RadarMP: Motion Perception for 4D mmWave Radar in Autonomous Driving

Ruiqi Cheng[1], Huijun Di[1*], Jian Li[2, 3*], Feng Liu[4], Wei Liang[1]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2]Key Laboratory of Electronic and Information Technology in Satellite Navigation, Ministry of Education, Beijing, China
[3]Radar Technology Research Institute, School of Information and Electronic, Beijing Institute of Technology, Beijing, China
[4]Beijing Racobit Electronic Information Technology Co., Ltd., Beijing, China
chengrui7@bit.edu.cn, ajon@bit.edu.cn, lijian_551@bit.edu.cn, liufeng@racobit.com, liangwei@bit.edu.cn

Abstract

Accurate 3D scene motion perception significantly enhances the safety and reliability of an autonomous driving system. Benefiting from its all-weather operational capability and unique perceptual properties, 4D mmWave radar has emerged as an essential component in advanced autonomous driving. However, sparse and noisy radar points often lead to imprecise motion perception, leaving autonomous vehicles with limited sensing capabilities when optical sensors degrade under adverse weather conditions. In this paper, we propose RadarMP, a novel method for precise 3D scene motion perception using low-level radar echo signals from two consecutive frames. Unlike existing methods that separate radar target detection and motion estimation, RadarMP jointly models both tasks in a unified architecture, enabling consistent radar point cloud generation and pointwise 3D scene flow prediction. Tailored to radar characteristics, we design specialized self-supervised loss functions guided by Doppler shifts and echo intensity, effectively supervising spatial and motion consistency without explicit annotations. Extensive experiments on the public dataset demonstrate that RadarMP achieves reliable motion perception across diverse weather and illumination conditions, outperforming radar-based decoupled motion perception pipelines and enhancing perception capabilities for full-scenario autonomous driving systems.

Project: — https://github.com/chengrui7/RadarMP

Introduction

Millimeter-wave (mmWave) radar plays a crucial role in autonomous driving perception and navigation (Wang et al. 2022; Wang, Wang, and Liang 2024; Li et al. 2025) systems due to its unique wavelength, which can penetrate weather obstacles. However, conventional CFAR-based radar detection methods (Pace and Taylor 1994; Blake 1988) rely on statistical assumptions and lack the capacity to model complex background clutter or dynamic scenes, resulting in degraded detection performance and producing sparse, noisy radar point clouds. Recent research has proposed deep learning methods (Cheng et al. 2022; Fan et al. 2024; Roldan et al. 2024) that utilize dense point clouds from LiDAR and camera to supervise radar target detection. Due to their different electromagnetic characteristics, using optical sensors to train a radar point enhancement model forces radar to focus on some less prominent reflections in the heatmap and echo signals, thereby hindering the complementarity of multimodal sensing in autonomous systems (Venon et al. 2022).

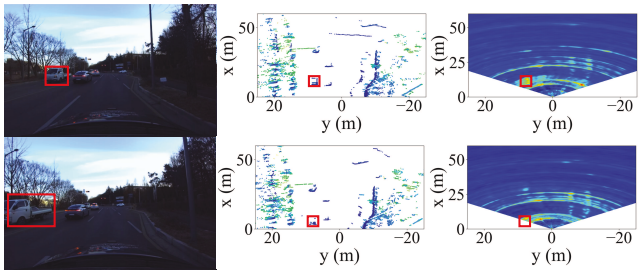

Figure 1: **Motivation schematic.** The red boxes mark a target inter-frame motion in three modalities. The image and LiDAR are shown for visualization purposes only. In the radar heatmap, target motion aligns with the direction of energy propagation, revealing our key motivation.

Precise 3D scene motion perception is essential for scene understanding in autonomous driving. Scene flow, which estimates the pointwise motion in the 3D world by leveraging two consecutive frames from cameras (Yin and Shi 2018; Bayramli, Hur, and Lu 2023) or LiDARs (Shen et al. 2023; Cheng and Ko 2023; Qingwen et al. 2024), has been studied for years. With the emergence of 4D mmWave radar, the improved spatial resolution has made it feasible to perform the scene flow estimation task. But radar point clouds produced by low signal-to-noise ratio (SNR) target detectors exhibit significant noise and temporal inconsistency, leading to relatively unreliable and inaccurate scene flow estimation. To the best of our knowledge, Ding et al. (Ding et al. 2022, 2023) employ self-supervision and cross-modal supervision to estimate scene flow from two frames of radar point clouds on the VoD dataset (Palffy et al. 2022), making them the only existing works in this context, and their performance remains substantially inferior to LiDAR-based methods.

Our motivation stems from the assumption that the energy flow of target points across adjacent frames of radar echo signals should be consistent with the motion field, as shown in Figure 1. In contrast, the energy flow of noise points tends

*indicates corresponding authors.

to be disordered and irregular. To enhance the scene motion perception capability of mmWave radar and remove false targets from radar echoes, we design a radar target detector that is consistent with the motion field of the adjacent radar frame, and simultaneously outputs scene flow estimation while detecting targets. Our method begins with the two successive 4D radar tensors, derived via multi-dimensional FFT, to perform consistent radar target detection and scene flow estimation. The radar tensor is a low-level representation of radar signals, storing echo intensity in a 4D cube (i.e., a tesseract). Several works (Ding et al. 2024; Chae, Kim, and Yoon 2024) have achieved high-accuracy object detection and occupancy prediction tasks using radar tesseract, which demonstrates the excellent performance of this signal format in mmWave radar applications.

Our contributions can be summarized as follows:

1. We propose RadarMP, the first architecture that jointly addresses mmWave radar target detection and scene flow estimation tasks, using adjacent frame radar tesseract signal inputs to generate consistent radar point clouds and scene flow outputs.
2. We introduce multiple specialized self-supervised loss functions based on the Doppler characteristics and echo intensity of radar signals to supervise both point cloud generation and scene flow estimation.
3. We conduct extensive experiments on the public dataset to validate the performance and effectiveness of the proposed method, which significantly enhances the motion perception capability of mmWave radar in full-scenario autonomous driving systems.

Preliminary

Radar Tesseract Generation Workflow

Radar transmits electromagnetic beams via its transmit (TX) antennas, which are reflected by targets and received by the receive (RX) antennas as echo signals. Most 4D mmWave radars use Frequency-Modulated Continuous Wave (FMCW) signals for transmission. A single frame of FMCW radar typically consists of multiple transmission cycles (i.e., chirp) where the signal frequency increases linearly over a short time within a chirp. Each TX-RX antenna pair processes the echo signals in a radar frame through a mixer and an Analog-to-Digital converter (ADC), resulting in digital signals referred to as raw ADC data. All raw ADC data are organized along the signal duration, chirp index, and antenna pair dimensions to form a 3D complex data cube, where the three axes correspond to fast time, slow time, and channel, respectively.

Fast Fourier Transforms (FFTs) are applied along the corresponding dimensions of the ADC data to extract detailed physical-domain information to construct a 4D radar tensor: range FFT recovers the propagation delay as range bins r, Doppler FFT estimates relative radial velocity d, and two spatial FFTs across the antenna array yield azimuth a and elevation e angles of arrival (AoA). The detailed workflow is depicted in Figure 2. In this paper, we refer to the 4D tensor as the tesseract, where each cell corresponds to the echo intensity at a location (d, r, a, e) in the Doppler–range–azimuth–elevation space for a radar frame.

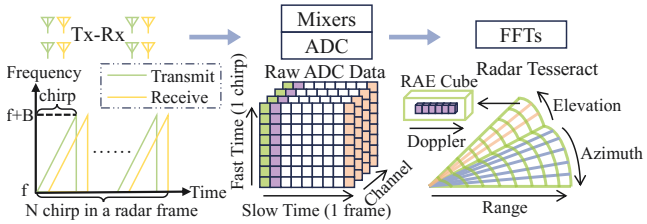

Figure 2: **Tesseract generation pipeline**. Radar antenna array transmits multiple chirp signals per cycle. After receiving the echoes, the signals are mixed and sampled by ADCs to obtain the raw radar data, which is transformed into a radar tesseract via multi-dimensional FFT.

Radar Tesseract for Motion Perception

The radar tesseract exhibits a dense structure capable of capturing motion in complex 3D environments, combining the depth dimension of LiDAR with the dense coverage characteristic of camera images. It preserves the comprehensive measurement of raw radar signals, avoiding the sparsity, noise, and clutter commonly introduced during traditional radar point cloud preprocessing. For specific targets that are often overlooked during radar signal preprocessing but are critical in autonomous driving scenarios (such as pedestrians wearing low-reflectivity clothing, pets with fur, or asphalt road surfaces), the tesseract could significantly enhance motion perception and contribute to the safety and reliability of autonomous driving systems.

Despite its advantages, using the radar tesseract for motion perception remains a challenging task. The dense structure of the radar tesseract results in substantial memory consumption (each frame in the Kradar dataset occupies nearly 300 MB), necessitating carefully designed models and processing strategies to mitigate GPU memory pressure and enhance computational efficiency. Moreover, the noise within the tesseract is further amplified by the multipath effects inherent to mmWave signals, needing effective filtering mechanisms to suppress the adverse impact of such noise on motion perception performance.

Methodology

Task Defination

In this work, we address the motion perception task using two consecutive radar tesseracts output from a 4D mmWave radar. The input consists of a source frame $\mathbf{S} \in \mathbb{R}^{D\times R\times A\times E}$ and a target frame $\mathbf{T} \in \mathbb{R}^{D\times R\times A\times E}$, where D, R, A, and E denote the Doppler, range, azimuth, and elevation dimensions respectively.

Our framework simultaneously solves two complementary subtasks:

1. **Segmentation Prediction**: We generate a binary segmentation mask $\mathbf{M} \in \{0, 1\}^{R\times A\times E}$ for the source frame

$\mathbf{S}$, where $\mathbf{M}(r, a, e) = 1$ identifies valid targets at spatial position (r, a, e), while $\mathbf{M}(r, a, e) = 0$ denotes noise points.

2. **Scene Flow Estimation**: For each detected target point in the source frame $\mathbf{S}$ (where $\mathbf{M}(r, a, e) = 1$), we estimate a 3D scene flow field $\mathbf{F} = \{\mathbf{f}_i\}$. Each flow vector $\mathbf{f}_i = (\Delta r_i, \Delta a_i, \Delta e_i)$ represents the displacement of the target along the Range, Azimuth, and Elevation axes.

Overview

The overall architecture is illustrated in Figure 3. The two consecutive radar tesseracts are first unfolded along the three spatial planes, and a flow estimation network processes each plane to obtain coarse initial motion estimates and generate 3D reference points. After encoding along the Doppler channel dimension, the tesseracts are passed through dense 3D convolutions to extract multi-scale radar features. Combining the multi-scale features with the 3D reference points, RadarMP employs a multi-scale deformable cross-attention module to extract inter-frame correlation features. We enhance the correlation features with global context across different dimensions to distinguish motion cues, and finally decode to produce motion perception outputs trained in a self-supervised manner. The following section elaborates on the RadarMP framework and our tailored self-supervised loss functions.

Doppler Channel Encoding

Unlike the other three spatial dimensions, prior studies (Paek, Kong, and Wijaya 2023; Kong, Paek, and Lee 2024; Chae, Kim, and Yoon 2024) have considered the Doppler dimension D at each (r, a, e) coordinate to be redundant and often reduced to a minimal-dimensional representation by applying average or max pooling. However, the Doppler axis at each spatial location encodes critical motion-related attributes, providing both semantic and physical cues for segmentation and scene flow estimation, relatively.

To this end, it is necessary to encode the energy values along the Doppler dimension. We treat the Doppler axis in each voxel as feature channels and apply a multi-layer perceptron (MLP) to transform them into a compact representation. The encoded output is a Doppler-aware feature vector $F_d \in \mathbb{R}^{C_d \times R \times A \times E}$. In processing the Doppler axis, we not only consider the raw power, but also incorporate the corresponding Doppler velocity associated with each index. The energy distribution along the Doppler axis reflects the confidence of each spatial location with respect to different Doppler velocities. To capture this, we apply both Softmax and Gumbel-Softmax (for one-hot) (Jang, Gu, and Poole 2017) functions to encode the Doppler velocity. For each spatial location (r, a, e), we denote the raw Doppler bins as $P_d \in \mathbb{R}^D$. The Doppler velocity feature F_v is computed as follows:

$$\begin{aligned} F_{v1} &= \mathrm{sum}(\mathrm{matmul}(Ax_d, \mathrm{Softmax}(P_d))), \\ F_{v2} &= \mathrm{sum}(\mathrm{matmul}(Ax_d, \mathrm{GumbelSoftmax}(P_d))), \end{aligned} \tag{1}$$

where $Ax_d \in \mathbb{R}^D$ denotes the Doppler axis. Finally, we concatenate F_v with F_d forming the final Doppler-aware representation $F_{dv} \in \mathbb{R}^{(C_d+2) \times R \times A \times E}$.

Note that the Doppler encoding process is applied identically to both the source frame $\mathbf{S}$ and target frame $\mathbf{T}$. Consequently, this approach enables us to reduce the first dimension of tesseract to $D/8$ while retaining the essential characteristics of the Doppler bins.

Correlation Feature Extraction

Directly applying dense correlation over the entire 3D radar spherical space, as in image-based cost volumes, would lead to severe memory overhead. Deformable attention (Zhu et al. 2021) has demonstrated strong performance in 3D object detection and occupancy prediction tasks, while significantly reducing the computational complexity of attention modules. Inspired by this, we treat the source frame $\mathbf{S}$ as the query and the target frame $\mathbf{T}$ as the value, and compute a correlation field using cross-deformable attention between the two radar tesseracts.

As a definition, given a query feature $\mathbf{q}$ and its corresponding reference point $\mathbf{p}$, deformable attention updates the query by aggregating features from the value feature $\mathbf{V}$ according to the following equation:

$$\begin{aligned} &\mathrm{DeformAttn}(\mathbf{q}, \mathbf{p}, \mathbf{V}) = \\ &\sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mk} \cdot W'_m \mathbf{V}(\mathbf{p} + \Delta \mathbf{p}_{mk}) \right], \end{aligned} \tag{2}$$

where $\Delta \mathbf{p}_{mk}$ and A_{mk} are learnable sampling offsets and learnable attention weight predicted from the query $\mathbf{q}$ for its m_{th} head and k_{th} sampling point, $\mathbf{V}(\mathbf{p} + \Delta \mathbf{p}_{mk})$ is the value features at the sample location $(\mathbf{p} + \Delta \mathbf{p}_{mk})$, and W_m and W'_m are the learnable transformation matrix for the m_{th} attention head.

Correlation Reference Point For optimal correlation feature capture through deformable attention, we should align the reference points as closely as possible to their actual warped locations in the target frame. Inspired by the projection of 3D cubes onto 2D planes in DPFT (Fent, Palffy, and Caesar 2024), we project both $\mathbf{S}$ and $\mathbf{T}$ onto range-azimuth (RA), range-elevation (RE), and elevation-azimuth (AE) planes. Subsequently, we employ a pretrained PWC-Net (Sun et al. 2018) architecture to predict the energy flow directions on these 2D projections. We slightly modify the original PWC-Net to regress 2D motion fields at three different scales (i.e., 1, 1/2, and 1/4) for providing multi-scale reference point. The output motion fields consist of three 2D flow components: $\mathbf{F}_{ra}$, $\mathbf{F}_{re}$, and $\mathbf{F}_{ae}$. Taking the RA-plane flow as an example, it is defined as $\mathbf{F}_{ra} = \{f^l_{ra}\}$, where each $f^l_{ra} \in \mathbb{R}^{2 \times \frac{R}{2^l} \times \frac{A}{2^l}}$, with $l = 0, 1, 2$.

The reference point coordinates $\mathbf{P} = \{\mathbf{p}_l\}$ at each feature level l are computed by averaging the 2D flow predictions across three spatial planes (RA, RE, and AE). Specifically, the estimated motion in each plane is extended along the missing spatial dimension to construct full 3D flow volumes. The reference coordinates $\mathbf{p}_l \in \mathbb{R}^{3 \times \frac{R}{2^l} \times \frac{A}{2^l} \times \frac{E}{2^l}}$ are obtained by applying the averaged flow displacements to the original query grid positions along the range, azimuth, and elevation axes.

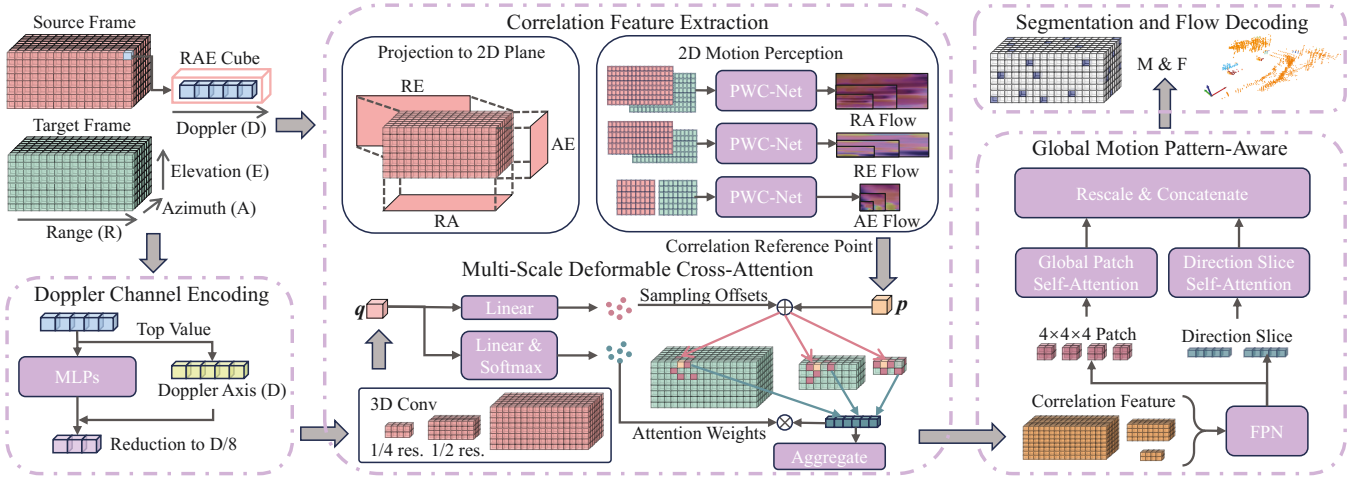

Figure 3: **Pipeline Overview.** RadarMP processes two consecutive radar tesseracts through Doppler encoding and correlation feature extraction, followed by global motion pattern perception to derive motion cues, and finally decodes them into segmentation masks and flow predictions.

Multi-Scale Deformable Cross Attention To capture motion information at multiple scales, we employ multi-scale deformable attention (Zhu et al. 2021; Li et al. 2023) to update the correlation features between $\mathbf{S}$ and $\mathbf{T}$. First, we extract two three-level feature pyramids, $F_L^{\mathbf{S}}$ and $F_L^{\mathbf{T}}$, from the Doppler-encoded features $F_{dv}^{\mathbf{S}}$ and $F_{dv}^{\mathbf{T}}$ using a ResNet3D backbone (Hara, Kataoka, and Satoh 2018). Taking the source frame as an example, the pyramid is defined as $F_L^{\mathbf{S}} = \{F_l^{\mathbf{S}}\}$, where each $F_l^{\mathbf{S}} \in \mathbb{R}^{C_l \times \frac{R}{2^l} \times \frac{A}{2^l} \times \frac{E}{2^l}}$, with $l = 0, 1, 2$. We apply the MLP to unify the channel dimensions across multiple scales and flatten them, producing the query $\mathbf{Q_S}$ and the value $\mathbf{V_T}$. In our method, the correlation feature F_c between the two radar tesseracts is computed via the following multi-scale deformable attention equation:

$$\text{MSDeformAttn}(\mathbf{q}, \mathbf{p}, \{\mathbf{v}_\mathbf{T}^l\}) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlk} \cdot W'_m \mathbf{v}_\mathbf{T}^l(\mathbf{p} + \Delta\mathbf{p}_{mlk}) \right], \quad (3)$$

where $\mathbf{V_T} = \{\mathbf{v}_\mathbf{T}^l\}$ is the multi-scale feature maps of the target frame, $\Delta\mathbf{p}_{mlk}$ and A_{mlk} are the learnable sampling offsets and learnable attention weight predicted from the query $\mathbf{q}$ for its k_{th} sampling point at the l_{th} feature level and the m_{th} head.

Both the key and query elements are from the flatten multi-scale feature maps. The reference point $\mathbf{p} \in \mathbf{P} = \{p_l\}$ for each query feature $\mathbf{q} \in \mathbf{Q_S} = \{\mathbf{q}_\mathbf{S}^l\}$ is computed at the corresponding scale via the pseudo warping operation. By applying Eq. (3) to perform multi-scale feature interaction on $F_{dv}^{\mathbf{S}}$ and $F_{dv}^{\mathbf{T}}$, the resulting multi-scale correlation feature map $F_L^{\mathbf{C}} = \{F_l^{\mathbf{C}}\}$ is derived as follows:

$$F_l^{\mathbf{C}} = \text{MSDeformAttn}\big(\mathbf{q}, \mathbf{p}, \{\mathbf{v}_\mathbf{T}^l\}\big), \quad \mathbf{q} \in \mathbf{q}_\mathbf{S}^l. \quad (4)$$

By applying a Feature Pyramid Network (FPN) (Lin et al. 2017) to aggregate the multi-scale correlation features, we obtain the correlation representation $F_c \in \mathbb{R}^{C_c \times R \times A \times E}$, which captures the correspondence between the two radar frames across multiple resolutions.

Global Motion Pattern-Aware Module

Decoding target segmentation requires awareness of global motion patterns to distinguish the spatial distribution of tesseract motion. The motion characteristics of noise, static targets, and dynamic targets are disordered, globally correlated, and locally correlated, respectively. To provide sufficient cues for accurate segmentation, we introduce two self-attention modules to capture global context and enhance effective target detection.

Global Patch Self-Attention F_c is derived from the summation of multi-scale correlation feature map $F_L^{\mathbf{C}}$ and the query feature $\mathbf{Q_S}$ during output. Consequently, F_c contains both context feature and correlation feature. To minimize memory consumption, we employ the token reduction strategy in conjunction with ViT (Dosovitskiy et al. 2021). Specifically, F_c is partitioned into $4 \times 4 \times 4$ patches, with each patch treated as one token in the Transformer encoder. Since the pointwise motion vectors are correlated with the spatial coordinates of the corresponding targets, we similarly convert the polar coordinates of all spatial elements in the radar tesseract into $4 \times 4 \times 4$ patches as the positional encoding for the corresponding feature tokens.

Direction Slice Self-Attention The final target flow field is also correlated with the directional vector of each point target, which remains constant across all range bins within the same (a, e) slice. To preserve fine-grained segmentation features that may be degraded by volumetric patching, we propose a slicing strategy along the AE plane. We treat all range bins at the (a, e) of F_c as a token to the Transformer encoder and input the directional vector as the positional encoding for the corresponding (a, e) token, thereby enhancing the fine-grained representation of global motion pattern-aware.

Segmentation and Flow Decoding

The features with segmentation cues obtained from the global patch self-attention and direction slice self-attention do not match the spatial dimensions of the original tesseract, and thus need to be rescaled to the original resolution. Obtained from global patch self-attention, $F_p \in \mathbb{R}^{C_p \times \frac{R}{4} \times \frac{A}{4} \times \frac{E}{4}}$ is restored by rearranging the patches along the channel dimension, resulting in $F'_p \in \mathbb{R}^{\frac{C_p}{64} \times R \times A \times E}$. $F_s \in \mathbb{R}^{R \times A \times E}$ is the output of the direction slice self-attention, whose spatial dimensions are compressed along the range axis, leaving only azimuth and elevation. We set its channel dimension equal to the number of range bins, so that each channel corresponds to the feature of one range bin. We expand the channel dimension of F_s to 1 and concatenate it with F'_p to obtain the complete global motion pattern feature $F_g \in \mathbb{R}^{C_g \times R \times A \times E}$.

The global motion pattern feature F_g and the correlation feature F_c, which contains the source frame's contextual information, are respectively passed through MLPs and then concatenated to form a unified feature F_u. This unified feature is then fed into the segmentation head and the flow head to produce the final outputs: the binary segmentation confidence $\mathbf{M}_s \in \mathbb{R}^{1 \times R \times A \times E}$ and the scene flow prediction $\mathbf{F}_s \in \mathbb{R}^{3 \times R \times A \times E}$, where each value in $\mathbf{M}_s \in [0, 1]$.

Loss Function

To supervise motion perception, we introduce three self-supervised loss terms: segmentation energy loss $\mathcal{L}_{se}$, energy flow loss $\mathcal{L}_{ef}$, and radial flow segmentation loss $\mathcal{L}_{rfs}$. The overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{se} + \mathcal{L}_{ef} + \mathcal{L}_{rfs}. \tag{5}$$

This loss function jointly optimizes the network from three aspects: energy distribution, energy flow direction, and the interaction between energy flow and energy distribution on the Doppler channel, respectively.

Segmentation Energy Loss We apply the Gumbel-Softmax and mean operations across the Doppler dimension of the tesseract to obtain the maximum and summation energy features, respectively. These are concatenated to form E_f, which represents the maximum energy characteristics at each (r, a, e) coordinate. Then, the noise energy level τ_f is estimated from the local and channel energy statistics. The greater the difference between E_f and τ_f, the higher the relative energy of the point, indicating a higher likelihood of being a target candidate. Moreover, segmentation masks should exhibit consistency between the source and target frames. The segmentation energy loss is as follows:

$$\begin{aligned} \mathcal{L}_{se} &= \mathbf{M}_s - \mathrm{sigmoid}(E_f^{\mathbf{S}} - \tau_f^{\mathbf{S}}) \\ &+ \mathbf{M}_s \times (\mathrm{warp}(\mathbf{M}_s, \mathbf{F}_s) - \mathrm{sigmoid}(E_f^{\mathbf{T}} - \tau_f^{\mathbf{T}})). \end{aligned} \tag{6}$$

Energy Flow loss The flow field of a target point must align with its energy flow direction. Based on this assumption, we introduce the Energy Flow Loss. Unlike optical flow in images, the energy flow field also includes disordered noise flows. To mitigate the impact of such noise on the loss function, we use energy intensity as the weighting factor, encouraging the model to focus more on the flow directions of target points. The energy flow loss is as follows:

$$\mathcal{L}_{ef} = E_f^{\mathbf{S}} \times (E_f^{\mathbf{S}} - \mathrm{warp}(E_f^{\mathbf{T}}, \mathbf{F}_s)). \tag{7}$$

Radial Flow Segmentation Loss The Doppler value (i.e., radial relative velocity) multiplied by the inter-frame time Δt should approximate the radial projection of the truth flow at the target point, which is the core insight behind the self-supervision of the radial flow segmentation loss. We define the Doppler candidate values F_v (obtained from Doppler channel encoding, as shown in Eq. (1)). For convenience, we convert the polar coordinates volume (obtained from the global patch self-attention) into Cartesian coordinates C. The directional vector volume O (obtained from the direction slice self-attention) represents the direction vectors of grid centers relative to the radar. The radial flow segmentation loss is as follows:

$$\begin{aligned} \delta_v &= F_v - \frac{\mathrm{warp}(C, \mathbf{F}_s) - C}{\Delta t} \odot O, \\ \mathcal{L}_{rfs} &= \mathbf{M}_s - \mathrm{sigmoid}(\alpha(\beta - \delta_v^2)), \end{aligned} \tag{8}$$

where α and β represent the tolerance for δ_v.

Experiment

Experimental Setup

Dataset We conduct experiments on the K-Radar dataset (Paek, Kong, and Wijaya 2022), which is currently the only autonomous driving dataset that provides radar signals in the tesseract format. Moreover, the K-Radar dataset also includes time-synchronized multi-view images, LiDAR point clouds, odometry information, and annotations for 3D object detection and tracking. The front-view images and LiDAR point clouds enable comparison of the performance of different modalities in motion perception. We utilize the odometry data, along with the 3D detection and tracking annotations, to generate scene flow labels.

Implementation In our experiments, we train the model using the Adam optimizer (Kingma and Ba 2015). The learning rate is initially set to 0.001 and decays exponentially by a factor of 0.9 every 2 epochs. The multi-scale deformable cross-attention module in our method is configured with three attention heads and 50 sampling points. For the self-attention modules, we adopt the native PyTorch implementation with two heads and two layers. We train RadarMP using three NVIDIA RTX 3090 GPUs. And RadarMP achieves 7.6 fps inference on a 3090 using 7.5 GB GPU memory.

Segmentation Evaluation

Metrics To evaluate the segmentation performance between target and noise in the radar tesseract, we need to compare the occupancy consistency of the output against the LiDAR point cloud. When the output target point has at least 3 LiDAR points within its local neighborhood, we consider it a true positive. Based on this criterion, we compute probability of detection (P_d) and probability of False

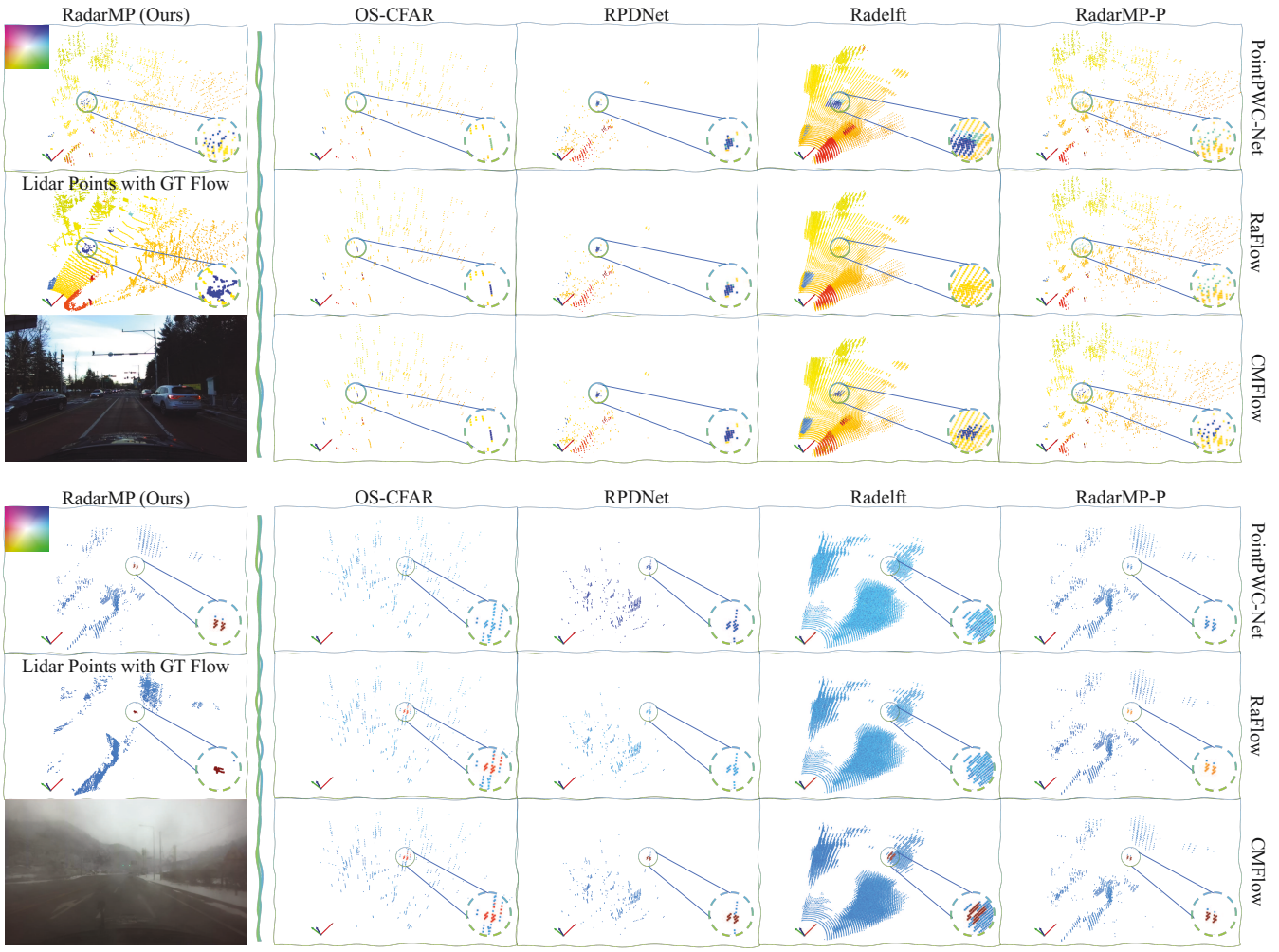

Figure 4: **Qualitative results.** The **left** side shows the motion perception output of RadarMP alongside LiDAR point clouds filtered by RoI with ground-truth scene flow. Columns 1–4 on the **right** side display radar target detection results from three segmentation baseline methods and our RadarMP (RadarMP-P), while rows 1–3 correspond to flow prediction results from different scene flow baselines. A dynamic object in the scene is zoomed in at the bottom right to highlight the accuracy of non-rigid motion estimation. Colors indicate motion vectors in the XY plane only and RGB image is used for visualization only.

Alarm (P_{fa}) as segmentation metrics. We also calculate the Chamfer Distance (CD) between the predicted points and the voxel-filtered LiDAR point set to evaluate the spatial distribution of the target points. Moreover, we calculate the average target energy and average noise energy based on the segmentation results to compute the signal-to-noise ratio (SNR) metric.

Baselines For a comprehensive comparison, we select the traditional OS-CFAR method and two learning-based methods (Cheng et al. 2022; Roldan et al. 2024) that enhance 3D radar point clouds using LiDAR supervision as baselines. In the OS-CFAR method, the number of background cells is set to 4, the guard cells to 1, and the false alarm rate to 1e-6. Moreover, we adopt the default hyperparameters for the learning-based methods to ensure fairness.

Results We quantitatively compare with baseline methods on the test sets, as presented in Table 1. Compared to relying on energy thresholds or LiDAR supervision for segmentation, our method integrates energy constraints and scene motion consistency, resulting in a significantly mean higher detection probability of target points. Benefiting from the combination of multiple loss functions, RadarMP avoids introducing excessive pseudo targets, maintaining a low false alarm rate while achieving a clear perception of the surrounding scene, as illustrated in Figure 4.

Flow Evaluation

Metrics We adopt four commonly used scene flow metrics (Gu et al. 2019) to evaluate the performance of flow field estimation: 1) EPE3D (m): the average end-point-error between the predicted and ground-truth scene flow of target

Method	P_d(%)↑	P_{fa}(%)↓	CD(m)↓	SNR(dB)↑
OS-CFAR	1.643	**0.311**	10.030	**5.477**
RPDNet	9.311	1.821	7.590	5.175
Radelft	<u>44.121</u>	6.200	<u>6.553</u>	4.329
RadarMP	**69.458**	<u>1.335</u>	**3.378**	<u>5.232</u>

Table 1: Radar target detection results. The bold number indicates the best result, and the underlined number represents the second-best result. ↑ means bigger values are better, and vice versa.

points, 2) AccS3D (%): the percentage of target points with endpoint error satisfying the strict condition (EPE3D < 0.05 m or relative error $< 5\%$), 3) AccR3D (%): the percentage of target points meeting the relaxed condition (EPE3D < 0.1 m or relative error $< 10\%$), 4) Outlier3D (Out3D) (%): the percentage of target points whose endpoint error exceeds the threshold (EPE3D > 0.3 m or relative error $> 10\%$).

Method/Metric		EPE3D↓	AccS3D↑	AccR3D↑	Out3D↓
PPN	**OS-CFAR**	0.49	5.55	9.39	93.34
	RPDNet	0.35	11.38	16.73	79.72
	Radelft	0.50	6.34	18.67	89.06
	RadarMP-P	0.22	20.01	39.74	59.62
RaFlow	**OS-CFAR**	0.33	11.64	20.89	82.40
	RPDNet	0.31	12.90	25.65	78.12
	Radelft	0.46	10.28	25.33	81.97
	RadarMP-P	0.18	18.88	40.64	53.40
CMFlow	**OS-CFAR**	0.28	15.73	32.81	72.96
	RPDNet	0.25	17.15	36.16	68.99
	Radelft	0.19	20.15	46.58	65.26
	RadarMP-P	<u>0.17</u>	<u>20.40</u>	**47.99**	<u>50.84</u>
RadarMP (Ours)		**0.15**	**21.37**	<u>46.87</u>	**44.73**

Table 2: Scene flow evaluation results. Baselines include combinations of a flow prediction model with different segmentation methods.

Baselines We selected two self-supervised scene flow estimation methods as flow evaluation baselines: PointPWC-Net (PPN), the first self-supervised method for point cloud scene flow estimation, and RaFlow(Ding et al. 2022) and CMFlow(Ding et al. 2023), the only two scene flow prediction models designed for mmWave radar point clouds supervised by self and cross-modal, respectively. For a fair comparison, we apply these three methods to the radar points generated from the segmentation baseline, as well as the target point obtained from RadarMP (RadarMP-P), to predict scene flow. The scene flow estimated above is subsequently evaluated against the flow predictions generated by RadarMP.

Results We show the evaluation metric results for flow prediction in Table 2. Taking the complementary self-supervised losses from segmentation and flow estimation, RadarMP predicts scene motion more accurately than baseline methods using only two consecutive radar tensors. Figure 4 presents qualitative comparisons with all baselines across two sample sequences. Compared to low-resolution and noisy point clouds, the motion of strong reflectors, such as vehicles, is more precisely captured within the tesseract. At the same time, global energy flow cues could infer rigid body motion. It can be observed that our predicted flow fields closely match the ground-truth motion.

Ablation Study

To validate the effectiveness of the three self-supervised loss functions we designed, we conduct ablation studies on different combinations of these losses. The results are shown in Table 3, where the bottom row reports the performance of RadarMP with the complete loss configuration. Each loss term contributes to improving the overall performance in both object detection and motion perception. To examine the dependence on PWCNet, which provides reference points for 2D flow predictions across three planes, the ablation study shows that turning off this module and using each cube's own location as its reference point worsens the EPE3D metric by 0.31. When replacing PWCNet with a weak version, the EPE3D metric worsens by 0.07.

$\mathcal{L}_{se}$	$\mathcal{L}_{ef}$	$\mathcal{L}_{rfs}$	P_d (%)↑	P_{fa}(%)↓	EPE3D (m)↓
√	√		62.033	2.258	0.209
√		√	56.224	3.847	0.788
	√	√	19.846	17.136	0.621
√	√	√	**69.458**	**1.335**	**0.157**

Table 3: Ablation experiments on loss terms. The √ indicates that the loss function is enabled in the model.

Conclusion

In summary, we propose RadarMP, a novel framework that leverages inter-frame energy propagation to enable motion perception for millimeter-wave radar in autonomous driving systems. RadarMP addresses two key tasks in an end-to-end manner: 1) target detection from low-level radar tesseract signals by performing motion-consistent segmentation of targets and noise in dense radar tensors, and 2) scene flow prediction for each segmented target point by extracting consecutive frame correlation features. To retain radar sensing independence and complementarity, we design self-supervised loss functions tailored to radar characteristics. Extensive experiments on the K-Radar dataset demonstrate both qualitative and quantitative superiority of RadarMP for radar-based motion perception. We believe our work could inspire and advance future research on 4D imaging radar in motion perception for autonomous driving.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (NSFC) under Grant No.62172043.

References

Bayramli, B.; Hur, J.; and Lu, H. 2023. RAFT-MSF: Self-supervised monocular scene flow using recurrent optimizer. *International Journal of Computer Vision*, 131(11): 2757–2769.

Blake, S. 1988. OS-CFAR theory for multiple targets and nonuniform clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 24(6): 785–790.

Chae, Y.; Kim, H.; and Yoon, K.-J. 2024. Towards Robust 3D Object Detection with LiDAR and 4D Radar Fusion in Various Weather Conditions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15162–15172.

Cheng, W.; and Ko, J. H. 2023. Multi-Scale Bidirectional Recurrent Network with Hybrid Correlation for Point Cloud Based Scene Flow Estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10007–10016.

Cheng, Y.; Su, J.; Jiang, M.; and Liu, Y. 2022. A Novel Radar Point Cloud Generation Method for Robot Environment Perception. *IEEE Transactions on Robotics*, 38(6): 3754–3773.

Ding, F.; Palffy, A.; Gavrila, D. M.; and Lu, C. X. 2023. Hidden Gems: 4D Radar Scene Flow Learning Using Cross-Modal Supervision. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9340–9349.

Ding, F.; Pan, Z.; Deng, Y.; Deng, J.; and Lu, C. X. 2022. Self-Supervised Scene Flow Estimation With 4-D Automotive Radar. *IEEE Robotics and Automation Letters*, 7(3): 8233–8240.

Ding, F.; Wen, X.; Zhu, Y.; Li, Y.; and Lu, C. X. 2024. RadarOcc: Robust 3D Occupancy Prediction with 4D Imaging Radar. In *Advances in Neural Information Processing Systems*, volume 37, 101589–101617. Curran Associates, Inc.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Fan, C.; Zhang, S.; Liu, K.; Wang, S.; Yang, Z.; and Wang, W. 2024. Enhancing mmWave Radar Point Cloud via Visual-inertial Supervision. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 9010–9017.

Fent, F.; Palffy, A.; and Caesar, H. 2024. DPFT: Dual Perspective Fusion Transformer for Camera-Radar-based Object Detection. *IEEE Transactions on Intelligent Vehicles*, 1–11.

Gu, X.; Wang, Y.; Wu, C.; Lee, Y. J.; and Wang, P. 2019. HPLFlowNet: Hierarchical Permutohedral Lattice FlowNet for Scene Flow Estimation on Large-Scale Point Clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3249–3258.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6546–6555.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *9th International Conference on Learning Representations, ICLR, 2015*.

Kong, S.-H.; Paek, D.-H.; and Lee, S. 2024. RTNH+: Enhanced 4D Radar Object Detection Network using Two-Level Preprocessing and Vertical Encoding. *IEEE Transactions on Intelligent Vehicles*, 1–14.

Li, J.; Huang, W.; Wang, Z.; Liang, W.; Di, H.; and Liu, F. 2025. FloNa: floor plan guided embodied visual navigation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'25. ISBN 978-1-57735-897-8.

Li, Y.; Yu, Z.; Choy, C.; Xiao, C.; Alvarez, J. M.; Fidler, S.; Feng, C.; and Anandkumar, A. 2023. VoxFormer: Sparse Voxel Transformer for Camera-Based 3D Semantic Scene Completion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9087–9098.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.

Pace, P.; and Taylor, L. 1994. False alarm analysis of the envelope detection GO-CFAR processor. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3): 848–864.

Paek, D.-H.; Kong, S.-H.; and Wijaya, K. T. 2022. K-Radar: 4D Radar Object Detection for Autonomous Driving in Various Weather Conditions. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Paek, D.-H.; Kong, S.-H.; and Wijaya, K. T. 2023. Enhanced K-Radar: Optimal Density Reduction to Improve Detection Performance and Accessibility of 4D Radar Tensor-based Object Detection. arXiv:2303.06342.

Palffy, A.; Pool, E.; Baratam, S.; Kooij, J. F. P.; and Gavrila, D. M. 2022. Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset. *IEEE Robotics and Automation Letters*, 7(2): 4961–4968.

Qingwen, Z.; Yi, Y.; Peizheng, L.; Olov, A.; and Patric, J. 2024. SeFlow: A Self-supervised Scene Flow Method in Autonomous Driving. In *ECCV*, 353–369.

Roldan, I.; Palffy, A.; Kooij, J. F. P.; Gavrila, D. M.; Fioranelli, F.; and Yarovoy, A. 2024. A Deep Automotive Radar Detector Using the RaDelft Dataset. *IEEE Transactions on Radar Systems*, 2: 1062–1075.

Shen, Y.; Hui, L.; Xie, J.; and Yang, J. 2023. Self-Supervised 3D Scene Flow Estimation Guided by Superpoints. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5271–5280.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8934–8943.

Venon, A.; Dupuis, Y.; Vasseur, P.; and Merriaux, P. 2022. Millimeter Wave FMCW RADARs for Perception, Recognition and Localization in Automotive Applications: A Survey. *IEEE Transactions on Intelligent Vehicles*, 7(3): 533–555.

Wang, H.; Wang, W.; Liang, W.; Hoi, S. C. H.; Shen, J.; and Gool, L. V. 2022. Active Perception for Visual-Language Navigation. *International Journal of Computer Vision*, 607–625.

Wang, Z.; Wang, H.; and Liang, W. 2024. Mastering Scene Rearrangement with Expert-Assisted Curriculum Learning and Adaptive Trade-Off Tree-Search. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8039–8046.

Yin, Z.; and Shi, J. 2018. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1983–1992.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.