MmWave Radar Perception Learning using Pervasive Visual-Inertial Supervision

Kezhong Liu[1], Yiwen Zhou[1], Mozi Chen[1], Jianhua He[3], Jingao Xu[4], Zheng Yang[5], Chris Xiao Xuan Lu[6], Shengkai Zhang[1,2,*]

***Abstract*—This paper introduces a radar perception learning framework guided by data collected from commonly equipped visual-inertial (VI) sensor suites on smart vehicles. Unlike existing approaches that rely on dense point clouds from 3D LiDARs, which are costly and not widely deployed, this method leverages the broader availability of VI data. However, visual images alone lack the ability to capture the three-dimensional motion of moving targets, which limits their effectiveness in supervising motion-related tasks. To overcome this limitation, the framework integrates multiple perception tasks such as odometry estimation, motion segmentation, and scene flow prediction into a unified learning process. The first component is an odometry estimation module that combines deterministic ego-motion models with data-driven learning results. This fusion helps accurately infer the scene flow of static background points while minimizing drift. The second component is a supervision signal extraction module that aligns optical and millimeter-wave radar measurements to guide the learning of radar scene flow and rigid transformations. This module improves the reliability of dynamic point supervision through joint constraints across sensing modalities. The third component introduces a feature-selection module designed for cross-modal learning. It enhances the accuracy of motion segmentation and enforces consistency between odometry and scene flow, resulting in more coherent radar perception outputs. Experimental evaluations show that this framework achieves superior performance in challenging conditions such as smoke-obscured environments. It surpasses state-of-the-art (SOTA) methods that depend on high-cost LiDAR systems.**

***Index Terms*—Scene flow, 4D mmWave radar, cross-modal learning, SLAM.**

I. Introduction

Millimeter-wave (mmWave) radar enhances the robustness of autonomous driving perception in adverse environments and plays an important role in ensuring the safety and reliability of large-scale deployment of intelligent transportation systems. Its resilience to environmental degradations makes it a compelling alternative to vision-based sensors in safety critical scenarios. However, the limited sensing resolution of mmWave radar manifested as sparse and noisy point clouds poses substantial challenges for key perception tasks including odometry estimation, motion segmentation, and scene flow estimation. To mitigate these limitations, recent methods typically adopt a cross-modal learning paradigm in which radar perception models are supervised using dense 3D point clouds from high-end LiDAR sensors [2]–[5]. While such supervision can effectively capture the 3D motion of objects and yield accurate training signals, this approach suffers from high deployment cost, limited sensor availability, and a strong reliance on large-scale annotated datasets to ensure generalization across diverse driving conditions.

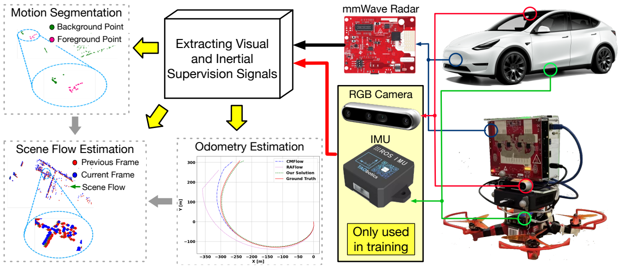

Fig. 1. The proposed system utilizes visual and inertial data to generate supervision signals for training the mmWave radar perception model. This model supports tasks including motion segmentation, scene flow estimation, and odometry estimation. Notably, visual and inertial inputs are needed only during the training phase.

In the industry, several approaches are used to collect training data. One involves deploying a fleet of specialized data collection vehicles for long-term operation. The other utilizes every user's vehicle to gather data, *e.g.*, Tesla. In addition to these real-world sources, companies also rely on synthetic data generation techniques, including simulation, procedural environments, and photorealistic rendering, together with domain adaptation, to produce large-scale, diverse, and controlled training sets. However, these synthetic pipelines still need to be seeded and validated with high-quality real-world data, which is often obtained using expensive sensor suites such as high-channel LiDAR and RTK-GPS, and is therefore difficult to scale. For this reason, methods that can exploit large amounts of crowdsourced data, even from lower cost sensor configurations, remain crucial for efficiently expanding training corpora. Note that LiDAR based perception can degrade significantly under adverse weather such as fog, rain and snow, as well as under strong vibrations, whereas mmWave radars generally maintain more stable detections in such conditions [6], [7]. At the same time, cameras have become pervasive in production vehicles, which makes the

Part of this work has been presented at IEEE IROS 2025 [1]

Authors[1] are with State Key Laboratory of Maritime Technology and Safety, School of Navigation, Wuhan University of Technology, Wuhan, China. {kzliu, zyw293423, chenmz}@whut.edu.cn

Author[2] is also with School of Information Engineering, Wuhan University of Technology, Wuhan, China. shengkai@whut.edu.cn

Author[3] is with University of Essex, Colchester, U.K. j.he@essex.ac.uk

Author[4] is with Carnegie Mellon University, Pittsburgh, USA. jingaox@andrew.cmu.edu

Author[5] is with Tsinghua University, Beijing, China. hmilyyz@gmail.com

Author[6] is with University College London, London, U.K. xiaoxuan.lu@ucl.ac.uk

*Corresponding author: Shengkai Zhang (shengkai@whut.edu.cn).

combination of radar and vision particularly attractive for robust perception. Thus, guiding the radar perception learning through images is more favorable for crowdsourcing training data from cameras. However, monocular 2D images lack direct 3D information [8]–[10], which makes it challenging to estimate the motion of moving points in three-dimensional space. This limitation hinders both scene flow estimation and motion segmentation tasks.

This study proposes VISC+, a unified framework for mmWave radar perception, leveraging supervision from commonly deployed visual-inertial (VI) sensors. As illustrated in Fig. 1, VISC+ jointly learns the tightly coupled radar perception tasks of odometry estimation, motion segmentation, and scene flow estimation by leveraging their mutual dependencies and complementary supervision. The framework employs self-supervised learning, leveraging visual and inertial inputs to provide cross-modal supervision. At test time, the model relies solely on radar point clouds, making it well-suited for deployment in cost-constrained and vision-degraded environments, such as fog or smoke. The core idea of VISC+ is to leverage the inherent coupling between perception tasks to compensate for missing or noisy supervision. For instance, ego-motion estimates from VI-SLAM provide frame-to-frame alignment and can supervise the scene flow of background points. These background flows, in turn, offer cues for separating static and dynamic regions in the radar scans. For dynamic points, the radar's relative radial velocity (RRV) and the visual optical flow offer complementary motion cues. These cues are jointly constrained across consecutive frames to recover the complete 3D scene flow. Through this design, VISC+ transforms sparse and modality-limited sensor data into rich cross-task supervision, enabling accurate radar perception without requiring any ground-truth labels or expensive sensors during training.

While the coupling of radar perception tasks provides opportunities for supervision, constructing reliable cross-modal guidance from VI data introduces three main challenges. First, visual-inertial odometry tends to accumulate drift in the absence of loop closure or GNSS, which compromises the supervision accuracy for static-point scene flow. Second, the motion of dynamic points is only partially observable. Optical flow offers 2D motion in the image plane, and radar provides radial velocity measurements, but neither alone is sufficient to recover complete 3D trajectories. This limitation makes it difficult to generate reliable supervision for dynamic scene flow. Third, visual and inertial signals must be carefully transformed into effective cross-modal supervision. Without dense 3D annotations, it remains challenging to use them to guide radar-based odometry, segmentation, and scene flow estimation in a unified framework.

To address the above challenges, VISC+ incorporates three key modules. First, a recursive sensor fusion module compensates for the drift in VI odometry by combining kinematic IMU integration with a learned motion model trained via dead reckoning. This approach produces a more stable estimation of ego-motion, which in turn enhances the supervision of static scene flow. Second, a supervision extraction module integrates optical flow, RRV, and geometric information across frames to guide the learning of dynamic scene flow. Third, a feature-selection cross-modal learning module enhances motion segmentation by transferring background-consistent visual features to the radar domain, while enforcing consistency across tasks. Together, these modules enable accurate and self-supervised radar perception using only visual-inertial data during training.

Contributions. Our method enhances mmWave radar perception with the assistance of widely available VI sensor data through: 1) Drift-free odometry estimation. We propose a recursive sensor fusion module that integrates kinematic model-based IMU integration with a learned neural motion model to reduce temporal drift in VI odometry. 2) Supervision signal extraction from optical and mmWave data. For dynamic points, we develop a supervision module that leverages optical flow from images and radar's RRV to generate pseudo ground-truth scene flow. 3) Feature-selection cross-modal learning. We introduce a learning strategy that selects background visual features from segmented images to supervise radar motion segmentation. By enforcing consistency among odometry, segmentation, and scene flow, this module jointly refines the overall radar perception. We validate VISC+ on both synthetic (CARLA) and real-world datasets collected with a custom sensor platform. Extensive experiments demonstrate that VISC+ consistently outperforms LiDAR-supervised baselines in challenging scenarios, including dense smoke environments, thereby highlighting the potential of low-cost VI sensors for scalable radar perception learning.

II. Related Work

Perception learning by camera. Monocular 3D motion field perception is known to be a severely ill-posed problem [11]. To address this, many vision-based methods rely on external depth information, either from RGB-D sensors [12], [13] or stereo camera systems [14]. Some approaches also use monocular images combined with predicted depth maps [11], [15]. The availability of large-scale image datasets [16] has enabled highly accurate supervised models [14], but these methods often face challenges related to domain overfitting. To overcome the need for labeled data and improve generalization, unsupervised learning strategies have been developed [17], although they typically yield lower performance. Furthermore, most vision-based systems require favorable lighting conditions, which limits their robustness. As an alternative, LiDAR has been widely used in perception tasks due to its active light emission and strong adaptability to various environments.

Perception learning by LiDAR. LiDAR-generated 3D point clouds provide direct access to point-wise 3D motion fields. Recent supervised methods [18]–[20] have reached state-of-the-art accuracy in this area. However, these models rely on manually labeled scene flow data, which demands significant annotation effort. To reduce this burden, self-supervised approaches [4], [21] have been proposed, enabling model training without ground-truth labels. As expected, the lack of true annotations leads to a drop in accuracy for these methods. Despite their promising outcomes, LiDAR remains vulnerable in challenging weather conditions such as fog, heavy rain, and storms, which limits its reliability for smart vehicles.

Perception learning by mmWave radar. mmWave radar has gained increasing interest for its ability to provide full 3D perception and maintain reliability under harsh weather conditions [22]–[25]. Nevertheless, its short wavelength at the millimeter scale leads to limited environmental resolution, resulting in sparse point cloud outputs [26]–[31]. Given the challenges in labeling such sparse data, many existing methods adopt self-supervised learning strategies [32], [33]. Among them, CMFlow [32] stands out as a state-of-the-art model, benefiting from supervision derived through multiple redundant sensors such as cameras, high-end LiDAR, and RTK-GPS. However, the required high-performance LiDAR systems can cost several thousand dollars [34], and RTK-GPS depends on infrastructure like reference station networks, which are not broadly deployed. As a result, CMFlow's training process relies on expensive setups mounted on vehicles operating in urban environments. This raises the cost of data collection and increases the risk of long-tail perception issues in autonomous driving. In contrast, our approach focuses on learning radar perception with affordable cameras and IMUs, which are widely integrated into commercial vehicles.

III. System Design of VISC+

A. Overview

Typical radar perceptional tasks include odometry estimation, motion segmentation, and scene flow estimation. Odometry estimation aims to compute a vehicle's position and orientation. The goal of motion segmentation is to label points in a scene if they are stationary or moving. Scene flow estimation aims to compute the 3D non-rigid motion field of a scene [35]. In our context, mmWave radar provides point clouds to describe the scene. The inputs are two consecutive point clouds $\mathcal{F}_1$ and $\mathcal{F}_2$. Each point $\mathbf{f}_i \in \mathcal{F}_1$, $\mathbf{f}_i = \{\mathbf{s}_i, c_i\}, i = 1, 2, \cdots, N_1$, where N_1 denotes the number of points in $\mathcal{F}_1$ and $\mathbf{s}_i \in \mathbb{R}^3$ is the 3D position of point i. $c_i \in \mathbb{R}$ denotes the RRV of radar points. The scene flow $\mathbf{L}$ is a set of point-wise 3D vectors with respect to the first point cloud. Each $\mathbf{l}_i \in \mathbb{R}^3$ in $\mathbf{L}$, $i = 1, 2, \cdots, N_1$, represents the translation of a point from frame $\mathcal{F}_1$ to frame $\mathcal{F}_2$, generating a corresponding point $\mathbf{s}'_i = \mathbf{s}_i + \mathbf{l}_i$ in frame $\mathcal{F}_2$.

Fig. 2 illustrates the architecture of the VISC+ system, which comprises three main modules: a *recursive sensor fusion* module, an *optical-mmWave supervision* module, and a *feature-selection cross-modal learning* module. The system takes as input RGB images from a monocular camera, IMU data including linear acceleration and angular velocity, and point clouds from a mmWave radar. The processing begins by extracting point-wise latent features from two consecutive radar point clouds [32], [33]. These features are then fed into two separate network heads to estimate an initial scene flow and a preliminary motion segmentation map. Feature extraction is performed synchronously on the two consecutive radar frames by a shared-weight encoder, producing paired feature tensors. The two heads operate in parallel on these paired tensors, forwarding the initial scene flow and preliminary motion segmentation to the odometry module. Based on these outputs, the radar odometry between the two frames is computed using the Kabsch algorithm [36], which yields a rigid-body transformation. The estimated rigid transform is fed back to (i) align the second-frame points prior to flow refinement and (ii) enforce a transform-consistency constraint on the odometry estimation. As a result, we obtain coarse predictions for odometry, motion segmentation, and scene flow, all aligned to the reference frame $\mathcal{F}_1$ of the radar.

To supervise the above coarse estimates, we explore the opportunity from VI-SLAM technology. The recursive sensor fusion module (see §III-B) utilizes RGB images and IMU measurements to produce centimeter-level odometry, which serves as a constraint for radar-based pose estimation. To supervise the scene flow of moving points, the optical-mmWave supervision module combines optical flow with the radar's RRV, offering partial 3D motion information. These cues supervise the dynamic component of the scene flow (see §III-D), without altering the inference pipeline. These cross-modal cues are employed to guide the learning of the dynamic portion of the scene flow. In addition, with the delicate image segmentation from Segment Anything Model [37] and the 3D reconstruction from VI-SLAM [38], we provide more accurate motion segmentation labels from the feature-selection cross-modal learning module (refer to §III-C) to constrain the motion segmentation mask. Finally, the cross-modal refinement module combines the motion segmentation labels and the odometry estimation to constrain the scene flow estimation.

B. Recursive Visual-inertial Fusion

The odometry provided by VI-SLAM [38] already leverages the fusion of visual and inertial data. To further reduce temporal drift, one might consider introducing a third sensing modality. However, this can be avoided. Prior work on IMU-based dead reckoning [39]–[41] has shown that the IMU alone is capable of supporting independent odometry estimation, making it a viable solution without additional sensor requirements.

1) Inertial-learning Deep Neural Network: **Network architecture.** To estimate odometry from the IMU's sequential 3D acceleration and angular velocity readings, we adopt a recurrent neural network (RNN) structure, specifically a Long Short-Term Memory (LSTM) network, for temporal sequence modeling. This choice follows prior deep inertial odometry frameworks such as [40], [42], where LSTM-based encoders have been shown to effectively learn motion patterns and drift compensation from noisy IMU data. Though recent work has also explored Transformer-based architectures for IMU-only localization [43], [44]. These methods highlight the potential of attention mechanisms when the trajectory must be recovered from IMU alone. In contrast, our method operates in a visual-inertial setting, where a model-based VI odometry backbone already provides strong geometric constraints, and the IMU branch is used as a learned complement. We therefore adopt a lightweight LSTM-based design for the IMU network and focus our contributions on the fusion of learned IMU odometry with model-based VI odometry. Based on the effectiveness reported in [45], we utilize a fully connected LSTM (FC-LSTM) to capture temporal dependencies. The architecture comprises two stacked LSTM layers followed by a fully connected layer. We select a two-layer configuration as it offers improved

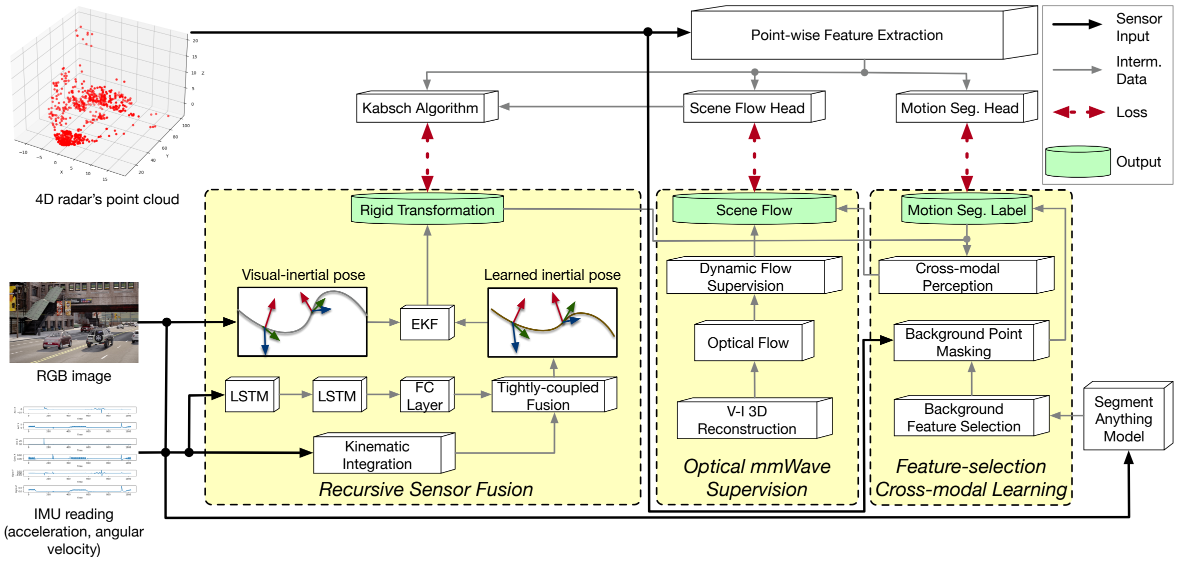

Fig. 2. Overview of the VISC+ system, with the newly proposed components marked in yellow.

performance over a single layer, while adding a third layer yields negligible gains, consistent with observations in [46], [47].

Each LSTM layer consists of 64 hidden units, with the state of the k-th unit represented as $\mathbf{L}_k$ for $k = 1, 2, \dots, 64$. Each $\mathbf{L}_k$ processes the IMU input data $\hat{\boldsymbol{a}}_k$ and $\hat{\boldsymbol{\omega}}_k$, along with the output from its preceding unit, to capture temporal features. IMU signals are de-biased using estimated constant biases, rotated to the radar frame for gravity removal, and per-axis normalized. Segments are aligned to camera frames during training and to the two radar timestamps at inference to match the network input window. The combined output of the LSTM layers is then projected into a 1×12 vector and reshaped into a 3×4 matrix $[R \mid \boldsymbol{t}]$ that represents the relative pose between consecutive radar frames in the first-frame radar reference. The network predicts rotation and translation only, no velocity terms are included.

Loss function. The network is trained to predict 3D translations and their associated uncertainties between adjacent image frames. To this end, we adopt a hybrid loss function that includes both the mean squared error (MSE) loss and the Gaussian maximum likelihood (ML) loss during training.

$$\mathcal{L}_{\text{MSE}}(\boldsymbol{t}, \hat{\boldsymbol{t}}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|^2, \tag{1}$$

$$\mathcal{L}_{\text{ML}}(\boldsymbol{t}, \hat{\boldsymbol{\Gamma}}, \hat{\boldsymbol{t}}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left(\log\det(\hat{\boldsymbol{\Gamma}}_i) + \|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|^2 \right), \tag{2}$$

where $\boldsymbol{t} = \boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_N$ represent the ground-truth 3D translations obtained from the VI-SLAM system. Here, N denotes the total number of IMU segments aligned with visual frames in the training set. The predicted translations and their corresponding uncertainty estimates from the neural network are denoted as $\hat{\boldsymbol{t}} = \hat{\boldsymbol{t}}_1, \hat{\boldsymbol{t}}_2, \cdots, \hat{\boldsymbol{t}}_N$ and $\hat{\boldsymbol{\Gamma}} = \hat{\boldsymbol{\Gamma}}_1, \hat{\boldsymbol{\Gamma}}_2, \cdots, \hat{\boldsymbol{\Gamma}}_N$, respectively. For each segment i $(1 \leq i \leq N)$, $\hat{\boldsymbol{\Gamma}}_i$ is a 3×3 covariance matrix describing the uncertainty of the predicted translation. Following the diagonal parameterization method in [40], the network outputs a vector of coefficients $\hat{\boldsymbol{c}}_i = [\hat{c}_{ix}, \hat{c}_{iy}, \hat{c}_{iz}]^\top$, and the covariance is computed as $\hat{\boldsymbol{\Gamma}}_i = \text{diag}(e^{2\hat{c}_{ix}}, e^{2\hat{c}_{iy}}, e^{2\hat{c}_{iz}})$.

2) Two-stage Sensor Fusion: **Tightly-coupled fusion.** A graph-based optimization approach is utilized to tightly fuse the results from the IMU kinematics and the statistical learning model, which has been shown to enhance estimation accuracy [38], [48]–[50]. We form a diagonal translation covariance $\Sigma_t = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$ from the network output by comparing $\boldsymbol{t}$ with IMU preintegration over the same window and taking an exponential moving average of per-axis residuals, followed by floor clipping to avoid overconfidence, and a Huber loss is applied. A rotation weight is obtained from the geodesic distance between the network rotation R and the Kabsch estimate, and mapped to an equivalent variance used for the rotation factor. Factor weights are normalized before graph optimization.

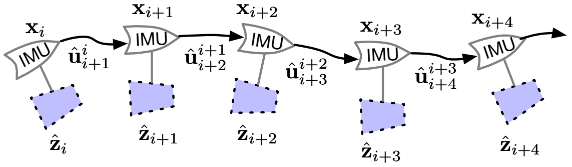

Fig. 3. Graph-based optimization.

Fig. 3 shows the graph representation of our odometry estimation problem.

Let $\mathbf{x}_i$ represent the vehicle state at timestamp i. At each step i, the deep neural network produces an output $\hat{\mathbf{z}}_i$, which contains both the estimated 3D translation $\hat{\boldsymbol{t}}_i$ and its associated uncertainty $\hat{\boldsymbol{\Gamma}}i$. Since the IMU operates at a higher frequency (100 Hz) compared to the camera (20 Hz), the IMU data is segmented based on image frame intervals. We integrate the IMU readings within each segment to compute the relative motion $\hat{\mathbf{u}}i + 1^i$ between two successive image frames.

To enable real-time computation, we adopt a sliding window strategy that retains m consecutive states during the optimization process [50]. The details of the sliding window strategy has been thoroughly resolved in our conference publication [1]. The overall state vector is defined as $\boldsymbol{\mathcal{X}} = \left[\mathbf{p}_1^0; \mathbf{p}_2^0; \cdots; \mathbf{p}_m^0\right]$, where each $\mathbf{p}_i^0$ represents the position at time i relative to the initial position indexed by 0. The optimization aims to compute the maximum a posteriori (MAP) estimate by minimizing the Mahalanobis distance of all residuals derived from sensor measurements:

$$\min_{\boldsymbol{\mathcal{X}}} \left\{ \sum_{i \in \mathcal{N}} \left\| \hat{\mathbf{z}}_i - \mathbf{H}_i \boldsymbol{\mathcal{X}} \right\|_{\mathbf{P}_i}^2 + \sum_{j \in \mathcal{I}} \left\| \hat{\mathbf{u}}_{j+1}^j - \mathbf{H}_{j+1}^j \boldsymbol{\mathcal{X}} \right\|_{\mathbf{P}_{j+1}^j}^2 \right\}, \tag{3}$$

where $\mathbf{H}_i$ and $\mathbf{H}_{j+1}^j$ are information matrices for neural network observations and IMU kinematic predictions, respectively. $\mathcal{N}$ is the set of network-based translations and uncertainties. $\mathcal{I}$ denotes the set of raw IMU measurements. The Mahalanobis norm in the objective function takes into account the correlations of the data set, which are key for any high-precision inertial-based autonomous system. where $\mathbf{H}_i$ denotes the information matrix associated with the neural network outputs, while $\mathbf{H}_{j+1}^j$ corresponds to the IMU-based kinematic constraints. The sets $\mathcal{N}$ and $\mathcal{I}$ represent the network-predicted translations with uncertainties and the raw IMU measurements, respectively. The Mahalanobis norm used in the cost function captures correlations within the data, which is essential for achieving high-precision inertial navigation. For each network prediction $\hat{\mathbf{z}}_i$, the geometric constraint involves only the translation component $\hat{\boldsymbol{t}}_i$ between states $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. This leads to a constraint of the form $\hat{\boldsymbol{t}}_i = \mathbf{p}_{i+1}^0 - \mathbf{p}_i^0$, from which the information matrix $\mathbf{H}_i$ can be derived. The corresponding covariance $\mathbf{P}_i$ is initialized based on the predicted uncertainty $\hat{\boldsymbol{\Gamma}}_i$. On the IMU side, given the accelerometer and gyroscope measurements $\hat{\boldsymbol{a}}_t$ and $\hat{\boldsymbol{\omega}}_t$ at time t, one can compute the relative motion estimate $\hat{\mathbf{u}}_{j+1}^j$ along with its associated information matrix $\mathbf{H}_{j+1}^j$. The covariance $\mathbf{P}_{j+1}^j$ is obtained via discrete-time uncertainty propagation over the interval Δt_j [38].

$$\mathbf{H}_i = [\underbrace{0, \ldots, 0}_{(i-1)\text{ zeros}}, -1, 1, 0, \ldots, 0]_m. \tag{4}$$

All individual matrices $\mathbf{H}_i$ from the neural network outputs are aggregated to construct the complete information matrix corresponding to the network-based observations. The initial covariance $\mathbf{P}_i$ is set using the predicted uncertainty $\hat{\boldsymbol{\Gamma}}_i$ provided by the network. Note that $\mathbf{H}_i$ will be a zero vector when $i\%2 = 0$ as the network does not observe the vehicle's velocity.

For the IMU side, given the non-gravitational acceleration $\hat{\boldsymbol{a}}_t$ and angular velocity $\hat{\boldsymbol{\omega}}_t$ at time t, the relative motion between frames, denoted as $\hat{\mathbf{u}}_{j+1}^j$, can be derived as part of the kinematic model.

$$\begin{aligned} \hat{\mathbf{u}}_{j+1}^j = \begin{bmatrix} \hat{\boldsymbol{p}}_{j+1}^j \\ \hat{\boldsymbol{v}}_{j+1}^j \end{bmatrix} &= \begin{bmatrix} \iint_{t \in [j, j+1]} \hat{\boldsymbol{R}}_t^j \hat{\boldsymbol{a}}_t \, \mathrm{d}t^2 \\ \int_{t \in [j, j+1]} \hat{\boldsymbol{R}}_t^j \hat{\boldsymbol{a}}_t \, \mathrm{d}t \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{p}_{j+1}^j - \mathbf{p}_j^j - \mathbf{v}_j^j \Delta t_j \\ \mathbf{v}_{j+1}^j - \mathbf{v}_j^j \end{bmatrix} = \mathbf{H}_{j+1}^j \boldsymbol{\mathcal{X}}, \end{aligned} \tag{5}$$

where the vehicle's relative orientation $\hat{\boldsymbol{R}}_{j+1}^j$ is computed as $\hat{\boldsymbol{R}}_{j+1}^j = \int_{t \in [j, j+1]} \boldsymbol{R}_t^j \lfloor \hat{\boldsymbol{\omega}}_t \times \rfloor, \mathrm{d}t$, and $\lfloor \hat{\boldsymbol{\omega}}_t \times \rfloor$ denotes the skew-symmetric matrix derived from the angular velocity $\hat{\boldsymbol{\omega}}_t$. To mitigate drift in heading estimation, external sensors such as a compass can be used for correction. The time duration between two consecutive states is denoted as Δt_j, over which the associated covariance $\mathbf{P}_{j+1}^j$ is obtained via discrete-time uncertainty propagation. At this point, all the information matrices of the system have been explicitly defined. We then use Ceres Solver to solve the optimization problem.

Loosely-coupled fusion. At time step i, we obtain the state ${}^I\hat{\boldsymbol{x}}_i$, which consists of the position ${}^I\hat{\mathbf{p}}_i^0$ estimated from the learned-inertial module and the heading ${}^I\hat{\boldsymbol{q}}_i^0$ adjusted using the IMU compass. Here, ${}^I\hat{\boldsymbol{q}}_i^0$ denotes the quaternion that represents rotational orientation. These estimates are utilized to correct the temporal drift present in the visual-inertial odometry ${}^V\hat{\boldsymbol{x}}_i = [{}^V\hat{\mathbf{p}}_i^0, {}^V\hat{\boldsymbol{q}}_i^0]$, as computed by VINS [38]. Since VINS [38] has already estimated the odometry, the state propagation is trivial in our fusion. Given a covariance matrix of sensing noise $\boldsymbol{W}$, the state covariance $\boldsymbol{P}_k$ is propagated as $\boldsymbol{P}_{k+1} = \boldsymbol{P}_k + \boldsymbol{W}$. Then, we derive the measurement update when taking a learned-inertial odometry. Since the rotation has to be linearized on the manifold, we apply error-based filtering. We define the odometry vector $\boldsymbol{X}$ and the error-state vector $\tilde{\boldsymbol{X}}$ as

$$\boldsymbol{X} = (\boldsymbol{q}^0, \mathbf{p}^0), \ \tilde{\boldsymbol{X}} \ \ = (\delta\boldsymbol{\theta}, \delta\mathbf{p}), \tag{6}$$

where $\delta\boldsymbol{\theta} \ =^V \ \hat{\boldsymbol{q}}^0 \otimes \left({}^I\hat{\boldsymbol{q}}^0\right)^{-1}$, $\otimes$ denotes the quaternion multiplication. $\delta\mathbf{p} \ =^V \ \hat{\mathbf{p}}^0 -^I \hat{\mathbf{p}}^0$. Then the measurement function $h(\boldsymbol{X})$ with respect to state i can be written as,

$$h(\boldsymbol{X}) = \left[\boldsymbol{q}_i^0, \mathbf{p}_i^0\right]^\top = \left[{}^I\hat{\boldsymbol{q}}_i^0, {}^I\hat{\mathbf{p}}_i^0\right]^\top + \left[{}^I\boldsymbol{n}_{pi}, \mathbf{0}\right]^\top. \tag{7}$$

where ${}^I\boldsymbol{n}_{pi}$ is the random noise following the normal distribution $\mathcal{N} \sim \left(0, \hat{\boldsymbol{\Gamma}}_i\right)$ and $\hat{\boldsymbol{\Gamma}}_i$ is provided by the inertial learning module. Then we take the partial derivative of $h(\boldsymbol{X})$ with the error-state variable to obtain the linearized measurement matrix H. Finally, we use H and $[\hat{\boldsymbol{\Gamma}}_i, \mathbf{0}]^\top$ to compute the Kalman gain and update the odometry $\boldsymbol{X}$ and the covariance matrix $\boldsymbol{P}$. In practice, $\boldsymbol{W}$ is set as a diagonal covariance that reflects the expected magnitude of residual sensor noise and modeling error and is kept fixed across all experiments, while $\hat{\boldsymbol{\Gamma}}_i$ is directly taken from the inertial learning module without additional tuning. The remaining extended Kalman filter update equations follow the standard form and are omitted here for brevity.

C. Feature-selection Cross-modal Supervision

To better supervise the perception model, we combine multiple related tasks to suppress the noisy supervision signals from cross-modal sensing. Based on the state-of-the-art point-wise feature extraction method [32], consider a radar frame containing N points. The network produces a coarse scene flow estimation $\hat{\mathbf{L}}^c = \mathbf{l}_i \in \mathbb{R}^3{}_{i=1}^N$ along with a motion segmentation probability map $\hat{\mathbf{M}}^c = \hat{m}_i \in [0,1]_{i=1}^N$. A value of $\hat{m}_i < 0.5$ indicates that point i is considered static. With both the coarse flow $\hat{\mathbf{L}}^c$ and the segmentation confidence $\hat{\mathbf{M}}^c$, we estimate the radar odometry $\hat{\mathbf{T}} \in \mathrm{SE}(3)$ using the Kabsch algorithm [36].

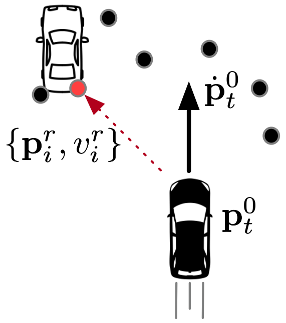

Fig. 4. An illustration of absolute velocity computation.

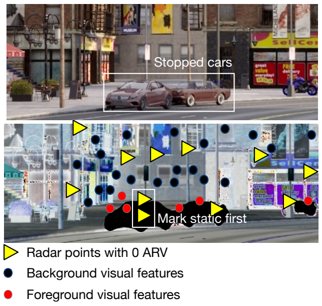

Fig. 5. Leveraging background visual features to find more background radar points.

It is important to note that this odometry only provides valid scene flow vectors for static background points. Better motion segmentation to distinguish background points will benefit the scene flow estimation. Thus, the three tasks, *i.e.*, odometry, motion segmentation, and scene flow, are tightly coupled. In the following, we elaborate on the design for better motion segmentation.

1) Background Point Masking: Intuitively, all static points on the background should have absolute radial velocities (ARVs) close to 0. We denote the ARV of point i as v_i^a. With the estimated odometry, as shown in Fig. 4, at time t, its velocity with respect to the initial frame can be trivially deduced by the first derivative of position $\mathbf{p}_t^0$. Meanwhile, the radar on the vehicle reads a point-cloud frame $\mathcal{F}$. Each point $\mathbf{f}_i \in \mathcal{F}$ contains the point's relative position $\mathbf{s}_i^r$ and the RRV v_i^r. We compute the relative radial velocity in the first radar frame. Ego motion is compensated using the Kabsch transform between the two frames. For each point, we remove the ego-induced radial component from the raw Doppler and obtain a residual RRV used for masking, and the RRV v_i^r. Then v_i^a writes:

$$v_i^a = v_i^r + \frac{\mathbf{s}_i^r \cdot \dot{\mathbf{p}}_t^0}{\|\mathbf{s}_i^r\|}. \tag{8}$$

We set a small threshold ϵ_v. Considering the rigid transformation accuracy, $\epsilon_v = 0.1$ m/s in our model. The pseudo motion segmentation label of point i is static, *i.e.*, $m_i = 0$, when $v_i^a < \epsilon_v$. We put a static radar point into a set $\mathbf{M}^b$. Note that this motion segmentation principle will falsely mark static labels on tangentially moving points. To avoid mislabeling tangential movers as static, we apply short temporal voting across three consecutive frames and require consistency with a locally estimated rigid motion. A point is marked as background only when both conditions are satisfied.

To further constrain the motion segmentation, we leverage the 3D reconstruction of VI-SLAM. Existing LiDAR-based solutions [51] can track foreground points with a dense point cloud. However, the VI 3D reconstruction [38], [52] can only track background static points. To address this issue, we resort to advanced image segmentation tools [37], [53] to find out the visual features that have no explicit object label. We segment objects which are likely to move, *e.g.*, vehicles, pedestrians, and animals. Features without assigning valid labels are marked as background points, forming a background set $\mathbf{V}^{\text{bg}}$.

We first use $\mathbf{V}^{\text{bg}}$ to mark radar points whose $v_i^a < \epsilon_v$ as background points, as shown in Fig. 5. Then, we exploit an intuition that static radar points should be spatially related to background visual features in $\mathbf{V}^{\text{bg}}$. For ease of processing, we segment the radar points frame by frame. For two consecutive radar frames $\mathbf{S}_1$ and $\mathbf{S}_2$, VI-SLAM [38] can easily compute the frame transformation and obtain the 3D reconstruction with respect to frame $\mathbf{S}_1$. In other words, we transform the 3D positions of background visual features to the newest frame, aligning with the frame of the radar's new reading. Thus, for each background point i in $\mathbf{M}^b$, we find its top-k closest visual features in $\mathbf{V}^{\text{bg}}$, where k is a hyper-parameter in our model. We choose $k = 4$ through tests (refer to § IV-A4). A radar point is accepted only if the mean of the k distances is below a threshold T_d and the sample variance is below T_v. The thresholds are chosen once on a small validation split for each dataset and remain fixed during testing. If the mean or the variance of these k distances is too big, typically determined by thresholding, we exclude this point from $\mathbf{M}^b$. If multiple radar points compete for the same visual feature, we keep the pair with the highest score and discard the others. The score is $\exp(-d/s) \cdot p_{\text{vis}}$, where d is the mean distance and p_{vis} is the visual mask confidence. This score becomes the confidence weight.

With accurate background labels $\mathbf{M}^b$, we cannot simply mark other radar points in the frame as foreground points due to the ghost points generated by multipath. Similar to the criterion of labeling background points, we assign points foreground labels, *i.e.*, $m_i = 1$ for point i, by the top-k strategy applying to the foreground visual features $\mathbf{V}^{\text{fg}}$ from image segmentation. Then we obtain a set of foreground points $\mathbf{M}^f$. Note that $\mathbf{M}^b \cap \mathbf{M}^f$ may not cover all radar points in a frame. For any point $\mathbf{s}_k \notin \mathbf{M}^b$ and $\notin \mathbf{M}^f$, it may be a ghost point that randomly comes out from nowhere. Thus, we randomly set its label. We compute the closest distance d_k^f to a foreground visual feature and the closest distance d_k^b to a background visual feature. Then we label it as a background point with a probability of $p_k = \frac{d_k^b}{d_k^b + d_k^f}$.

2) Cross-modal Perception Refinement: **Motion segmentation refinement**. Now we have obtained a moving probability map $\hat{\mathbf{M}}^c$ from the point-wise feature extraction [32] and the pseudo motion segmentation ground-truth label $\mathbf{M}^c$ from § III-C1. Our system aims to adjust the estimated moving probability to agree with $\mathbf{M}^c$ by the following loss:

$$\mathcal{L}_{\text{seg}} = \frac{1}{2}\left(\frac{\sum_{i=1}^{N}(1-m_i)\log(1-\hat{m}_i)}{\sum_{i=1}^{N}(1-m_i)} + \frac{\sum_{i=1}^{N} m_i \log \hat{m}_i}{\sum_{i=1}^{N} m_i}\right). \tag{9}$$

D. Optical-mmWave Supervision Extraction

Our proposed framework generates supervisory signals that facilitate self-supervised learning of both the radar's rigid-body transformation and scene flow. Once trained, the model is capable of estimating these quantities solely from sparse mmWave radar point clouds, even under challenging conditions such as smoke-filled environments.

Odometry refinement. As described in § III-B, we have access to reliable vehicle odometry. This allows for straightforward computation of the radar's pseudo ground-truth transformation $\mathbf{T} \in \mathrm{SE}(3)$ between two adjacent frames. Additionally, an estimated transformation $\hat{\mathbf{T}} \in \mathrm{SE}(3)$ can be obtained using the Kabsch algorithm [36], providing a basis for refining odometry accuracy. Consider two radar frames, $\mathcal{F}_1$ and $\mathcal{F}_2$, with a known transformation from frame 1 to frame 2 given by $\mathbf{T}_1^2 = \begin{bmatrix} \boldsymbol{R}_1^2 & \boldsymbol{t}_1^2 \\ \mathbf{0} & 1 \end{bmatrix}$. A point s_{i1} observed in $\mathcal{F}_1$ can be mapped to the coordinate system of $\mathcal{F}2$ using the transformation as follows:

$$\mathrm{s}'_{i2} = \boldsymbol{R}_1^2 \mathrm{s}_i^1 + \boldsymbol{t}_1^2. \tag{10}$$

The objective of our system is to refine the estimated odometry $\hat{\mathbf{T}} = \begin{bmatrix} \hat{\boldsymbol{R}} & \hat{\boldsymbol{t}} \\ \mathbf{0} & 1 \end{bmatrix}$ to closely match the pseudo ground-truth transformation $\mathbf{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0} & 1 \end{bmatrix}$. To achieve this, we define the following loss function:

$$\mathcal{L}_{\text{trans}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \left(\boldsymbol{R} \hat{\boldsymbol{R}}^{\top} - \mathbf{I}_3 \right) \mathrm{s}_{i1} + \boldsymbol{t} - \hat{\boldsymbol{t}} \right\|_2 . \tag{11}$$

Scene flow refinement. The improved estimation of rigid transformation and motion segmentation yields more reliable supervisory cues for learning the scene flow of background points. In this work, we adopt the refinement strategy proposed in [32]. Concretely, we incorporate two types of loss terms: one derived from optical flow labels extracted from RGB images, denoted as $\mathcal{L}_{\text{opt}}$, and the other based on the radar point cloud itself, denoted as $\mathcal{L}_{\text{self}}$, as introduced in [32]. The total scene flow loss is then defined as:

$$\mathcal{L}_{\text{flow}} = \lambda_{\text{opt}} \mathcal{L}_{\text{opt}} + \mathcal{L}_{\text{self}}, \tag{12}$$

where λ_{opt} = 0.1 in experiments.

IV. System Implementation and Evaluation

VISC+ operates with inputs from a 4D mmWave radar, an RGB camera, and an IMU. The framework depends on raw inertial data, specifically accelerations and angular velocities, which are not provided in current public 4D radar datasets [54]–[58]. To address this limitation, we employ both synthetic data generated using the Carla simulator [59] and real-world data collected through a custom-built platform mounted on a drone.

Platform. As depicted in Fig. 6, the 4D radar system consists of a cascade of four TI AWR2243 modules, comprising 12 transmit and 16 receive antennas. Under the MIMO configuration, it supports a maximum detection range of 150 meters. The angular resolutions achieved are 1.4° in azimuth and 18° in elevation. For visual data acquisition, we utilize an Intel Realsense D435 RGB camera. Inertial readings are obtained from the high-precision IMU integrated within the CUAV v5+ flight controller. All sensor data streams are transmitted to an Intel NUC11TNKi5 unit equipped with a 2.6 GHz Intel Core i5 processor and 16 GB of RAM, running Ubuntu 20.04. A Robot Operating System (ROS) framework is employed to manage inter-sensor communication and synchronize the timestamped data. The collected dataset is subsequently transferred to a backend server for training purposes. All models are implemented in PyTorch 1.7.0 with Python 3.7. On a workstation equipped with an NVIDIA RTX 2060 GPU, the average inference time for one radar frame pair, including all network components, is approximately 42 ms, corresponding to about 23 frames per second.

Dataset. In the simulation environment, we equip a virtual vehicle with an RGB camera, an IMU, and a 4D mmWave radar to collect data across 8 distinct scenes. The resulting dataset contains 7119 frames, each with synchronized RGB images and radar point clouds. On average, around 80% of the objects in each frame are dynamic. The dataset also includes 23750 IMU sequences. The total recording time spans approximately 400 seconds, covering a trajectory of 2.53 kilometers. For real-world data collection, we conduct experiments in both outdoor and indoor environments. The outdoor dataset is captured along a roadway adjacent to our campus, while the indoor dataset is gathered within our laboratory. The outdoor portion comprises 5312 synchronized frames of images and radar point clouds, whereas the indoor set includes about 1328 frames. In terms of inertial data, we collect roughly 39600 IMU sequences outdoors and 9900 sequences indoors, with the entire trajectory exceeding 2.1 kilometers. All supervision signals derived from vision or IMU, including optical flow, SAM masks, and VINS odometry, are generated offline on the training split only. It is worth noting that, in both settings, approximately 60% of the visual targets are in motion.

Ground Truth Labeling. Although VISC+ is designed for self-supervised learning and does not rely on ground truth during training, annotated labels are still necessary for performance evaluation. Following the protocol in [32], we generate ground truth scene flow using object detection results (i.e., bounding boxes) along with accurate radar odometry. For the indoor dataset, ground truth odometry is obtained using the NOKOV motion capture system, which provides high-precision pose information. In outdoor environments, we utilize VINS-Fusion [1] to integrate RGB images, IMU data, and GPS signals for producing reliable odometry estimates. For static background points, their corresponding scene flow vectors are labeled based on the radar's ground truth motion.

Metrics. In line with the evaluation protocol from [33], we assess the performance of our framework across three sub-tasks using six metrics. For scene flow estimation, we adopt 1) the average end-point error (EPE, in meters), which measures the mean Euclidean distance between the predicted and ground-truth flow vectors; 2) the accuracy metrics $AccS$ and $AccR$, representing the proportion of points with $EPE <$ 0.05 or 0.1 meters, respectively, reflecting strict and relaxed accuracy thresholds. For motion segmentation, we use the mean intersection-over-union ($mIoU$), which computes the IoU between dynamic and static regions in each frame and averages the results across the dataset. Regarding odometry estimation, we report two standard metrics: the relative translation error (RTE) and the relative angular error (RAE), which quantify

[1] https://github.com/HKUST-Aerial-Robotics/VINS-Fusion

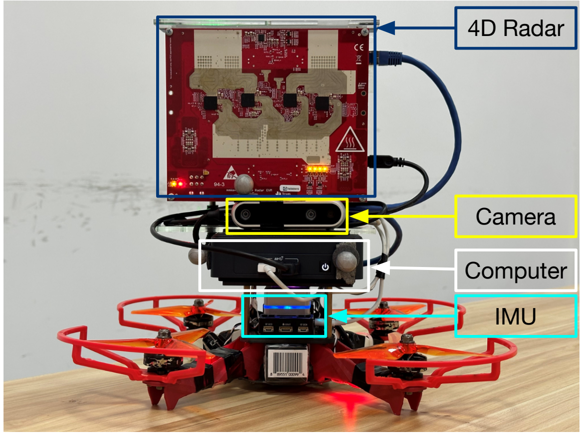

Fig. 6. The key components of our customized platform include a 4D radar, an RGB camera, an IMU, and a mini-computer.

the accuracy of estimated trajectories in both position and orientation.

Baselines. We compare our approach with two state-of-the-art radar scene flow methods: RAFlow [33] and CMFlow [32]. RAFlow utilizes radar point cloud radial velocities as implicit supervision to train the scene flow network. CMFlow, by contrast, requires high-precision external sensors such as LiDAR and RTK-GPS to obtain cross-modal supervisory signals. In our implementation, we adapt CMFlow by replacing LiDAR with depth point clouds reconstructed from a camera, and we substitute RTK-GPS with odometry results from VINS-Fusion.

We partitioned the dataset into training, testing, and validation sets in a ratio of $5:3:2$. In the experiments, we use the Adam optimizer to train all models. The initial learning rate is 0.001, exponentially decreasing by 0.9 per epoch.

A. Performance Evaluation

1) Overall Results: We conduct a comparative evaluation of VISC+ against existing baseline methods. As presented in Table I, our approach delivers results on par with CMFlow, despite utilizing only low-cost visual-inertial sensors. Notably, VISC+ shows clear improvements over the original VISC framework across both simulated and real-world scenarios. On the synthetic data, VISC+ improves $AccS$ by 3.7% and $AccR$ by 2.7%, while reducing the EPE by 1.4%. On the real-world data, VISC+ achieves a 5.1% increase in $AccS$ and a 1.2% reduction in EPE, demonstrating the effectiveness of the proposed feature-selection cross-modal learning module. Compared to CMFlow, which leverages high-precision LiDAR and RTK-GPS for supervision, VISC+ still shows a slight performance gap. For example, on the synthetic data, $AccS$ and $AccR$ are 1.2% and 0.6% lower, respectively, while the EPE remains within 0.7% of CMFlow. These results suggest that our method approaches the accuracy of high-cost sensor-based supervision, while being significantly more practical for large-scale deployment. We note that real-world performance may still be affected by temporal misalignment among sensors. To address this, we apply online temporal calibration using VINS [38] to correct the clock differences between the camera and IMU. In addition, radar frames are synchronized through the system clock of ROS. These steps help ensure the accuracy of cross-modal supervision during training.

TABLE I
METHOD COMPARISON

Method	Super-vision	EPE [m]↓	AccS↑	AccR↑
RAFlow (synth.)	Self	0.224	0.286	0.525
RAFlow (real.)	Self	0.283	0.224	0.475
CMFlow (synth.)	Cross	0.136	0.481	0.774
CMFlow (real.)	Cross	0.169	0.402	0.703
VISC (synth.)	Cross	0.139	0.458	0.749
VISC (real.)	Cross	0.172	0.375	0.711
VISC+ (synth.)	Cross	**0.137**	**0.475**	**0.769**
VISC+ (real.)	Cross	0.170	0.394	0.709

2) Ablation Study: We conduct ablation studies to investigate the impact of different modules on scene flow estimation, as shown in Table. II. The average translation error of VI-SLAM is 0.44 meters, and the average rotation error is $0.05°$. VI-SLAM shows the inferior performance of scene flow estimation due to the cumulative errors of ego-motions and the inaccurate position estimates of dynamic points. After incorporating the recursive sensor fusion module, the EPE decreases by 65.3% on the synthetic dataset, and by 62.7% on the real-world dataset. This indicates that the sensor fusion method effectively mitigates the drift in VI-SLAM and improves the accuracy of scene flow estimation. After incorporating the optical-mmWave supervision extraction module, the EPE decreased by 25.2% on synthetic data and by 33.0% on real-world data, indicating that cross-modal constraints from optical flow and radar enhance the supervision quality for dynamic scene flow estimation. Similarly, after adding the feature selection module, the EPE decreased by 34.7% on synthetic data and by 32.2% on real-world data, demonstrating the effectiveness of background point masking in VISC+.

TABLE II
ABLATION STUDY

Method	EPE [m]↓	AccS↑	AccR↑
VI−SLAM [38] (synth.)	0.424	0.009	0.011
VI−SLAM [38] (real-w.)	0.485	0.002	0.007
VI−SLAM+Recur. Fusion (synth.)	0.147	0.431	0.729
VI−SLAM+Recur. Fusion (real-w.)	0.181	0.346	0.629
VI−SLAM+Feature Select. (synth.)	0.277	0.257	0.489
VI−SLAM+Feature Select. (real-w.)	0.329	0.207	0.425
VI−SLAM+Opt-mm. Extract. (synth.)	0.317	0.236	0.435
VI−SLAM+Opt-mm. Extract. (real-w.)	0.325	0.182	0.408
VISC (synth.)	0.139	0.458	0.749
VISC (real-w.)	0.172	0.375	0.696
VISC+ (synth.)	**0.137**	**0.475**	**0.769**
VISC+ (real-w.)	0.170	0.394	0.709

Furthermore, when both modules are combined, the results show that compared to VI-SLAM, the EPE decreases by 67.2% on the Carla dataset. When using the real-world data, it

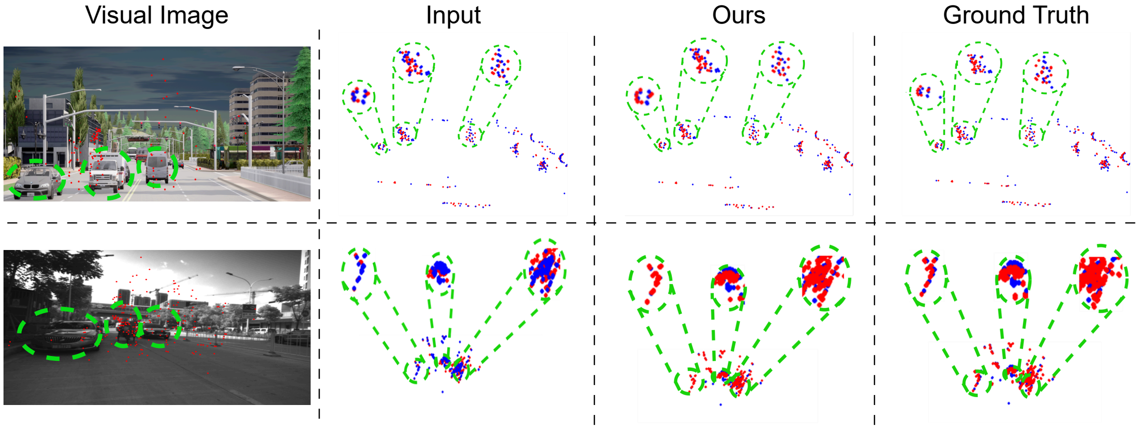

Fig. 7. Qualitative scene flow visualizations in two representative scenarios are illustrated. From left to right, the subfigures show: (1) RGB images overlaid with projected radar points; (2) two consecutive radar point clouds, where the first frame is shown in red and the second in blue; (3) the first point cloud warped using the predicted scene flow alongside the second frame; (4) the first point cloud aligned using the ground-truth scene flow and compared with the second frame. Dynamic objects are emphasized using green circles, and detailed regions are magnified for clarity.

shows similar results in that the EPE decreases by 64.5%. This indicates that motion segmentations and odometry estimation are coupled with scene flow estimation. Their more accurate results can benefit the scene flow estimation. In Fig. 7, we compare the results of predicted scene flow to the ground truth in synthetic and real-world scenarios.

Furthermore, when combining all three modules, our final method VISC+ achieves the best results: a 67.7% reduction in EPE compared to VI-SLAM on synthetic data, and a 64.9% reduction on real-world data. These results demonstrate that accurate odometry estimation, motion segmentation, and cross-modal supervision are tightly coupled with scene flow estimation, and their joint optimization significantly boosts overall performance. In Fig. 7, we compare the predicted scene flow with the ground truth in both synthetic and real-world scenarios.

3) Testing in Smoke-filled Environments: VISC+ is designed to enhance perception reliability in environments where visual sensing is degraded, such as adverse weather conditions. To validate its robustness, we evaluate VISC+ in smoke-filled settings through both simulation and real-world indoor testing. In the simulation, foggy scenes are generated in Carla, while in the physical setup, varying levels of smoke density are created using a smoke machine in the lab. In simulation, smoke densities of 30%, 50%, and 70% are defined using the fog attribute in the CARLA renderer, with corresponding reductions in visual range, contrast, and depth-dependent attenuation. In real indoor tests, smoke density levels are controlled by adjusting the output of a smoke machine, with levels measured using a smoke density sensor. Light, medium, and heavy smoke correspond to specific particle concentrations, with temporal variability controlled by maintaining a stable smoke output during testing. As shown in Table III, VISC+ maintains strong performance across different smoke conditions. Notably, it slightly surpasses CMFlow in scene flow accuracy, despite CMFlow relying on expensive LiDAR data during training.

For instance, under 30% synthetic smoke density, VISC+ reduces EPE by 30.9% and improves AccS by 46.2%, and AccR by 28.3%. Similar improvements are observed at 50% and 70% smoke levels, where VISC+ maintains lower EPE and better robustness in degraded visual conditions. In real-world indoor environments, VISC+ also outperforms both CMFlow and the original VISC. Under light smoke, VISC+ achieves a 27.1% reduction compared to CMFlow and a 5.6% reduction compared to VISC. Meanwhile, AccS improves by 22.7% over VISC and 57.4% over CMFlow. Even under heavy smoke, VISC+ maintains stable performance, with AccS and AccR significantly higher than CMFlow, indicating stronger generalization in severely degraded conditions.

Fig. 8 illustrates the qualitative comparison between VISC+ and CMFlow under different levels of smoke densities. We can see that under heavy smoke conditions, VISC+ can better estimate the scene flow. These results demonstrate that our mmWave-based scene flow estimation model, trained in benign environments, can be applied to smoke-filled environments. Based on these experiments, we believe that all commercial vehicles, not just a few specialized LiDAR-equipped vehicles, can contribute to data collection for training radar scene flow models. A more efficient data collection process results in more diverse data, thus alleviating the long-tail problem in autonomous vehicle perception.

4) Performance on Subtasks: In addition to scene flow estimation, we evaluated the motion segmentation task. Regarding the hyper-parameter k (c.f. Sec. III-C), we conduct a comparison for different values in Table. IV. Through experiments, we found that when k is set to 4, it effectively utilizes the distance information between feature points and static point clouds, resulting in relatively higher accuracy in scene flow estimation. In Table. VI, both ARV and feature selection techniques demonstrated the potential to enhance the performance of motion segmentation. Fig. 9 illustrates the motion segmentation results in two scenarios. The results demonstrate that our motion segmentation can achieve multi-object segmentation in complex scenes. We further report class-wise performance for static and dynamic points. On the evaluation set, the IoU for static points is 45.1, and the IoU for

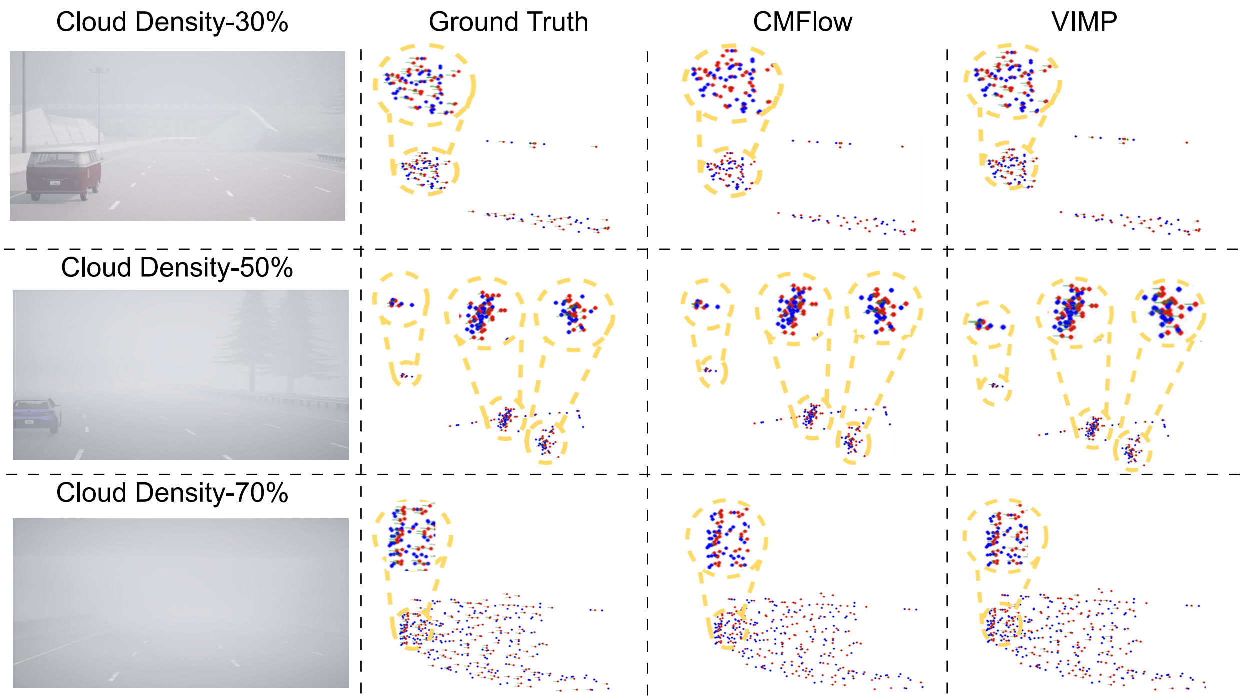

Fig. 8. Scene flow estimation under different levels of smoke densities. From left to right: 1) visual images; 2) two consecutive frames of point clouds, the first frame (red), the second frame (blue), and the ground truth scene flow (depicted in green); 3) the CMFlow results; 4) the VISC results. We highlight moving objects in brown circles and zoom in on them.

TABLE III
COMPARISON IN SMOKE-FILLED INDOOR ENVIRONMENTS

Smoke Density	EPE [m]↓	AccS↑	AccR↑
CMFlow-30% (synth.)	0.259	0.264	0.501
CMFlow-50% (synth.)	0.273	0.260	0.495
CMFlow-70% (synth.)	0.322	0.211	0.439
VISC-30% (synth.)	0.194	0.323	0.587
VISC-50% (synth.)	0.225	0.281	0.516
VISC-70% (synth.)	0.275	0.259	0.490
VISC+-30% (synth.)	**0.179**	**0.386**	**0.643**
VISC+-50% (synth.)	0.195	0.347	0.533
VISC+-70% (synth.)	0.249	0.300	0.518
CMFlow-Light (real-w.)	0.280	0.230	0.485
CMFlow-Med. (real-w.)	0.298	0.218	0.451
CMFlow-Heavy (real-w.)	0.355	0.135	0.227
VISC-Light (real-w.)	0.216	0.295	0.568
VISC-Med. (real-w.)	0.255	0.275	0.510
VISC-Heavy (real-w.)	0.305	0.215	0.448
VISC+-Light (real-w.)	**0.204**	**0.362**	**0.601**
VISC+-Med. (real-w.)	0.221	0.328	0.525
VISC+-Heavy (real-w.)	0.265	0.278	0.504

dynamic points is 43.5. The proportion of dynamic points in the evaluation set is approximately 60%. The proposed approach effectively utilizes visual feature points to constrain static radar point clouds.

TABLE IV
IMPACT OF HYPER-PARAMETERS

k	EPE [m]↓	AccS↑	AccR↑	mIoU↑
2	0.140	0.405	0.726	42.0
3	0.141	0.401	0.714	43.6
4	**0.139**	**0.408**	**0.749**	**45.2**
5	0.170	0.373	0.646	44.4

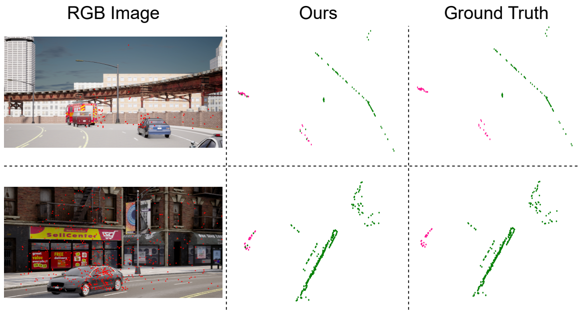

Fig. 9. Motion segmentation results. The left column shows the RGB image with the projected radar point cloud. In the middle and right columns, static points are green and dynamic points are pink.

TABLE V
SCENE FLOW ACCURACY BY ODOMETRY-ERROR BIN.

RTE bin	EPE [m] ↓	AccS ↑	AccR ↑
0–1% (synth.)	0.139	0.474	0.765
1–2% (synth.)	0.141	0.470	0.758
2–4% (synth.)	0.148	0.464	0.754
≥4% (synth.)	0.156	0.459	0.750
0–1% (real-w.)	0.173	0.392	0.706
1–2% (real-w.)	0.176	0.385	0.698
2–4% (real-w.)	0.185	0.382	0.693
≥4% (real-w.)	0.192	0.376	0.685

We obtain odometry between consecutive radar frames under the constraints of VI-SLAM and the recursive sensor fusion module. In Table VII, we evaluate the odometry estimation task, showing that VISC+ achieves further improvements over the original VISC due to the enhanced cross-task consistency introduced by the feature-selection module. Using the accumulated odometry, we visualize the trajectories in Fig. 10 for two representative scenarios. Although CMFlow achieves lower RTE and RAE in short-range motion, its reliance on high-precision LiDAR and RTK supervision limits its performance in longer trajectories where accumulated drift becomes significant. In contrast, VISC+ demonstrates more stable and accurate long-term trajectory estimation, benefiting from recursive fusion and task-level supervision refinement. Despite using only low-cost sensors, VISC+ yields lower accumulated drift than CMFlow, highlighting the robustness and practicality of our proposed framework in large-scale real-world navigation tasks. We also evaluate scene-flow robustness under different odometry qualities. Test trajectories are partitioned into non-overlapping 10 m segments. For each segment we compute relative translation and rotation errors of odometry. Segments are grouped by RTE into four bins: 0–1%, 1–2%, 2–4%, and ≥4%. For each bin we report EPE, AccS, and AccR. For simulation we use ground-truth poses. For real-world sequences we use VINS only to define the bins. The results are summarized in Table V. The degradation is smooth across bins, indicating the flow head remains robust even when the motion prior is less accurate.

TABLE VI
MOTION SEGMENTATION

ARV	Feature Select.	mIoU↑
		15.7
✓		23.8
✓	✓	**45.2**

TABLE VII
ODOMETRY ESTIMATION

VIO	Recur. Fus.	RTE [m]↓	RAE [°]↓
		0.185	0.203
✓		0.116	0.180
✓	✓	**0.076**	**0.147**

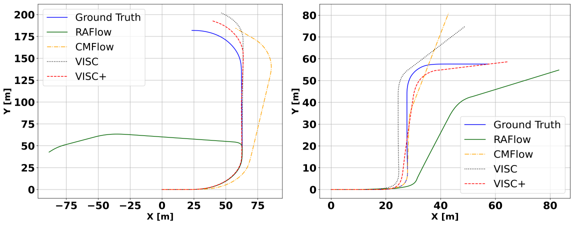

Fig. 10. The trajectories are plotted based on accumulated odometry for two scenarios.

V. Conclusion

This work introduces VISC+, a self-supervised learning framework enhanced by visual-inertial sensing, aimed at improving mmWave radar perception and enabling scalable crowd-sourced training. The key innovation lies in integrating deterministic motion models with learned statistical estimations to recursively leverage IMU data for accurate odometry, thereby providing reliable supervision for static background points. Then, we develop an optical-mmWave supervision extraction module that generates supervisory signals for both rigid-body transformation and scene flow estimation. Furthermore, our proposed feature-selection cross-modal learning module produces a more accurate motion segmentation leveraging background points. It further establishes consistency constraints on the scene flow and odometry estimation and jointly refine their results. Using datasets collected from both the Carla simulator and our custom-built sensor platform, we conduct extensive experiments demonstrating that VISC+ can surpass state-of-the-art methods that rely on expensive LiDAR systems. Looking ahead, we plan to further investigate the sensing capabilities of mmWave radar, with a particular focus on enhancing elevation angle resolution to improve the robustness and precision of radar-based perception.

Acknowledgement

This work was supported in part by EPSRC with RC Grant No EP/Y027787/1, UKRI under grant No EP/Y028317/1, Horizon European program under grant No 101086228, and the Royal Society International Exchanges 2023 under Grant No. IEC/NSFC/233318.

References

[1] K. Liu, Y. Zhou, M. Chen, J. He, Z. Yang, C. X. Lu, and S. Zhang, "Visc: mmwave radar scene flow estimation using pervasive visual-inertial supervision," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

[2] G. Dong, Y. Zhang, H. Li, X. Sun, and Z. Xiong, "Exploiting rigidity constraints for lidar scene flow estimation," in *IEEE Proc. CVPR*, 2022, pp. 12 776–12 785.

[3] H. Mittal, B. Okorn, and D. Held, "Just go with the flow: Self-supervised scene flow estimation," in *IEEE Proc. CVPR*, 2020, pp. 11 177–11 185.

[4] S. A. Baur, D. J. Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "Slim: Self-supervised lidar scene flow and motion segmentation," in *IEEE Proc. ICCV*, 2021, pp. 13 126–13 136.

[5] L. Chen, X. He, X. Zhao, H. Li, Y. Huang, B. Zhou, W. Chen, Y. Li, C. Wen, and C. Wang, "Gocomfort: Comfortable navigation for autonomous vehicles leveraging high-precision road damage crowdsensing," *IEEE Transactions on Mobile Computing*, 2022.

[6] M. Skog, O. Kotlyar, V. Kubelka, and M. Magnusson, "Human detection from 4d radar data in low-visibility field conditions," *arXiv preprint arXiv:2404.05307*, 2024.

[7] P. K. Rai, E. Kowsari, N. Strokina, and R. Ghabcheloo, "Uncertainty-driven radar-inertial fusion for instantaneous 3d ego-velocity estimation," *arXiv preprint arXiv:2506.14294*, 2025.

[8] J. Hur and S. Roth, "Self-supervised multi-frame monocular scene flow," in *IEEE Proc. CVPR*, June 2021, pp. 2684–2694.

[9] G. Yang and D. Ramanan, "Upgrading optical flow to 3d scene flow through optical expansion," in *IEEE Proc. CVPR*, 2020, pp. 1334–1343.

[10] L. Zhou, S. Leng, Q. Wang, and Q. Liu, "Integrated sensing and communication in uav swarms for cooperative multiple targets tracking," *IEEE Transactions on Mobile Computing*, 2022.

[11] J. Hur and S. Roth, "Self-supervised monocular scene flow estimation," in *IEEE Proc. CVPR*, 2020, pp. 7396–7405.

[12] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz, "Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation," in *ECCV*, 2018, pp. 468–484.

[13] C. Eising, J. Horgan, and S. Yogamani, "Near-field perception for low-speed vehicle automation using surround-view fisheye cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 13 976–13 993, 2021.

[14] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, and J. Kautz, "Sense: A shared encoder network for scene-flow estimation," in *IEEE/CVF Proc. ICCV*, 2019, pp. 3195–3204.

[15] F. Brickwedde, S. Abraham, and R. Mester, "Mono-sf: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes," in *IEEE/CVF Proc. ICCV*, 2019, pp. 2780–2790.

[16] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Proc. CVPR*, 2016, pp. 4040–4048.

[17] S. Lee, S. Im, S. Lin, and I. S. Kweon, "Learning residual flow as dynamic motion from stereo videos," in *IEEE/RSJ Proc. IROS*, 2019, pp. 1180–1186.

[18] H. Wang, J. Pang, M. A. Lodhi, Y. Tian, and D. Tian, "Festa: Flow estimation via spatial-temporal attention for scene point clouds," in *IEEE Proc. CVPR*, 2021, pp. 14 173–14 182.

[19] Y. Wei, Z. Wang, Y. Rao, J. Lu, and J. Zhou, "Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds," in *IEEE Proc. CVPR*, 2021, pp. 6954–6963.

[20] J. Hou, P. Yang, X. Dai, T. Qin, and F. Lyu, "Enhancing cooperative lidar-based perception accuracy in vehicular edge networks," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[21] R. Li, C. Zhang, G. Lin, Z. Wang, and C. Shen, "Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior," in *IEEE Proc. CVPR*, 2022, pp. 16 959–16 968.

[22] R. Xu, Z. Xiang, C. Zhang, H. Zhong, X. Zhao, R. Dang, P. Xu, T. Pu, and E. Liu, "Sckd: Semi-supervised cross-modality knowledge distillation for 4d radar object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 8933–8941.

[23] F. Rennie, D. Williams, P. Newman, and D. De Martini, "Doppler-aware odometry from fmcw scanning radar," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 5126–5132.

[24] C. Doer and G. F. Trommer, "x-rio: Radar inertial odometry with multiple radar sensors and yaw aiding," *Gyroscopy and Navigation*, vol. 12, no. 4, pp. 329–339, 2021.

[25] P. K. Rai, N. Strokina, and R. Ghabcheloo, "Representation learning for place recognition using mimo radar," *IEEE Open Journal of Intelligent Transportation Systems*, 2025.

[26] X. Zhang, D. Zhang, Y. Xie, D. Wu, Y. Li, and D. Zhang, "Waffle: A waterproof mmwave-based human sensing system inside bathrooms with running water," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, vol. 7, no. 4, 2024.

[27] H. Liu, X. Liu, X. Xie, X. Tong, and K. Li, "Pmtrack: Enabling personalized mmwave-based human tracking," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, vol. 7, no. 4, 2024.

[28] D. Cao, R. Liu, H. Li, S. Wang, W. Jiang, and C. X. Lu, "Cross vision-rf gait re-identification with low-cost rgb-d cameras and mmwave radars," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, vol. 6, no. 3, 2022.

[29] H. Arroyo, P. Keir, D. Angus, S. Matalonga, S. Georgiev, M. Goli, G. Dooly, and J. Riordan, "Segmentation of drone collision hazards in airborne radar point clouds using pointnet," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[30] T. Thilakanayake, O. De Silva, T. R. Wanasinghe, G. K. Mann, and A. Jayasiri, "All weather radar image enhancement and semantic segmentation method for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[31] Z. Zhang, H. Lai, D. Huang, X. Fang, M. Zhou, and Y. Zhang, "Reta: 4d radar-based end-to-end joint tracking and activity estimation for low-observable pedestrian safety in cluttered traffic scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4413–4426, 2023.

[32] F. Ding, A. Palffy, D. M. Gavrila, and C. X. Lu, "Hidden gems: 4d radar scene flow learning using cross-modal supervision," in *IEEE Proc. CVPR*, 2023, pp. 9340–9349.

[33] F. Ding, Z. Pan, Y. Deng, J. Deng, and C. X. Lu, "Self-supervised scene flow estimation with 4-d automotive radar," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8233–8240, 2022.

[34] "Velodyne lidar," https://velodynelidar.com/.

[35] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 475–480, 2005.

[36] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.

[37] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[38] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[39] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *IEEE Proc. ICRA*, 2020, pp. 3146–3152.

[40] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "Tlio: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[41] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.

[42] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[43] Y. Zhu, J. Zhang, W. Chen, C. Zhu, S. Yan, and Q. Chen, "Rest-imu: A two-stage resnet-transformer framework for inertial measurement unit localization," *Sensors*, vol. 25, no. 11, p. 3441, 2025.

[44] B. Rao, E. Kazemi, Y. Ding, D. M. Shila, F. M. Tucker, and L. Wang, "Ctin: Robust contextual transformer network for inertial navigation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 5, 2022, pp. 5413–5421.

[45] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[46] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[47] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *Proc. ICML*. PMLR, 2015, pp. 843–852.

[48] S. Zhang, S. Tang, W. Wang, T. Jiang, and Q. Zhang, "Conquering textureless with rf-referenced monocular vision for mav state estimation," in *IEEE Proc. ICRA*, 2021, pp. 146–152.

[49] S. Zhang, W. Wang, N. Zhang, and T. Jiang, "Lora backscatter assisted state estimator for micro aerial vehicles with online initialization," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4038–4050, 2022.

[50] ——, "Rf backscatter-based state estimation for micro aerial vehicles," in *IEEE Proc. INFOCOM*, 2020, pp. 209–217.

[51] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *IEEE Proc. IROS*, 2020, pp. 10 359–10 366.

[52] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[53] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Proc. CVPR*, 2015, pp. 3431–3440.

[54] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[55] A. Palffy, E. Pool, S. Baratam, J. Kooij, and D. Gavrila, "Multi-class road user detection with 3+1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, pp. 1–1, 2022.

[56] X. Peng, M. Tang, H. Sun, L. Servadei, and R. Wille, "4d mmwave radar in adverse environments for autonomous driving: A survey," *arXiv e-prints*, pp. arXiv–2503, 2025.

[57] J. Deng, W. Ye, H. Wu, X. Huang, Q. Xia, X. Li, J. Fang, W. Li, C. Wen, and C. Wang, "Cmd: A cross mechanism domain adaptation dataset for 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[58] L. Yang, X. Zhang, J. Li, C. Wang, J. Ma, Z. Song, T. Zhao, Z. Song, L. Wang, M. Zhou, Y. Shen, and C. Lv, "V2x-radar: A multi-modal dataset with 4d radar for cooperative perception," *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[59] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.