1" on MMVR.

just adding the vertical radar view at the input, DETR with top-K selection exhibits a noticeable performance improvement over RFMask. Built upon DETR (Top-K), RETR (TPE@Dec.) implements two enhancements: 1) TPE at the decoder and 2) tri-plane BBox loss, resulting in further improvements with a gain of 6.23 in AP, 10.75 in AP_{75}, and 4.57 in AR_{10}, highlighting the importance of TPE and supervision at the vertical radar view. By further incorporating TPE at the encoder, the full version of RETR achieves an impressive performance improvement over RFMask, demonstrating increases of 15.38 in AP, 22.30 in AP_{50}, and 18.58 in AP_{75}, respectively. The results under "P2S2" on MMVR can be seen in Appendix G.

HIBER: Table 2 presents the main results on the HIBER dataset under "WALK". Similar to Table 1, we observe a similar trend of performance improvement from DETR to RETR variants. Numerically, we see increases of 4.32 in AP, 7.37 in AP_{50}, and 4.21 in AP_{75}, when directly comparing RETR to RFMask. These performance improvements are smaller compared with those in Table 1. This is potentially because the HIBER data under "WALK" predominantly involves walking, where RFMask's fixed-height vertical proposals may work fine. In contrast, MMVR under "P2" includes more diverse activities such as sitting, leading to likely overestimated vertical proposals for RFMask and thus greater improvements in MMVR than HIBER. The results under "MULTI" on HIBER can be seen in Appendix G.

Visualization of Cross-Attention Map: Fig. 6 presents the cross-attention map at the last decoder layer between predicted BBoxes (via object queries) and multi-view radar features. RETR accurately predicts the subject in the background of the image plane (middle panel) with a forward-bending posture (Query 1). The cross-attention maps of Query 1, with respect to horizontal (left) and vertical (right) radar features, highlight areas with features contributing the most to Query 1. These contributing areas in the vertical plane are more stretched along the depth axis compared with those in the horizontal plane. Notably, the contributing areas from the two views share similar depth intervals. For Query 2 which identifies the subject in the foreground, the cross-attention maps shift its focus to contributing areas at closer depth compared with those for Query 1, indicating an effective 3D spatial embedding of object queries at the RETR output. We provide more visualizations in Appendix H.

Limitation: We present failure cases in Fig. 15 of Appendix H. Predicting arm positions remains challenging, suggesting that RETR may not focus its attention on regions with weak radar reflections. Moreover, multi-path reflections from the ground, ceiling, and other strong scatterers (e.g., metal) can cause (first-order or second-order) ghost targets and elevate the noise floor. Traditional signal processing techniques can mitigate these effects but require access to raw radar data. Alternatively, ghost targets can be labeled in the multi-view radar heatmaps, though this can be time-consuming and costly. One can then extend RETR to classify output queries to one of $\{\emptyset, person, ghost\}$, alongside regressing queries to the BBox parameters.

5.3 Ablation Studies

We report ablation studies with RETR under "P2S1" on MMVR. Further results of ablation studies can be seen in Appendix G.

Tunable Dimension Ratio α: Table 3a presents the ablation study of the tunable dimension ratio α and its impact on the object detection performance in terms of $\texttt{AP}_{50}$ (primary vertical axis) and AP and $\texttt{AP}_{75}$ (secondary vertical axis). The results indicate that $\alpha = 0.6$ yields the best performance. The detection performance gradually decreases as α approaches to 0 and 1.

Learnable Transformation (LT): To evaluate the effectiveness of the Learnable Transformation in Section 4.5, we compare AP and $\texttt{AR}_1$ metrics of RETRs with and without LT. The results in Table 3b indicate that it is possible to incorporate the radar-to-camera geometry into the end-to-end radar perception pipeline without the need for a cumbersome calibration step, while still achieving comparable perception performance.

Tri-Plane Loss for RETR: Table 3c compares RETR with a bi-plane BBox loss (horizontal radar plane and image plane) to that with the tri-plane loss (including the vertical radar plane). The results highlight the necessity of accounting for the vertical BBox loss and the importance of leveraging features from the vertical radar heatmap, leading to a performance improvement of 4.47 in AP.

6 Conclusion

In this paper, we introduced RETR, extending DETR to the multi-view radar perception with carefully designed modifications such as depth-prioritized feature similarity via TPE, a tri-plane loss from radar and camera coordinates, and a learnable radar-to-camera transformation. Experimental results over two radar datasets and comprehensive ablation studies demonstrate that RETR significantly outperforms both RFMask and DETR baseline methods.

Broader Impacts: Indoor radar perception technologies, including RETR, offer a wide range of social applications in navigating and monitoring subjects such as the elderly, infants, robots, and humanoids, enhancing safety and energy efficiency while preserving privacy. However, it is crucial that perception results remain secure and private to prevent misuse in inferring subject attributes such as gender, size, and height. These technologies could potentially be used to advance indoor surveillance without individuals' acknowledgment or consent.

References

[1] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. *ACM Trans. Graph.*, 34(6), 2015. URL https://doi.org/10.1145/2816795.2818072.

[2] Sizhe An, Yin Li, and Umit Ogras. mRI: Multi-modal 3D human pose estimation dataset using mmWave, RGB-D, and inertial sensors. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27414–27426, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/af9c9c6d2da701da5a0acf91ec217815-Paper-Datasets_and_Benchmarks.pdf.

[3] 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 (ECCV)*, pp. 213–229, 2020. URL https://doi.org/10.1007/978-3-030-58452-8_13.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF*

International Conference on Computer Vision (ICCV), pp. 9630–9640, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00951.

[5] Fangqiang Ding, Xiangyu Wen, Yunzhou Zhu, Yiming Li, and Chris Xiaoxuan Lu. RadarOcc: Robust 3D occupancy prediction with 4D imaging radar. *arXiv:2405.14014*, 2024. URL https://arxiv.org/abs/2405.14014.

[6] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. URL https://doi.org/10.1007/s11263-009-0275-4.

[7] Sevgi Zubeyde Gurbuz and Moeness G. Amin. Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring. *IEEE Signal Processing Magazine*, 36(4):16–28, 2019. URL https://doi.org/10.1109/MSP.2018.2890128.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. URL https://doi.org/10.1109/CVPR.2016.90.

[9] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4): 814–830, 2016. URL https://doi.org/10.1109/TPAMI.2015.2465908.

[10] Zhengdong Hu, Yifan Sun, Jingdong Wang, and Yi Yang. DAC-DETR: Divide the attention layers and conquer. In *Advances in Neural Information Processing Systems*, volume 36, pp. 75189–75200, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/edd0d433f8a1a51aa11237a6543fc280-Paper-Conference.pdf.

[11] Seung-Hyun Kong, Dong-Hee Paek, and Sangyeong Lee. RTNH+: Enhanced 4D radar object detection network using two-level preprocessing and vertical encoding. *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2024. URL https://10.1109/TIV.2024.3428696.

[12] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. URL https://doi.org/10.1007/978-3-540-68279-0_2.

[13] John. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2003. URL https://doi.org/10.1007/978-1-4419-9982-5.

[14] Shih-Po Lee, Niraj Prakash Kini, Wen-Hsiao Peng, Ching-Wen Ma, and Jenq-Neng Hwang. HuPR: A benchmark for human pose estimation using millimeter wave radar. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5715–5724, 2023. URL https://doi.org/10.1109/WACV56688.2023.00567.

[15] Peizhao Li, Pu Wang, Karl Berntorp, and Hongfu Liu. Exploiting temporal relations on radar perception for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17050–17059, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Li_Exploiting_Temporal_Relations_on_Radar_Perception_for_Autonomous_Driving_CVPR_2022_paper.pdf.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. URL https://doi.org/10.1109/ICCV.2017.324.

[17] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=oMI9PjOb9Jl.

[18] Shilong Liu, Tianhe Ren, Jiayu Chen, Zhaoyang Zeng, Hao Zhang, Feng Li, Hongyang Li, Jun Huang, Hang Su, Jun Zhu, and Lei Zhang. Detection transformer with stable matching. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6468–6477, 2023. URL https://doi.org/10.1109/ICCV51070.2023.00597.

[19] Yang Liu, Feng Wang, Naiyan Wang, and Zhao-Xiang Zhang. Echoes beyond points: Unleashing the power of raw radar data in multi-modality fusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53964–53982, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a8f7f12b29d9b8c227785f6b529f63b7-Paper-Conference.pdf.

[20] Chris Xiaoxuan Lu, Muhamad Risqi U. Saputra, Peijun Zhao, Yasin Almalioglu, Pedro P. B. de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. milliEgo: single-chip mmwave radar aided egomotion estimation via deep sensor fusion. In *The 18th Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 109–122, 2020. URL https://doi.org/10.1145/3384419.3430776.

[21] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3631–3640, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00363.

[22] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision (3DV)*, pp. 565–571, 2016. URL https://10.1109/3DV.2016.79.

[23] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15651–15660, 2021. URL https://doi.org/10.1109/ICCV48922.2021.01538.

[24] Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. K-Radar: 4D radar object detection for autonomous driving in various weather conditions. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3819–3829, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/185fdf627eaae2abab36205dcd19b817-Paper-Datasets_and_Benchmarks.pdf.

[25] Yifan Pu, Weicong Liang, Yiduo Hao, Yuhui Yuan, Yukang Yang, Chao Zhang, Han Hu, and Gao Huang. Rank-DETR for high quality object detection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16100–16113, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34074479ee2186a9f236b8fd03635372-Paper-Conference.pdf.

[26] M. Mahbubur Rahman, Ryoma Yataka, Sorachi Kato, Pu (Perry) Wang, Peizhao Li, Adriano Cardace, and Petros Boufounos. MMVR: Millimeter-wave multi-view radar dataset and benchmark for indoor perception. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/10205.pdf.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. URL https://doi.org/10.1109/TPAMI.2016.2577031.

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio 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)*, pp. 658–666, 2019. URL https://doi.org/10.1109/CVPR.2019.00075.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015. URL https://10.1007/978-3-319-24574-4_28.

[30] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. *IEEE Sensors Journal*, 20(17): 10032–10044, 2020. URL https://doi.org/10.1109/JSEN.2020.2991741.

[31] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. RadHAR: Human activity recognition from point clouds generated through a millimeter-wave radar. In *The 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, mmNets '19, pp. 51–56, 2019. URL https://doi.org/10.1145/3349624.3356768.

[32] Mikael Skog, Oleksandr Kotlyar, Vladimír Kubelka, and Martin Magnusson. Human detection from 4D radar data in low-visibility field conditions. *arXiv:2404.05307*, 2024. URL https://arxiv.org/abs/2404.05307.

[33] Joan Solá, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv:1812.01537*, 2021. URL https://arxiv.org/abs/1812.01537.

[34] Shunqiao Sun, Athina P. Petropulu, and H. Vincent Poor. MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. *IEEE Signal Processing Magazine*, 37(4):98–117, 2020. URL https://10.1109/MSP.2020.2978507.

[35] Baptist Vandersmissen, Nicolas Knudde, Azarakhsh Jalalvand, Ivo Couckuyt, André Bourdoux, Wesley De Neve, and Tom Dhaene. Indoor person identification using a low-power FMCW radar. *IEEE Transactions on Geoscience and Remote Sensing*, 56(7):3941–3952, 2018. URL https://doi.org/10.1109/TGRS.2018.2816812.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[37] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for transformer-based detector. In *AAAI Conference on Artificial Intelligence*, volume 36, pp. 2567–2575, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20158.

[38] Zhi Wu, Dongheng Zhang, Chunyang Xie, Cong Yu, Jinbo Chen, Yang Hu, and Yan Chen. RFMask: A simple baseline for human silhouette segmentation with radio signals. *IEEE Transactions on Multimedia*, 25:4730–4741, 2023. URL https://doi.org/10.1109/TMM.2022.3181455.

[39] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. mmMesh: towards 3D real-time dynamic human mesh construction using millimeter-wave. In *MobiSys*, pp. 269–282, 2021. URL https://doi.org/10.1145/3458864.3467679.

[40] Jianfei Yang, He Huang, Yunjiao Zhou, Xinyan Chen, Yuecong Xu, Shenghai Yuan, Han Zou, Chris Xiaoxuan Lu, and Lihua Xie. MM-Fi: Multi-modal non-intrusive 4D human dataset for versatile wireless sensing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18756–18768, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3baf7a39d07e9f4f1e258a412df94521-Paper-Datasets_and_Benchmarks.pdf.

[41] Ryoma Yataka, Pu Wang, Petros Boufounos, and Ryuhei Takahashi. SIRA: Scalable inter-frame relation and association for radar perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15024–15034, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Yataka_SIRA_Scalable_Inter-frame_Relation_and_Association_for_Radar_Perception_CVPR_2024_paper.pdf.

[42] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S. Jaakkola, and Matt T. Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning (ICML)*, volume 70, pp. 4100–4109, 2017. URL https://proceedings.mlr.press/v70/zhao17d.html.

[43] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7356–7365, 2018. URL https://doi.org/10.1109/CVPR.2018.00768.

[44] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. RF-based 3D skeletons. In *The 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, pp. 267–281, 2018. URL https://doi.org/10.1145/3230543.3230579.

[45] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. Human tracking and identification through a millimeter wave radar. *Ad Hoc Networks*, 116, 2021. URL https://doi.org/10.1016/j.adhoc.2021.102475.

[46] Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. Cubelearn: End-to-end learning for human motion recognition from raw mmWave radar signals. *IEEE Internet of Things Journal*, 10(12):10236–10249, 2023. URL https://doi.org/10.1109/JIOT.2023.3237494.

[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=gZ9hCDWe6ke.

A Details of RETR Architecture

Transformer Encoder and Decoder: Fig. 7 illustrates the transformer encoder and decoder used in RETR. In the original DETR implementation, the image features from the CNN backbone are given in input to the transformer encoder, with spatial positional embeddings added to the queries and keys at each multi-head self-attention layer of the encoder. On the other hand, RETR extracts features from a shared-weight backbone for both horizontal and vertical views and obtains them as $\{P^{\mathrm{h}}_1, \cdots, P^{\mathrm{h}}_{WD/s^2}, P^{\mathrm{h}}_1, \cdots, P^{\mathrm{v}}_{HD/s^2}\}$. At this time, the positional encoding (TPE) is concatenated with the features (content). Subsequently, Top-K selection is applied to extract the most relevant features and reduce time and space complexity (i.e., $P^{\mathrm{h}}_5, P^{\mathrm{h}}_6, P^{\mathrm{v}}_6$ and P^{v}_7 in the left figure). These Top-K features from the horizontal and vertical views are concatenated to compose a single sequence of tokens, which are then fed to the transformer encoder. The encoder consists of a stack of multi-head self-attention layers, that allow for the consideration of correlations between the two views. The multi-head attention is simply the concatenation of M single attention heads followed by a projection layer L to regain the initial dimensionality. The common practice [36] is to use residual connections, `dropout`, and layer normalization:

$$\mathtt{mhAtt} = \mathtt{layernorm}\left(\mathtt{Que}\left(\boldsymbol{H}\right) + \mathtt{dropout}\left(L\tilde{\boldsymbol{H}}\right)\right), \tag{19}$$

$$\begin{aligned}\tilde{\boldsymbol{H}} =& \mathtt{Att}\left(\mathtt{Que}\left(\boldsymbol{H}\right), \mathtt{Key}\left(\boldsymbol{H}\right), \mathtt{Val}\left(\boldsymbol{H}\right), \boldsymbol{W}_1\right) \oplus \cdots \\ &\oplus \mathtt{Att}\left(\mathtt{Que}\left(\boldsymbol{H}\right), \mathtt{Key}\left(\boldsymbol{H}\right), \mathtt{Val}\left(\boldsymbol{H}\right), \boldsymbol{W}_M\right),\end{aligned} \tag{20}$$

where $\oplus$ is concatenation along the channel axis, and $\boldsymbol{W}$ denotes the weight tensor of attention.

The decoder receives the decoder embeddings, which we initially set to zero and concatenated with the object queries, and encoder memory (i.e. the output sequence of the encoder transformer), generating refined embeddings through multiple multi-head self-attention and cross-attention layers. In particular, the cross-attention layer utilizes the encoder memory to produce Keys and Values, which correlate with the Queries to produce the Refined Queries. In right figure of Fig. 7, the decoder embeddings which concatenated with the object queries are first input into the self-attention, and the output is then passed through a normalization layer. At this point, the values are added using a residual structure. Next, cross-attention between the encoder memory, used as the key, and the decoder embeddings is calculated. Similarly, a residual structure is employed as in the self-attention. This entire sequence is repeated $L_{\mathtt{cross}}$ times to obtain the final decoder embeddings.

Computational Complexity Following the computational complexity notation used in the DETR paper, every self-attention mechanism in the encoder has a complexity of $\mathcal{O}(d^2 2K + d(2K)^2)$ where

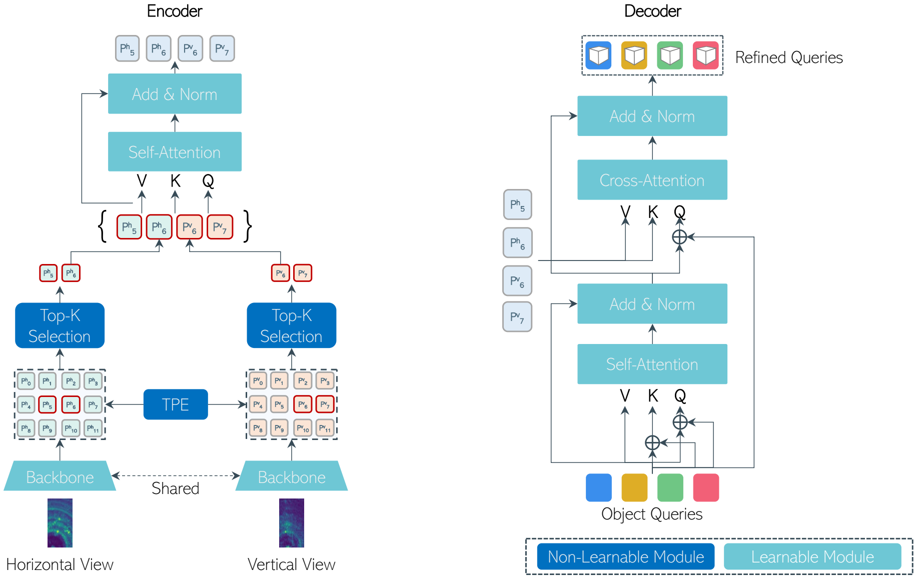

Figure 7: Illustration of (left) encoder and (right) decoder of RETR.

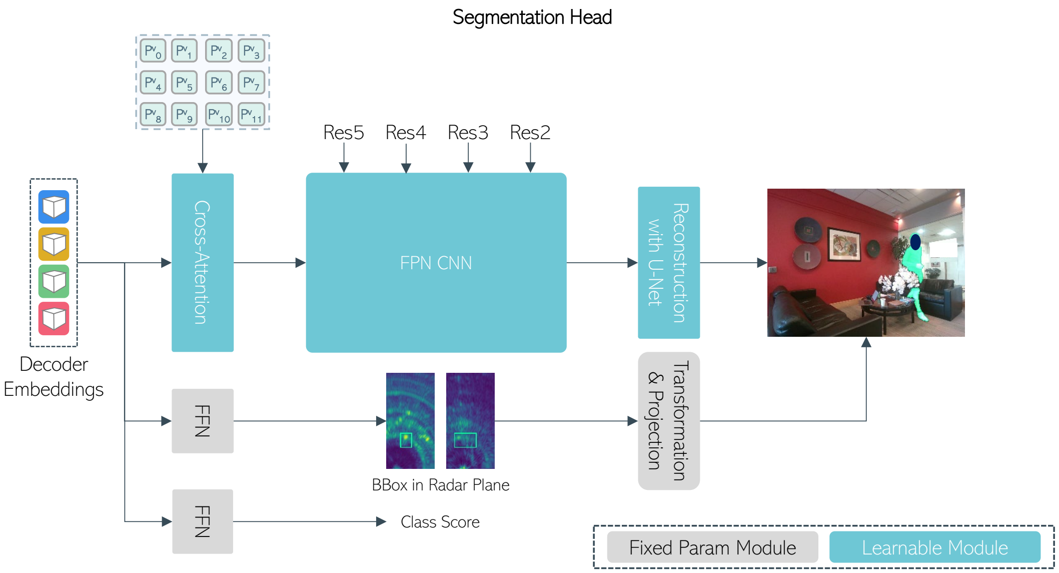

Figure 8: Illustration of segmentation head.

d is the embedding dimension and K is the number of selected features from the Top-K selection. The cost of computing a single query/key/value embedding is $\mathcal{O}(d'd)$ (with $d = Md'$ where M denotes the number of attention heads and d' the dimension in each head), while the cost of computing the attention weights for one head is $\mathcal{O}(d'(2K)^2)$. Other computations may be negligible. In the decoder, each self-attention mechanism has a complexity of $\mathcal{O}(d^2N + dN^2)$ where N is the number of queries, and the cross-attention between query and multi-view radar features has a complexity of $\mathcal{O}(d^2(N + 2K) + d2NK)$. In conclusion, the overall complexity of our RETR model is

$$\mathcal{O}(4d^2K + 4dK^2 + 2d^2N + dN^2 + 2dNK). \tag{21}$$

B Segmentation

Architecture of Segmentation Head: The original DETR is naturally extended by adding a segmentation head on top of the decoder outputs. Following this extension, our RETR enables segmentation by adding an architecture with a similar structure. Fig. 8 illustrates the segmentation architecture we implemented, consisting of a cross-attention layer, a feature pyramid network (FPN)-style CNN, and final light U-Net [29]. Given a single refined query, we use a cross-attention layer to generate attention heatmaps for each object at a low resolution. For the backbone output used in cross-attention, we utilized features extracted from the vertical heatmap, enhancing robustness to the height of the human. To increase the resolution of the mask, an FPN-style architecture is employed which also exploits the low-level backbone features at different layers (from 5 to 2) to generate some coarse segmentation masks. Since the FPN module is also responsible for lifting features from the radar view to the image plane, it does not have enough capacity to generate fine-grained segmentation masks. Thereby, we also add a very light U-Net to further refine the previously generated masks. It is important to note that our model, differently from the original DETR implementation, predicts a single binary mask for each query. Indeed, we exploit for each query the corresponding bounding box prediction in the radar plane, apply the Radar-to-Camera transformation and the 3D-to-2D image projection, to obtain the bounding box in the image plane. This bounding box is finally used to extract the corresponding portion from the ground truth segmentation mask, which is employed to supervise the segmentation prediction for the same query. As a loss function, we adopt the DICE/F-1 loss [22] and focal loss [16].

Training: We note that the segmentation head can be trained at the same time as the BBox head in an end-to-end manner, or we can first train the detection head and then freeze all weights and train only the segmentation head in a two-step process. We followed the original DETR and employed

Table 4: Segmentation results under “P2S1” on MMVR.

Model	Dim	Input	BBox Loss	IoU
RFMask	2D	H, V	H + I	65.30
DETR	2D	H	H + I	70.15
DETR (Top-K)	2D	H, V	H + I	75.76
RETR (TPE@Dec.)	3D	H, V	H + V + I	76.16
RETR	3D	H, V	H + V + I	77.07

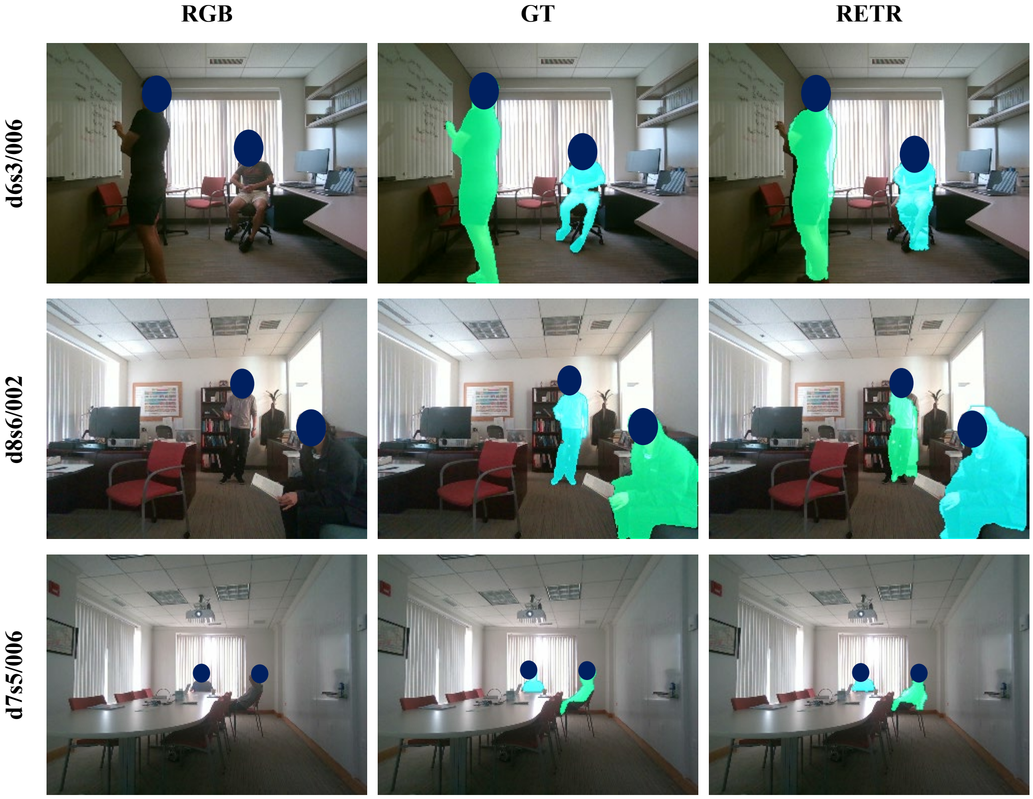

Figure 9: Visualization of segmentation results.

the latter strategy. During prediction, we filter out the detection with a confidence below 50%, then compute the per-pixel argmax to produce the final binary segmentation mask.

Main Results: We report quantitative results the segmentation tasks in Table 4. From this table, RETR (which combines all our contributions) achieves 77.07@IoU, which is a significant performance improvement over the conventional RFmask with a gap of 11.77@IoU. In addition, we point out how the DETR (Top-K) version (row 3) alone is able to increase the performance by almost 5%. We visualize the segmentation results in Fig. 9. Each row represents the data segment number in MMVR. It can be observed that RETR captures the shape of people with high fidelity. Notably, the results for d6s3 and d8s6 demonstrate that we are able to segment even complex postures, including sitting positions. Additionally, as shown in d7s5, RETR accurately estimates positions even when subjects are sitting far from the radar, such as at the back of the room. These results indicate that RETR can be easily extended from a detector to a segmentation model by adding a segmentation head, and it can accurately estimate masks. For more visualizations, including comparison with RFMask and failure cases, see Appdendix H.

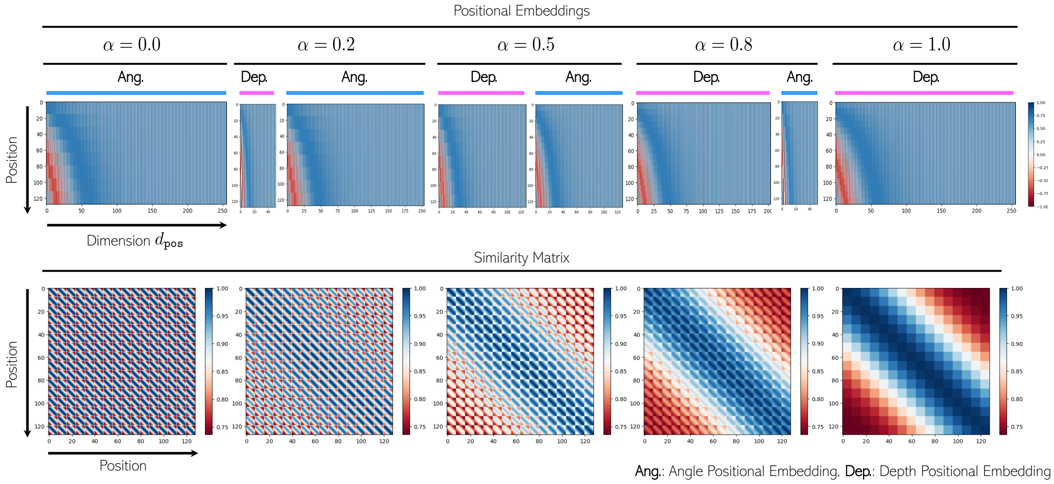

Figure 10: Visualization of TPE: (top row) positional embeddings for each axis; depth and angle, (bottom row) similarity through positions with dot product of positional embeddings. Each column denotes the α ($\alpha = 0.0, 0.2, 0.5, 0.8, 1.0$). The blue color represents the large value, and red color represents the low value. The range is $[-1, 1]$.

C Visualization of TPE

We visualize the positional embedding of each axis to observe the TPE. We calculated the positional embedding according to Eq. 14 and Eq. 15, and visualized each axis as a separate figure with several value of α ($\alpha = 0.0, 0.2, 0.5, 0.8, 1.0$). Fig. 10 shows the results. The top row is positional embeddings for each axis; depth and angle, and the bottom row is similarity through positions with dot product of positional embeddings. Each column denotes the α ($\alpha = 0.0, 0.2, 0.5, 0.8, 1.0$). The blue color represents the large value, and red color represents the low value. The top row show that the characteristic elements are concentrated in the first some dimensions from top row. In addition, $\alpha = 0.8$ and $\alpha = 1.0$ are expected to contain more depth features since the spread of depth features is larger than $\alpha = 0.5$ or lower. Furthermore, when we look at the similarity matrix (bottom row), the deeper blue color is concentrated in the center of the matrix at $\alpha = 0.8$ and $\alpha = 1.0$. This indicates that the depths are more closely matched to each other, and that the degree of similarity can be changed by changing the α.

D Multi-View MIMO-FMCW Radar Heatmap Generation

Fig. 11 illustrates the preprocessing flow of the multi-view radar heatmap using data from two MIMO-FMCW radars, which create two orthogonal virtual arrays composed of 86 elements spaced at half-wavelength intervals while transmitting multiple pulses. By sampling the pulses reflected back, a 3D data cube can be formed, which is structured along the horizontal/vertical arrays, ADC samples (intra-pulse or fast-time), and pulse samples (inter-pulse or slow-time). Performing a 3D fast Fourier transform (FFT) on this data cube yields radar spectra across the angle (azimuth for horizontal radar and elevation for vertical radar), range, and Doppler velocity domains. The SNR is further improved by integrating the 3D radar spectra along the Doppler domain, resulting in two radar heatmaps (range-azimuth and range-elevation) in polar radar coordinates. These heatmaps are then projected into the radar Cartesian coordinate system.

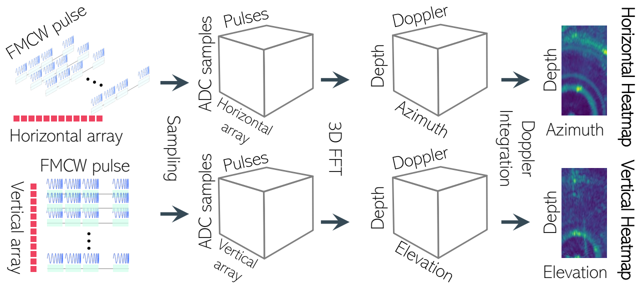

Figure 11: Multi-view heatmap preprocessing.

E Details of Experimental Settings

MMVR Dataset: MMVR [26] has 345K data frames collected from 25 human subjects over 6 different rooms (e.g, open/cluttered offices and meeting rooms) spanning over 9 separate days. MMVR consists of 2 parts: 1) 107.9K data frames of protocol 1 (P1): Open Foreground in a single open-foreground space with a single subject; and 2) 237.9K data frames of protocol 2 (P2): Cluttered space in 5 cluttered rooms with multiple subjects and multiple actions, including sitting postures. Data splits are set as same as S1 in MMVR. "P1" is used to establish the best possible radar perception benchmarks, while "P2" is designed for more challenging scenarios and for cross-environment and cross-subject generalization. The "P2" includes data such as sitting postures; therefore, we select this as the main dataset that we use in our experiments.

HIBER Dataset: HIBER [38] is an open-source multi-view radar dataset including horizontal and vertical radar heatmaps and annotations such as 2D and 3D poses, BBoxes, and segmentation masks. Among its data splits, "WALK" and "MULTI" are currently accessible. The "WALK" split includes 73.5K data frames, each featuring a single person per frame, while "MULTI" consistently includes two individuals per frame. We refined the original BBox labels in the HIBER dataset, addressing their initial overestimation by creating tighter BBoxes; see Fig. 12 for an illustration.

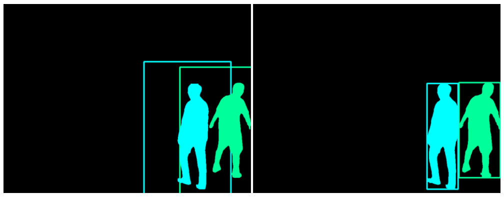

Figure 12: Original (left) versus Refined (right) BBoxes in the HIBER dataset.

Hyper-parameters: The hyper-parameters used in our experiments of Section 5 are shown in Table 5. The table is divided into three parts, Data, Model, and Training, each with parameter names, notations, and values for each dataset.

RFMask with Refined BBoxes: We use RFMask [38] as conventional method for BBox and segmentation tasks. However, RFMask can only predict relaxed BBoxes in the image plane due to its loss calculation being limited to the horizontal plane. Therefore, to train and predict using HIBER dataset (and also MMVR dataset), which consists of refined BBoxes as explained above, an additional module is required to convert the relaxed BBoxes predicted in the image plane into refined BBoxes. As a result, we modify RFMask in a way that the BBox loss is calculated on the image plane and backpropagates to learnable parameters in an end-to-end fashion. Specifically, we add an image BBox regression module alongside a horizontal BBox Regression module, enabling the conversion of BBox offsets to the image plane. By computing loss with respect to these offsets, we can learn refined BBoxes on the image plane. Additionally, the region proposals estimated by the region proposal network (RPN) are transformed into 3D BBoxes based on the fixed-height size, the same as the original RFMask. These BBoxes are then projected onto the image plane and a 3D-to-2D projection.

F Definition of Metrics

Mean Intersection over Union: We adopt average precision on intersection over union (IoU) [6] as an evaluation metric. IoU is the ratio of the overlap to the union of a predicted BBox A and annotated BBox B as:

$$\texttt{IoU}\left(A, B\right) = \frac{|A \bigcap B|}{|A \bigcup B|}. \tag{22}$$

Average Precision: Average Precision (AP) can then be defined as the area under the interpolated precision-recall curve, which can be calculated using the following formula:

$$\texttt{AP} = \sum_{i=1}^{n-1} \left(r_{i+1} - r_i\right) p_{\texttt{interp}}\left(r_{i+1}\right) \tag{23}$$

$$p_{\texttt{interp}}\left(r\right) = \max_{r' \geq r} p\left(r'\right), \tag{24}$$

Table 5: Details of hyper-parameters. Fixed-height size for HIBER dataset is depend on the environment.

	Name	**Notation**	**Value**	
			P2S1 / P2S2	WALK / MULTI
Data	# of training	-	190441 / 118280	58382 / 53690
	# of validation	-	23899 / 33841	3229 / 8260
	# of test	-	23458 / 85677	11931 / 8850
	Input radar heatmap size	$H \times W$	256×128	160×200
	Segmentation mask size	$H \times W$	240×320	624×820
	Resolution of range	cm	11.5	12.2
	Resolution of azimuth	deg.	1.3	1.3
	Resolution of elevation	deg.	1.3	1.3
	Scale	-	log	-
Model	Backbone	-	ResNet18	ResNet18
	Total dimension of positional embedding	-	256	256
	Ratio of depth dimension for TPE	α	0.6	0.6
	# of input frames	-	4	4
	Extracted feature map size	$H/s \times W/s$	64×32	40×56
	Top-K selection	-	magnitude	magnitude
	Top-K	K	256	256
	# of encoder blocks	$L_{\texttt{self}}$	6	6
	# of decoder blocks	$L_{\texttt{cross}}$	6	6
	# of head of multi-head attention	M	4	4
	# of queries	N	10	10
	Threshold for detection and segmentation	-	0.5	0.5
	Fixed-height size (pixel)	H	36	-
	Learning Transformation	-	True	False
Training	Loss weight for `GIoU` on horizontal plane	$\lambda^{\texttt{GIoU}}_{\texttt{hor}}$	0.5	1.0
	Loss weight for `GIoU` on vertical plane	$\lambda^{\texttt{GIoU}}_{\texttt{ver}}$	0.5	1.0
	Loss weight for `GIoU` on image plane	$\lambda^{\texttt{GIoU}}_{\texttt{image}}$	1.0	1.0
	Loss weight for $\texttt{L}_1$ on horizontal plane	$\lambda^{\texttt{L}_1}_{\texttt{hor}}$	0.5	1.0
	Loss weight for $\texttt{L}_1$ on vertical plane	$\lambda^{\texttt{L}_1}_{\texttt{ver}}$	0.5	1.0
	Loss weight for $\texttt{L}_1$ on image plane	$\lambda^{\texttt{L}_1}_{\texttt{image}}$	1.0	1.0
	Batch size	-	32	32
	Epoch for detection	-	100	100
	Epoch for segmentation	-	20	20
	Patience for early stopping	-	5	5
	Check val every N epoch for early stopping	-	2	2
	Optimizer	-	AdamW	AdamW
	Learning rate	-	1e-4	1e-4
	Sheduler	-	Cosine	Cosine
	Maximum number of epochs for sheduler	-	100	100
	Weight decay	-	1e-3	1e-3
	# of workers	-	8	8
	GPU (NVIDIA)	-	A40	A40
	# of GPUs	-	1	1
	Approximate training time	day	2	2

where The interpolated precision $p_{\texttt{interp}}$ at a certain recall level r is defined as the highest precision found for any recall level $r' \geq r$. We present three variants of average precision: $\texttt{AP}_{50}$, $\texttt{AP}_{75}$, and `AP`, where the former two represent the loose and strict constraints of IoU, while `AP` is the averaged score over 10 different IoU thresholds in $[0.5, 0.95]$ with a stepsize of 0.05.

Average Recall: Average recall (AR) [9] between 0.5 and 1 of `IoU overlap threshold` can be computed by averaging over the overlaps of each annotation gt_i with the closest matched proposal, that is integrating over the y : `recall` axis of the plot instead of the x : `IoU overlap threshold` axis. Let o be the `IoU` overlap and $\texttt{recall}\,(o)$ the function. Let $\texttt{IoU}\,(\mathrm{gt}_i)$ denote the `IoU` between

Table 6: Results under “MULTI” on HIBER dataset.

Model	Dim	Input	BBox Loss	AP	AP$_{50}$	AP$_{75}$	AR$_{1}$	AR$_{10}$
DETR	2D	H	H + I	17.00	52.79	6.08	16.21	31.57
DETR (Top-K)	2D	H, V	H + I	22.74	60.64	11.66	18.62	37.62
RETR (TPE@Dec.)	2D	H, V	H + I	23.02	61.77	12.51	19.17	38.00
RETR	2D	H, V	H + I	23.53	63.84	11.90	20.37	38.16
RETR (TPE@Dec.)	3D	H, V	H + V + I	26.36	69.50	14.33	20.76	40.90
RETR	3D	H, V	H + V + I	28.98	74.82	15.64	22.95	41.18

Table 7: Results under “P2S2” on MMVR dataset.

Model	Dim	Input	BBox Loss	AP	AP$_{50}$	AP$_{75}$	AR$_{1}$	AR$_{10}$
RFMask	2D	H, V	H + I	6.03	22.77	0.88	9.25	12.09
DETR (Top-K)	2D	H, V	H + I	7.70	31.16	1.56	16.57	18.26
RETR	2D	H, V	H + I	8.62	24.25	4.40	18.27	21.29
RETR	3D	H, V	H + V + I	9.29	34.69	2.49	20.68	22.82

Table 8: Impact of **Learnable Transformation** (LT) with additional precision and recall metrics.

LT	AP	AP$_{50}$	AP$_{75}$	AR$_{1}$	AR$_{10}$
-	42.97	80.54	41.97	40.20	55.58
✓	46.75	83.80	46.06	42.19	57.39

the annotation gt_i and the closest detection proposal:

$$\texttt{AR} = 2\int_{0.5}^{1} \texttt{recall}(o)\mathrm{d}o = \frac{2}{n}\sum_{i=1}^{n} \max\left(\texttt{IoU}\left(\mathrm{gt}_i\right) - 0.5, 0\right). \tag{25}$$

The followings are some variations of AR:

- AR$_{1}$: AR given 1 detection per data.
- AR$_{10}$: AR given 10 detection per data.
- AR$_{100}$: AR given 100 detection per data.

G Additional Ablation Study

To validate the effectiveness of our RETR, we conducted additional ablation studies. Unless otherwise specified, the hyperparameters follow those listed in the Table 5.

Results under “MULTI” on HIBER dataset: Table 6 shows the evaluation results on the HIBER dataset. From this table, it can be seen that using RETR improves performance across all metrics. Similar to the MMVR results, RETR with tri-plane loss shows enhanced performance. Additionally, the use of RETR with TPE in both the encoder and decoder also contributes to performance improvements. However, compared to the “P2S1” of MMVR, the performance improvement is smaller (the improvement is 15.28 AP from DETR to RETR). This is likely because, unlike “P2S1”, HIBER under “WALK” only involves walking actions, which benefit less from the use of 3D information.

Results under “P2S2” on MMVR dataset: “P2S2” (Cross-Session and Unseen Split) on the MMVR dataset first splits all data segments in d5, d6, d7, and d9 into train, validation, and test sets. Then, it is included all data in d8 in the test set such that one can assess the generalization performance of trained model for an unseen environment (d8). Therefore, “P2S2” is the most challenging scenario in the MMVR. Table 7 shows the evaluation results under “P2S2”. From the results in the table, we confirmed that the prediction performance was improved by using RETR. In particular, RETR outperforms RFMask by a margin of $11.92 + \texttt{AP}_{50}$. However, compared to the results of P2S1, there is a significant decrease in performance, and this is due to the unseen environment.

Table 9: Full table: **Tri-plane loss** can improve the performance.

BBox Loss	AP	AP_{50}	AP_{75}	AR_{1}	AR_{10}	IoU
$\text{H} + \text{I}$	42.18	80.49	39.24	39.39	53.84	75.15
$\text{H} + \text{V} + \text{I}$	46.75	83.80	46.06	42.19	57.39	77.07

Table 10: Impact of the value of K on object detection

K	AP	AP_{50}	AP_{75}	AR_{1}	AR_{10}
64	38.39	80.00	31.85	37.95	50.79
100	38.82	80.06	32.99	38.85	52.28
196	46.97	82.65	45.55	42.93	57.62
256	46.75	83.80	46.06	42.19	57.39

Table 11: Impact of training data size on object detection

# of data	AP	AP_{50}	AP_{75}	AR_{1}	AR_{10}
×0.1	37.10	78.30	31.30	37.50	50.33
×0.5	40.84	79.80	36.82	40.06	53.33
×1.0	46.75	83.80	46.06	42.19	57.39

Table 12: **Inference time** and frame rate (FPS) of RFMask and RETR.

Method	Time[ms]	FPS
RFMask	20.89	47.87
RETR	23.75	42.11

Impact of Tunable Dimension Ratio α in TPE: To investigate the impact of tuning in TPE, we observed the performance differences when varying the ratio of depth and angle dimensions (since the total dimension is $d_{\texttt{pos}} = 256$, e.g., a ratio $\alpha = 0.2$ means depth is rounded to $d_{\texttt{dep}} = \alpha d_{\texttt{pos}} = 102$ dimensions and angle to $d_{\texttt{ang}} = (1 - \alpha) d_{\texttt{pos}} = 154$ dimensions). In Section 5.3, we showed detection results in Table 3b with selected some α and metrics. Here, we show the results with more variants of α and metrics. Fig. 13 shows the result. The horizontal axis denotes the proportion of depth dimensions, and the vertical axis denotes the performance of various metrics. Note that AP_{50} refers to the primary axis, while AP and AP_{75} refer to the secondary axis. The figure shows that the highest performance is achieved when the depth proportion is $\alpha = 0.6$. The performance exhibits a peak at this point, indicating that prioritizing depth improves performance.

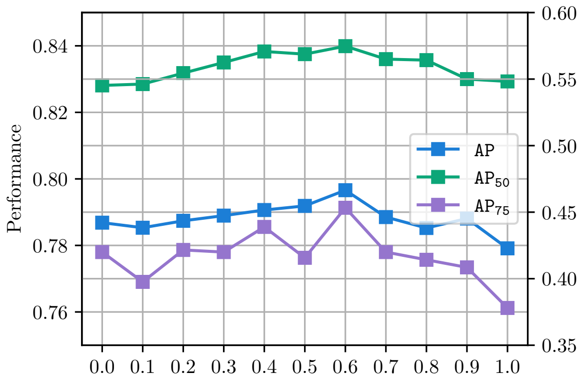

Figure 13: Full figure: Impact of **tunable dimension ratio α in TPE**.

Impact of Learnable Transformation: We expand Table 3b by including additional precision and recall metrics, providing a more comprehensive evaluation of the Learnable Transformation. The complete results are presented in Table 8.

Tri-Plane Loss: In Section 4.5, Table 3c compared RETR with a bi-plane BBox loss (horizontal radar plane and image plane) to that with the tri-plane loss (including the vertical radar plane). We show the more results with complete metrics. The results in Table 9 highlight the necessity of accounting for the vertical BBox loss and the importance of leveraging features from the vertical radar heatmap.

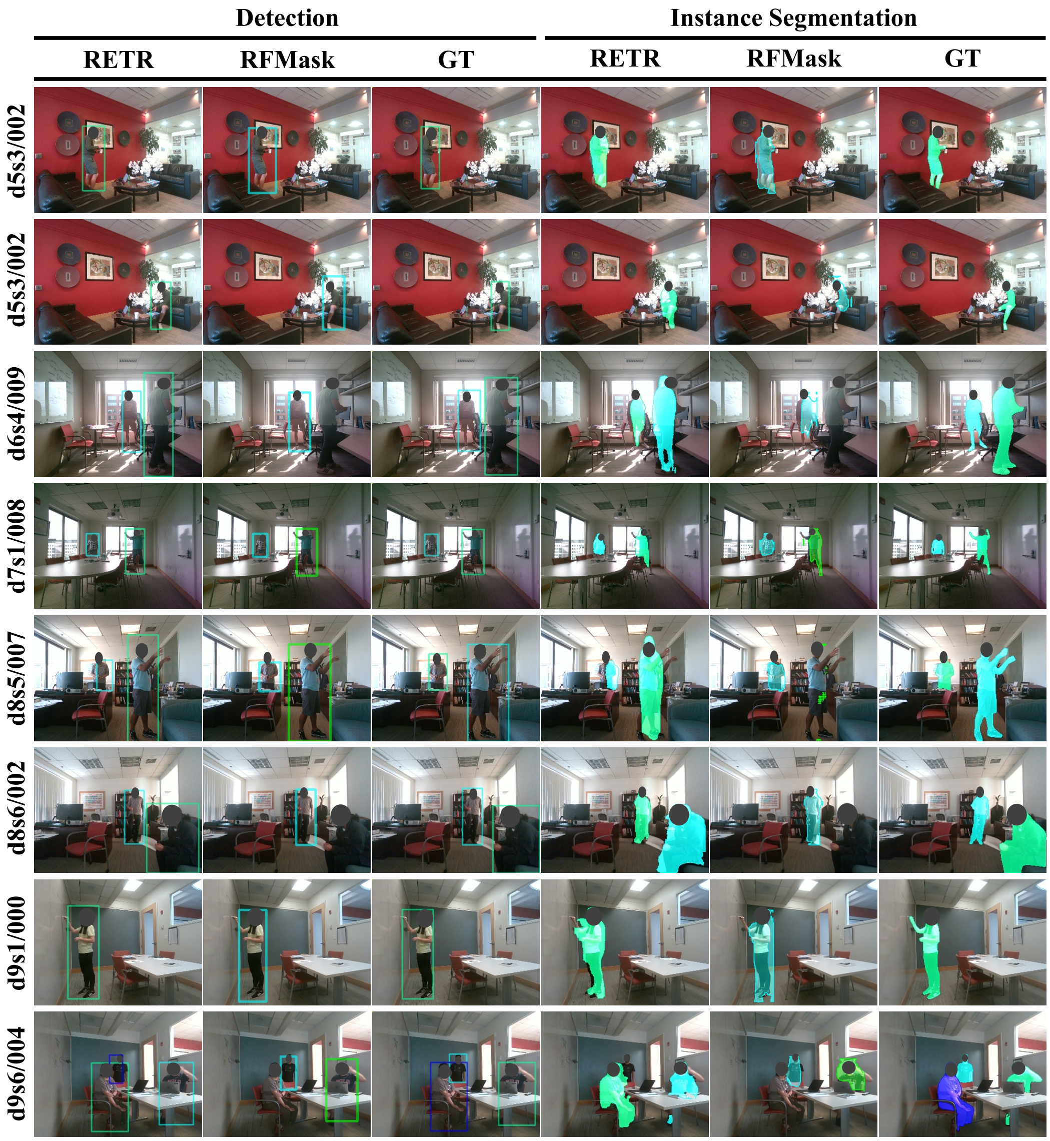

Figure 14: Visualization and comparison between RETR and RFMask. Each row indicates the segment name used from the "P2S1" test dataset.

Impact of the Value of K in Top-K Selection: Table 10 shows that, as the value of K increases (e.g., $K = 196$ and $K = 256$), object detection performance improves across both precision and recall metrics.

Impact of Training Data Size: Table 11 reports the effect of training data size on detection performance using the MMVR dataset. We compare the original data size ($\times 1.0$) with $190,441$ radar frames against reduced data sizes of half ($\times 0.5$) and one-tenth ($\times 0.1$). The results demonstrate a gradual improvement in detection performance as the data size increases.

Inference Time: Table 12 reports the average inference time in milliseconds, evaluated over all frames in the test data using an NVIDIA A40 GPU. RETR achieves an average inference time of 23.75 ms, which is comparable to that of RFMask at 20.89 ms.

H Visualization Result

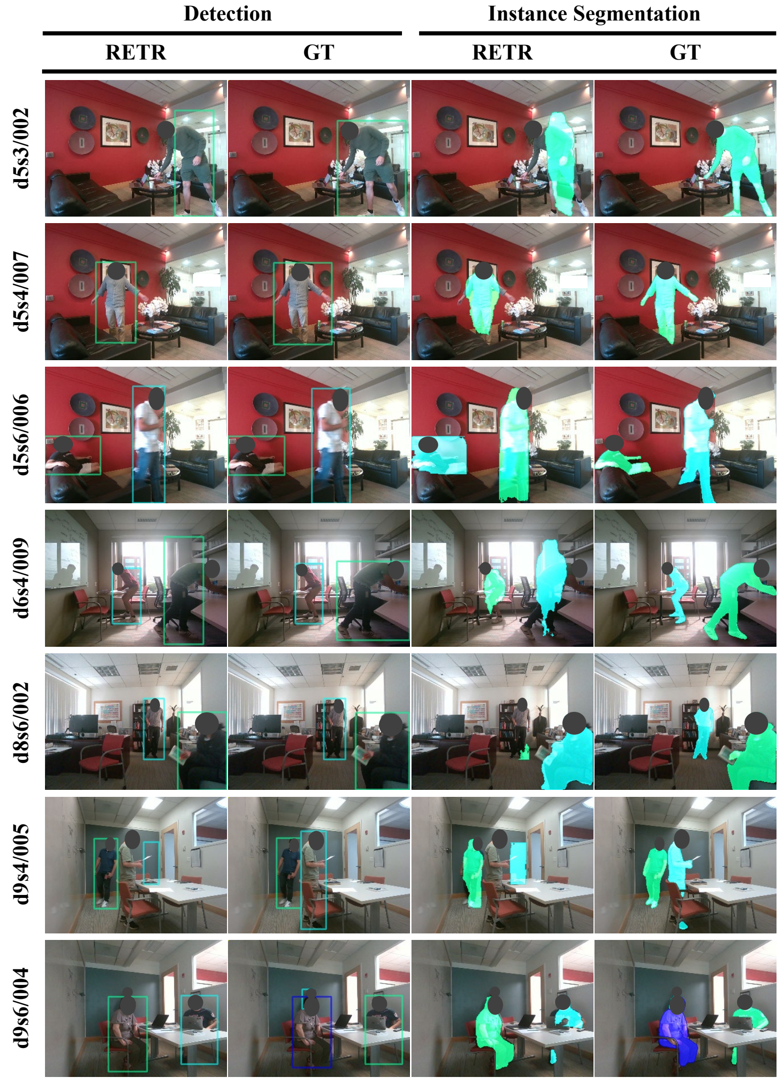

Figure 15: Visualization of failure cases. Each row indicates the segment name used from the "P2S1" test dataset.

Comparison: To compare the conventional RFMask with our RETR, we visualized the prediction results of each. Fig. 14 shows these results. Each row indicates the segment name used from the "P2S1" test dataset. In detection, RFMask has some miss-detections, whereas RETR accurately predicts even when there are multiple subjects. For instance, RFMask tends to fail in detecting people close to the camera, as seen in d8s6/002, but this is improved with RETR. In segmentation, RETR captures human shapes more accurately than RFMask. However, RETR can also fail in mask estimation, as seen in the example of d9s6.

Analysis of Failure Cases: We provide failure cases in Fig. 15. As shown in images such as d5s3/002 and d6s4/009, RETR occasionally mispredicts a bending-over person as standing. Additionally, as shown in d5s4/007, it is often challenging to predict the detailed position of the arms, leading to failures in both detection and segmentation. In some cases, such as d5s6/006 and d8s6/002, the segmentation mask region was excessively large or too narrow. Moreover, in instances such as d9s6/004, while the BBox prediction was successful, the segmentation failed. There was also a case, such as d9s4/005, where the inaccuracy in the BBox prediction led to an incorrect mask position.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the performance improvements in indoor perception achieved by the proposed algorithm using bullet points. Specifically, following each bullet point, Section 4 and Section 5 provide detailed descriptions of the proposed algorithm and its experimental results, demonstrating the effectiveness of the proposed method. Therefore, the overall content is consistent.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included limitations and impacts in Section 5 and Section 6, respectively. We evaluated two datasets; MMVR and HIBER, and the properties of each dataset are described in Section 5 and Appendix E. Moreover, we analyzed the limitations by visualizing the failure cases in the Appendix H.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provided the computational complexity in Appendix A. In the experiments, we provided mainly empirical results. However, by using two different datasets we showed that our proposed method is broadly applicable. In particular, we presented our results using several types of tables and figures.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We included in Appendix E all the important hyperparameters we used in our experiments. We also described the architecture design we built in detail in Appendix A. In addition, we included a split of the dataset in Appendix E as well. This allows us to reproduce the main experimental results in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We plan to release the source code online. And the link to the datasets used in the experiment was provided in Section 5.1.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We included details on their hyperparameters and data. See Table 5 for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the large computational complexity, statistical results are not included. However, we reported more comprehensive experimental results by conducting experiments on multiple datasets and under multiple conditions. See Section 5 and Appendix G for details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Table 5, which specifies computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We carefully confirmed it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We discussed social impacts in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: In our paper, we only evaluated on two datasets that are distributed through regular procedures. We also did not use a pre-training model, so there is no such a risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Each of papers: MMVR and HIBER datasets we used, is properly cited; see that each paper is cited in the dataset explanation in Section 5.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: Our submission includes only the paper. No new datasets or models are released. However, we plan to release the source code and detailed experimental conditions required to reproduce the experiments are included in the Appendix E.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.