RIFlow: Radar-Inertial scene flow and ego-motion joint estimation

Pengshuo Li, Alen Alempijevic, Teresa Vidal-Calleja

***Abstract*—Due to the sparsity and noise in radar point clouds, existing 4D radar ego-motion estimation methods typically remove dynamic objects and rely on static returns to reconstruct the background environment and estimate odometry. However, dynamic object motion is the unavoidable information needed to be estimated in highly dynamic scenarios, supporting navigation and autonomous driving. To address this, we present RIFlow, a tightly integrated radar-inertial framework that jointly estimates ego-motion and scene flow using 4D radar point cloud and inertial measurements. The proposed sliding-window factor graph incorporates the Doppler velocity into dynamic factors, enabling dynamic-object motion to be explicitly modelled rather than discarded as outliers. Specifically, we designed three cluster-level self-supervised losses that exploit motion smoothness and Doppler radial velocity. A key aspect of our proposed approach is that it explicitly couples scene flow and ego-motion, forming a feedback mechanism in which improved scene flow benefits ego-motion estimation, while more accurate ego-motion further enhances motion segmentation and scene flow estimation. For ego-motion, radar Doppler information is integrated with IMU preintegration to construct geometric and Doppler residuals. To support reliable dynamic-object constraints in sparse real-world data, we further introduce an accumulated point-cloud tracking front-end that generates temporally stable object proposals and reduces frame-wise clustering fragmentation. Experiments on the simulation and real-world HeRCULES dataset demonstrate that RIFlow jointly estimates scene flow and ego-motion without LiDAR supervision and improves ego-motion accuracy when reliable dynamic-object constraints are available.**

I. Introduction

Scene flow estimation aims to recover the 3D motion of the surrounding environment, providing essential motion cues for robot perception and understanding of 3D scenes. Its downstream applications include motion segmentation, path planning, and obstacle avoidance in autonomous driving systems [1]. The 4D millimetre-wave radar, which measures range, azimuth, elevation and Doppler radial velocity, has attracted increasing attention from the autonomous driving community. However, compared with LiDAR, 4D radar point clouds are typically sparser and noisier, making 4D Radar scene flow and ego-motion estimation more challenging and still underexplored.

Existing 4D radar scene-flow methods can be broadly divided into cross-modal supervised methods and radar-only self-supervised methods. Cross-modal approaches exploit LiDAR, cameras, GNSS, or visual-inertial information to generate point-wise supervision for radar scene-flow estimation. HiddenFlow [2] and RaliFlow [3] utilise LiDAR and camera modalities to obtain accurate supervision and enhance cross-modal feature interaction, while weakly supervised and instance-aware methods further use image-based semantic information to reduce matching ambiguities [4]. Other recent methods, such as VISC [5], FGRFlow [6], and DoGFlow [7], also leverage multiple modalities, or Gaussian-based pseudo-point information to compensate for the sparsity of radar point clouds and improve motion estimation. Although cross-modal supervisory learning achieves strong performance, it usually requires additional high-level sensors, extensive data processing, and scenario-specific supervision. In contrast, radar-only self-supervised methods exploit the inherent kinematic information provided by radar Doppler measurements. RaFlow [8] demonstrates that Doppler velocity can be used to constrain radar scene flow without dense external supervision, but its performance remains inferior to fully supervised cross-modal methods.

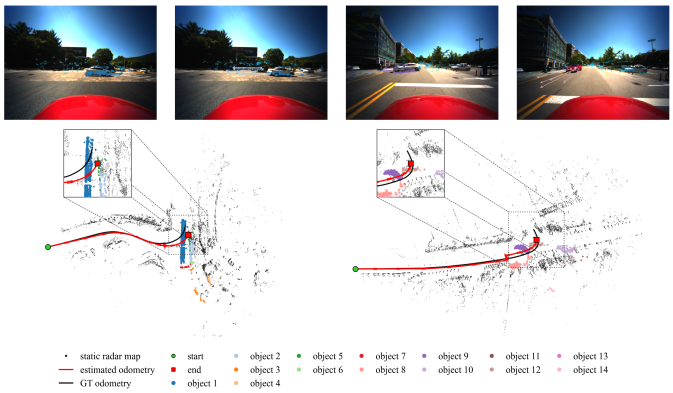


Fig. 1: **System output of RIFlow on a HeRCULES sequence.** RIFlow jointly estimates the ego-trajectory, static-scene map, and object-level scene-flow tracks from sparse 4D radar returns. Each moving object is represented by a object-motion vector.

More recent self-supervised approaches further incorporate diffusion models or traffic-level motion representations to improve radar scene-flow estimation [9, 10]. However, these methods depend on Kabsch-based ego-motion estimation and reliable persistent identity correspondence, which are difficult to obtain from sparse, noisy, and randomly distributed 4D radar point clouds. Moreover, existing pipelines estimate ego-motion and scene flow sequentially, leading to errors in static-dynamic segmentation and ego-motion compensation that propagate to the subsequent scene-flow estimation stage.

Radar-based ego-motion estimation has been improved from several complementary perspectives, including uncertainty modelling, Doppler velocity constraints, cluster-level geometric abstraction, loop closure, multi-session mapping, and

long-term localisation [11, 12]. The Radar-inertial odometry uses IMU preintegration and radar Doppler measurements to stabilise short-term motion estimation [13]. Spatially weighted Doppler methods exploit the reliability variation of radial velocity measurements for ego-motion estimations [14]. Cluster-based radar odometry abstracts sparse radar detections into more stable geometric structures [15]. Radar-inertial SLAM introduces scan registration and loop closure to suppress long-term drift [16]. Multi-session radar mapping and localisation further build reusable static radar maps for repeated localisation [17]. Nevertheless, these methods are still primarily static-scene driven, relying on static radar returns. This limits the use of radar measurements, even though Doppler measurements can provide information about dynamic objects. Our work differs by explicitly modelling dynamic objects as cluster-level flow variables and coupling their Doppler constraints with ego-motion estimation in a unified radar-inertial factor graph.

Dynamic SLAM extends conventional static-scene SLAM by incorporating dynamic entities into the estimation formulation, in which the robot trajectory, static map, and dynamic object poses, or motions, can be estimated jointly rather than treating moving observations purely as outliers [18]. Object-pose-based dynamic SLAM methods directly estimate the pose of each moving object, typically relying on feature-rich visual cameras. Early formulations such as SLAMMOT [19] and ClusterSLAM [20] incorporated moving objects as the state and used temporal rigidity and clustered dynamic observations to jointly optimise camera poses and object states. Subsequent visual dynamic SLAM systems, including ClusterVO [21], DynaSLAM II [22], and DOT [23], further combine moving-instance segmentation, object tracking, and SLAM to estimate ego-motion together with object poses in dynamic scenes. The geometric and semantic object priors were also exploited to stabilise object-level pose estimation and improve camera localisation. For example, objects are represented as 3D cuboids in CubeSLAM [24], dual quadrics in QuadricSLAM [25], and category-specific deep shape priors in DSP-SLAM [26]. Although these methods demonstrate the benefit of explicitly modelling object-level structures, their formulations rely on rich visual appearance information, reliable semantic segmentation, and stable object reference frames. These assumptions are difficult to satisfy for sparse and noisy 4D radar point clouds, where object extent is incomplete and persistent identity point correspondences are intermittent. Therefore, rather than estimating full object poses, we adopt an object-motion formulation that directly recovers the motion of dynamic radar clusters without instantiating strict object-specific pose frames.

Another class of dynamic SLAM methods estimates object motion directly, rather than representing each dynamic object by a time-varying pose. MVO [27] extends conventional visual odometry to multi-body motion estimation without requiring prior object models or planar motion assumptions. VDO-SLAM [28] further integrates semantic dynamic-object segmentation into a unified visual SLAM framework, estimating a spatio-temporal map without assuming known object shapes or motion models. Zhang et al. [29] improve dynamic multi-body visual odometry by jointly optimising object-level $SE(3)$ motion and optical flow. AirDOS [30] extends dynamic object-aware SLAM from rigid objects to articulated objects by introducing rigidity and motion constraints for individual rigid parts. These object-motion formulations can be further divided according to the coordinate frame used to represent dynamic motion. Object-centric formulations express motion relative to an object-specific frame, which is intuitive for rigid-object modelling but requires a reliable object reference frame. World-centric formulations instead express object motion in a common global frame, avoiding the need to instantiate and maintain object-specific frames for intermittently observed objects. Morris et al. [31] systematically compared object-centric and world-centric dynamic SLAM formulations and reported that the world-centric formulation better encodes rigid-body kinematics, improves object-motion estimation accuracy, and leads to more stable optimisation behaviour. DynoSAM [18] further provides an open-source smoothing-and-mapping framework that evaluates graph-based dynamic SLAM formulations by jointly estimating camera poses, static structure, dynamic object motions or poses, and dynamic object structure. Inspired by this world-centric principle, our formulation transfers the idea from visual dynamic SLAM to radar scene-flow estimation. Instead of maintaining a full object pose or an object-specific reference frame, each dynamic radar cluster motion is represented by a compact world-frame translation and jointly optimised with the radar-inertial ego-motion states. This representation is particularly suitable for sparse 4D radar, where dense object geometry, persistent point correspondences, and stable object-centric reference frames are difficult to obtain.

Given that the joint estimation of scene flow and ego-motion remains almost underexplored for sparse 4D radar, we propose a tightly integrated radar-inertial framework (RIFlow) that estimates both quantities using radar point clouds and inertial measurements. Unlike static-only odometry pipelines that discard moving objects, our formulation explicitly models dynamic-object motion in a world-centric pattern and incorporates radar Doppler measurements as dynamic factors in a unified sliding-window factor graph. The main contributions are summarised as follows:

- We formulate ego-motion and cluster-level scene flow as coupled variables in a radar-inertial factor graph, allowing dynamic object motion to provide useful Doppler constraints rather than being treated purely as outliers.
- We design three cluster-level self-supervised scene-flow losses based on motion smoothness and Doppler radial velocity, including a coupled loss that links scene flow and ego-motion estimation.
- We introduce an accumulated dynamic-object tracking front-end for sparse and noisy radar point clouds, which stabilises cluster-level motion estimation by reducing frame-wise over-egmentation and under-segmentation.

Experiments in simulation and on the real-world HeRCULES [32] dataset show that the proposed framework estimates radar

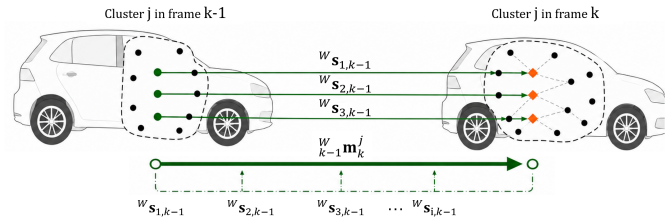


Fig. 2: **World-centric cluster-level scene-flow representation.** Sparse radar returns on a dynamic object are grouped into cluster j at frames $k-1$ and k. The green arrows illustrate point-wise scene-flow vectors ${}^{W}\mathbf{s}_{i,k-1}$ from selected radar returns in frame $k-1$ to their KNN-based correspondences in frame k. Instead of estimating independent point-level motions, RIFlow represents the dynamic object by a compact world-frame cluster-level displacement ${}^{W}_{k-1}\mathbf{m}^{j}_{k}$, which summarises the point-wise scene flows within the cluster.

scene flow and ego-motion without external supervision from LiDAR, cameras, or GNSS.

II. Methods

A. Notation

Let $\mathcal{T} = \{1, \ldots, K\}$ denote the ordered set of radar frames. We work with three rigidly related frames: a fixed world (inertial) frame W, the IMU body frame b, and the radar sensor frame r. The rigid extrinsic from the radar to the body frame, ${}^{b}\mathbf{T}_{r} = [{}^{b}\mathbf{R}_{r} \mid {}^{b}\mathbf{p}_{r}] \in SE(3)$, is given by the known extrinsic calibration. ${}^{A}\mathbf{R}_{B}$ defines the rotation from the B frame to the A frame, and ${}^{A}\mathbf{p}_{B}$ defines the position of the B origin expressed in the A frame. In this case, ${}^{b}\mathbf{R}_{r}$ defines the rotation from the radar frame to the body frame and ${}^{b}\mathbf{p}_{r}$ defines the position of the radar origin expressed in the body frame.

The body pose at frame k is written ${}^{W}\mathbf{T}_{b_k} = [{}^{W}\mathbf{R}_{b_k} \mid {}^{W}\mathbf{p}_{b_k}] \in SE(3)$, with ${}^{W}\mathbf{R}_{b_k} \in SO(3)$ and ${}^{W}\mathbf{p}_{b_k} \in \mathbb{R}^3$. A leading left superscript denotes the frame in which a quantity is expressed, and the paired subscript and superscript of ${}^{W}_{k-1}(\cdot)_k$ denote a quantity defined between frames $k-1$ and k. This *world-centric* convention removes the need to instantiate object-specific reference frame [18], a construct that sparse and intermittently detected 4D radar targets cannot reliably support.

1) Ego-motion state: The inertial state at frame k collects the body orientation, position, velocity, and the gyroscope and accelerometer biases, respectively,

$$\mathbf{x}_k = \{{}^{W}\mathbf{R}_{b_k}, {}^{W}\mathbf{p}_{b_k}, {}^{W}\mathbf{v}_{b_k}, \mathbf{b}^{g}_{k}, \mathbf{b}^{a}_{k}\}. \tag{1}$$

2) Radar measurements: A single 4D-radar scan at frame k yields a set of returns $\{({}^{r}\tilde{\mathbf{p}}_{i,k}, \tilde{v}_{i,k})\}_i$, where ${}^{r}\tilde{\mathbf{p}}_{i,k} \in \mathbb{R}^3$ is the measured position of return i in the radar frame and $\tilde{v}_{i,k} \in \mathbb{R}$ is its radial Doppler velocity. We denote the corresponding line-of-sight unit vector by

$$\mathbf{u}_{i,k} = \frac{{}^{r}\tilde{\mathbf{p}}_{i,k}}{\|{}^{r}\tilde{\mathbf{p}}_{i,k}\|_2}. \tag{2}$$

The Doppler velocity $\tilde{v}_{i,k} \in \mathbb{R}$ measures only the projection of the relative velocity onto $\mathbf{u}_{i,k}$.

B. World-Centric Dynamic Scene Flow Representation

Inspired by world-centric dynamic SLAM formulations [18, 31], we represent the scene flow of dynamic objects in a common observable reference frame rather than in an object-specific local frame. In our radar-inertial setting, this reference frame is the fixed world frame W, in which the ego-motion states, static map, and dynamic-object motions are jointly expressed. Different from visual Dynamic SLAM formulations that estimate object poses, object motions, and object structures, our radar formulation considers only the observable cluster-level translational motion, because sparse radar returns do not reliably support object-centric frames or dense object structures. The translational motion aligns well with the definition of scene flow, which utilises vectors to describe the displacement between two consecutive frames.

We index dynamic objects by clusters $j \in \mathcal{J}_k$. Rather than maintaining a persistent landmark for every radar return or estimating a full object pose $SE(3)$ for every dynamic object, we introduce a compact world-frame object-motion vector

$${}^{W}_{k-1}\mathbf{m}^{j}_{k} \in \mathbb{R}^3, \tag{3}$$

which denotes the translational motion of cluster j from frame $k-1$ to frame k, expressed in the world frame. Unlike an object-pose representation, this variable does not require an object-specific reference frame and object orientation. It captures only the observable translational component of the dynamic object's motion, supported by sparse radar returns.

For a radar return i assigned to cluster j, its world-frame position is denoted as ${}^{W}\bar{\mathbf{p}}_{i,k}$. Conceptually, if a material point on a dynamic object were observed in both consecutive frames, its world-frame scene flow would be

$${}^{W}\mathbf{s}_{i,k-1} = {}^{W}\bar{\mathbf{p}}_{i,k} - {}^{W}\bar{\mathbf{p}}_{i,k-1}. \tag{4}$$

Since persistent radar point correspondences are unreliable, RIFlow does not estimate this displacement at the point level. Instead, all returns assigned to cluster j read out the same cluster-level motion vector,

$${}^{W}\mathbf{s}^{j}_{i,k-1} = {}^{W}_{k-1}\mathbf{m}^{j}_{k}, \qquad i \in j. \tag{5}$$

Equivalently, the world-frame position of any return on cluster j is propagated by

$${}^{W}\bar{\mathbf{p}}_{i,k} \approx {}^{W}\bar{\mathbf{p}}_{i,k-1} + {}^{W}_{k-1}\mathbf{m}^{j}_{k}. \tag{6}$$

Therefore, the per-point scene flow is obtained directly from the parent cluster's world-frame object-motion vector ${}^{W}_{k-1}\mathbf{m}^{j}_{k}$. This pure-translation model should not be interpreted as assuming that real objects never rotate. Instead, it reflects the observability of 4D radar measurements. Because radar returns are sparse, noisy, and subject to detection dropout, the rotational component of an object's motion is weakly constrained and can be easily confused with changes in the visible radar scattering points. Introducing a full $SE(3)$ object pose would therefore add poorly observable degrees of freedom and can drive the optimisation toward degenerate solutions. The proposed world-centric object-motion vector provides a minimal

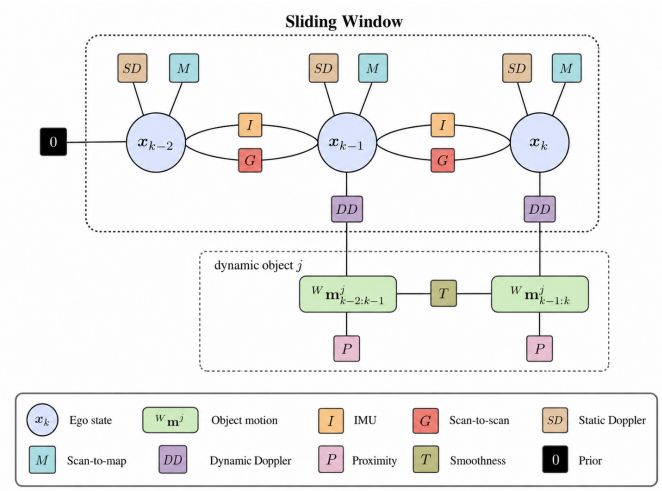


Fig. 3: **Sliding-window factor graph of RIFlow.** Ego-motion states $\mathbf{x}_k$ and cluster-level object-motion vectors ${}^{W}_{k-1}\mathbf{m}^{j}_{k}$ are jointly optimised. Static-scene factors constrain ego-motion, while dynamic-Doppler, proximity, and smoothness factors constrain object-motion variables and couple scene flow with ego-motion estimation.

but stable representation for radar scene-flow estimation: it preserves the dominant object motion observable from radar, avoids unreliable object-centric frames, and allows dynamic-object motion to be coupled with radar-inertial ego-motion estimation in the factor graph.

C. World-Centric Radar Dynamic Scene Flow Estimator

The formulation of the world-centric radar dynamic scene flow estimator optimises the compact joint state

$$\mathcal{X} = \Big\{ \mathbf{x}_k,\ {}^{W}_{k-1}\mathbf{m}^{j}_{k} \mid k \in \mathcal{T}_L,\ j \in \mathcal{J}_k \Big\}, \tag{7}$$

where $\mathcal{T}_L$ is the active sliding window of the most recent L frames. We estimate $\mathcal{X}$ by solving a single factor graph over the active window, illustrated in Fig. 3. The graph couples the inertial and radar constraints acting on the ego-motion states with the Doppler, geometric, and motion constraints acting on the object-motion vectors.

1) Inertial factor: Consecutive frames are linked by an on-manifold IMU preintegration factor [33, 34], yielding the 15-dimensional residual $\mathbf{r}_{\mathcal{I}} = [\mathbf{r}_{\mathbf{R}}^{\top}, \mathbf{r}_{\mathbf{v}}^{\top}, \mathbf{r}_{\mathbf{p}}^{\top}, \mathbf{r}_{ba}^{\top}, \mathbf{r}_{bg}^{\top}]^{\top}$ over the rotation, velocity, position, and bias-random-walk errors. The inertial factor provides a strong orientation constraint, which is essential for expressing the object-motion vector consistently in the world frame.

2) Static Doppler factor: For a return s classified as static, the ego state predicts its radial velocity, $\hat{v}_{s,k} = -\mathbf{u}_{s,k}^{\top}({}^{r}\mathbf{v}_{r_k} + {}^{r}\boldsymbol{\omega}_{r_k} \times {}^{r}\tilde{\mathbf{p}}_{s,k})$, where ${}^{r}\mathbf{v}_{r_k}$ is obtained from the body velocity and extrinsic calibration, and ${}^{r}\boldsymbol{\omega}_{r_k}$ is obtained from the bias-corrected gyroscope measurement. The residual is the scalar

$$\mathbf{r}^{SDf}_{s,k} = \tilde{v}_{s,k} - \hat{v}_{s,k} \in \mathbb{R}. \tag{8}$$

3) Dynamic Doppler factor: We approximate the cluster average velocity during the interval $[t_{k-1}, t_k]$ as ${}^{W}_{k-1}v^{j}_{k} \approx {}^{W}_{k-1}\mathbf{m}^{j}_{k}/\Delta t$. For a return n assigned to dynamic cluster j, it shares the same average velocity as the cluster. The predicted radial velocity combines the ego-motion with the object-motion vector,

$$\hat{v}^{\mathrm{dyn}}_{n,k} = -\mathbf{u}_{n,k}^{\top}\Big({}^{r}\mathbf{v}_{r_k} + {}^{r}\boldsymbol{\omega}_{r_k} \times {}^{r}\tilde{\mathbf{p}}_{n,k} - \big({}^{W}\mathbf{R}_{b_k}\,{}^{b}\mathbf{R}_{r}\big)^{\top}\tfrac{{}^{W}_{k-1}\mathbf{m}^{j}_{k}}{\Delta t}\Big), \tag{9}$$

giving the residual $\mathbf{r}^{DDf}_{n,k} = \tilde{v}_{n,k} - \hat{v}^{\mathrm{dyn}}_{n,k}$. Crucially, this factor requires only the assignment of the return n to cluster j; it does not rely on a persistent per-point correspondence across frames.

4) Spatial proximity factor: Since radar landmarks are absent, temporal geometry is imposed at the cluster level. Although the scene-flow variable is defined in the forward direction from $k-1$ to k, the proximity residual is evaluated by backward warping current returns to the previous scan, which avoids requiring persistent point identities. For an ego-compensated current point ${}^{W}\bar{\mathbf{p}}_{n,k}$, its predicted previous-frame location ${}^{W}\bar{\mathbf{p}}_{n,k} - {}^{W}_{k-1}\mathbf{m}^{j}_{k}$ is matched to its KNN-weighted correspondence ${}^{W}\bar{\mathbf{p}}^{*}_{n',k-1}$ in the previous scan [8], yielding

$$\mathbf{r}^{Spf}_{n,k} = \big({}^{W}\bar{\mathbf{p}}_{n,k} - {}^{W}_{k-1}\mathbf{m}^{j}_{k}\big) - {}^{W}\bar{\mathbf{p}}^{*}_{n',k-1} \in \mathbb{R}^3. \tag{10}$$

The spatial proximity factor does not require persistent radar point identities across frames. It only uses local nearest-neighbor association as a soft geometric constraint.

5) Translation smoothness factor: A constant-velocity prior on the cluster motion enforces temporal continuity,

$$\mathbf{r}^{Tsf}_{j,k} = {}^{W}_{k-1}\mathbf{m}^{j}_{k} - {}^{W}_{k-2}\mathbf{m}^{j}_{k-1} \in \mathbb{R}^3, \tag{11}$$

and regularises clusters that are small or only briefly observed.

6) Ego registration factors: The processing pipeline supplies two radar registrations (Sec. II-D): a relative pose $\hat{\mathbf{T}}^{G}_{k-1,k}$ from scan-to-scan matching and a a map-relative pose measurement $\hat{\mathbf{T}}^{M}_{wb_k}$ from scan-to-map registration. The radar-frame registration outputs are transformed into the body frame using the calibrated extrinsic. These enter as the $SE(3)$ residuals

$$\mathbf{r}^{G}_{k-1,k} = \mathrm{Log}\big((\hat{\mathbf{T}}^{G}_{k-1,k})^{-1}\,{}^{W}\mathbf{T}^{-1}_{b_{k-1}}\,{}^{W}\mathbf{T}_{b_k}\big) \in \mathbb{R}^6, \tag{12}$$

$$\mathbf{r}^{M}_{k} = \mathrm{Log}\big((\hat{\mathbf{T}}^{M}_{wb_k})^{-1}\,{}^{W}\mathbf{T}_{b_k}\big) \in \mathbb{R}^6, \tag{13}$$

both Huber-robustified. Factor $\mathbf{r}^{M}$ is the term through which the static-map representation suppresses long-horizon drift.

7) Joint objective: Collecting all active factors over the window, the estimate is obtained by minimising the whitened,

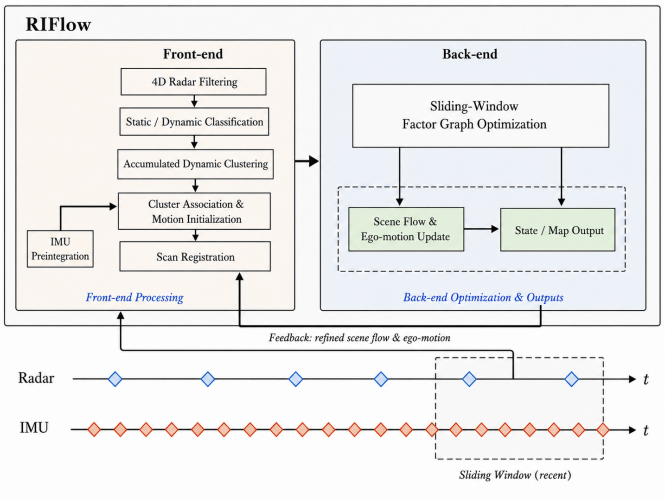


Fig. 4: **Overview pipeline of RIFlow.**

robustified non-linear least-squares cost

$$\begin{aligned}\mathcal{X}^* = \arg\min_{\mathcal{X}} & \sum_k \|\mathbf{r}_{\mathcal{I}}\|^2_{\Sigma_{\mathcal{I}}} + \sum_{s,k} \|\mathbf{r}^{SDf}_{s,k}\|^2_{\sigma^2_{(SDf)}} \\ & + \sum_{n,k} \rho\big(\|\mathbf{r}^{DDf}_{n,k}\|^2_{\sigma^2_{DDf}}\big) \\ & + \sum_{n,k} \rho\big(\|\mathbf{r}^{Spf}_{n,k}\|^2_{\Sigma_{Spf}}\big) + \sum_{j,k} \|\mathbf{r}^{Tsf}_{j,k}\|^2_{\Sigma_{Tsf}} \\ & + \sum_k \rho\big(\|\mathbf{r}^{G}_{k-1,k}\|^2_{\Sigma_G}\big) + \sum_k \rho\big(\|\mathbf{r}^{M}_{k}\|^2_{\Sigma_M}\big),\end{aligned} \tag{14}$$

where $\rho(\cdot)$ denotes the robust kernel and $\Sigma_{(\cdot)}$ the associated covariances. The problem is built over the most recent L frames and their active object-motion vectors, warm-started from the previous window, and solved with a sparse trust-region method. An estimate is produced at every radar frame without keyframe thinning, since both the Doppler and translation residuals depend on the instantaneous body velocity. The per-point scene flow is then read from Eq. (5).

D. Front-End Processing Pipeline

The front-end processing pipeline converts raw radar and inertial streams into the measurements and initial estimates required by the sliding-window factor graph 4. It provides static and dynamic radar measurements, temporally consistent dynamic-cluster labels, initial object-motion estimates, inertial preintegrants, and scan-registration constraints for the back end.

At each radar frame k, the pipeline produces

$$\begin{aligned}\mathcal{Z}_k = \Big\{ & \mathcal{S}_k,\, \mathcal{D}_k,\, \mathcal{J}_k,\, \big\{{}^{W}_{k-1}\hat{\mathbf{m}}^j_k\big\}_{j\in\mathcal{J}_k}, \\ & \Delta\hat{\mathcal{I}}_{k-1,k},\, \hat{\mathbf{T}}^G_{k-1,k},\, \hat{\mathbf{T}}^M_{wb_k}\Big\}.\end{aligned} \tag{15}$$

where $\mathcal{S}_k$ and $\mathcal{D}_k$ denote the static returns and dynamic candidates, respectively; $\mathcal{J}_k$ denotes the tracked dynamic clusters; ${}^{W}_{k-1}\hat{\mathbf{m}}^j_k$ is the initial world-frame object-motion estimate of cluster j; $\Delta\hat{\mathcal{I}}_{k-1,k}$ is the IMU preintegrated measurement [34]; and $\hat{\mathbf{T}}^G_{k-1,k}$ and $\hat{\mathbf{T}}^M_{wb_k}$ are the scan-to-scan and scan-to-map registration measurements. Unlike visual dynamic SLAM front ends that rely on dense optical flow, semantic masks, and persistent image features, our radar front end does not establish persistent per-point correspondence across frames. Instead, it uses Doppler consistency, short-term temporal accumulation, and cluster-level association to generate dynamic-object proposals for sparse 4D radar point clouds.

1) 4D-radar filtering: Raw 4D-radar point clouds contain ghost detections from multipath effects, which typically lie below the ground surface, and flickering random returns. Following [35], we apply two stages. *Ghost-point removal* fits a ground plane by RANSAC to candidate points that lie within a band of the mounting height and exhibit a near-vertical surface normal, and discards returns falling below that plane. *Random-point removal* transforms the previous scan into the current frame using the propagated ego-motion estimate and rejects any current return that has no neighbour within a fixed radius in the transformed cloud. Conservative range, elevation, and intensity guards additionally bound the field of view.

2) Static-dynamic classification: After filtering, the front-end partitions the radar returns into static returns and dynamic candidates using Doppler consistency. Using the ego-motion prediction from the previous sliding window, the pipeline predicts the sensor's radar-frame linear and angular velocities. For each radar return i at frame k, we compute the ego-compensated Doppler residual

$$\delta v_{i,k} = \tilde{v}_{i,k} + \mathbf{u}^\top_{i,k}\left({}^{r}\mathbf{v}_{r_k} + {}^{r}\boldsymbol{\omega}_{r_k} \times {}^{r}\tilde{\mathbf{p}}_{i,k}\right). \tag{16}$$

For an ideal static return, the measured Doppler velocity should be explained by the ego-induced radar motion, and therefore $\delta v_{i,k}$ should be close to zero. Returns satisfying $|\delta v_{i,k}| < \tau_v$ are labelled as static and inserted into $\mathcal{S}_k$. These returns are used by the static Doppler factor and the scan-registration module. The remaining returns are treated as dynamic candidates and inserted into $\mathcal{D}_k$ for accumulated clustering.

3) Accumulated clustering and tracking: A single 4D-radar scan is often too sparse to reliably segment dynamic objects, especially distant vehicles, pedestrians, or partially observed targets. Frame-wise clustering can therefore oscillate between over-segmentation and under-segmentation, producing unstable cluster-level motion estimates. To improve the density and temporal stability of dynamic-object proposals, we accumulate dynamic candidates over a short trailing window of recent frames. Each dynamic candidate from frame ℓ is transformed into the world frame using the ego estimate of that frame,

$${}^{W}\bar{\mathbf{p}}_{n,\ell} = {}^{W}\mathbf{R}_{b_\ell}\left({}^{b}\mathbf{R}_r\, {}^{r}\tilde{\mathbf{p}}_{n,\ell} + {}^{b}\mathbf{p}_r\right) + {}^{W}\mathbf{p}_{b_\ell}, \tag{17}$$

where ${}^{W}\bar{\mathbf{p}}_{n,\ell} = [{}^{W}\bar{x}_{n,\ell}, {}^{W}\bar{y}_{n,\ell}, {}^{W}\bar{z}_{n,\ell}]^\top$. The transformed dynamic candidates are then stacked in the world frame. Because the vertical coordinate of sparse radar detections is less reliable than the ground-plane position, clustering is performed in an augmented ground-plane Doppler feature space,

$$\boldsymbol{\phi}_{n,\ell} = \left[{}^{W}\bar{x}_{n,\ell},\, {}^{W}\bar{y}_{n,\ell},\, \delta v_{n,\ell}\right]^\top. \tag{18}$$

A density-based clustering algorithm is applied to this feature space, grouping returns only when they are mutually consistent in both position and residual radial velocity. The resulting accumulated clusters are denser and more stable than clusters extracted from a single radar scan.

4) Object-motion initialisation: For each tracked dynamic cluster j, the front end provides an initial estimate of its world-frame object-motion vector between frames $k-1$ and k. Let ${}^{W}\mathbf{c}_k^j$ and ${}^{W}\mathbf{c}_{k-1}^j$ denote the centroids of the matched cluster in the current and previous frames, respectively. The initial object-motion estimate is computed as

$$ {}^{W}_{k-1}\hat{\mathbf{m}}_k^j = {}^{W}\mathbf{c}_k^j - {}^{W}\mathbf{c}_{k-1}^j. \tag{19} $$

The accumulation window is very short, so motion smear is limited. A newly spawned cluster has no previous centroid and is therefore activated in the optimisation only after it has been observed in at least two consecutive frames. This centroid-based initialisation places the object-motion variable within the convergence basin of the nonlinear optimiser while remaining persistent identity correspondence-free. The back end subsequently refines ${}^{W}_{k-1}\mathbf{m}_k^j$ using the dynamic Doppler, spatial proximity, and temporal smoothness factors.

5) Scan registration: The static scene is represented by a rolling voxel map of recent static returns, implemented as an age-bounded voxel grid that is compact and incrementally maintained. At each frame, the current static point cloud is registered (i) to this map by a radar-adapted Generalised-ICP variant with anisotropic, range-scaled point covariances [36], producing the global pose $\hat{\mathbf{T}}^{M}_{wb_k}$, and (ii) to the previous static scan, producing the relative pose $\hat{\mathbf{T}}^{G}_{k-1,k}$. These supply the $SE(3)$ measurements of Eqs. (12)–(13).

III. Experiments

A. Implementation and Setup

The backend is implemented in C++ on top of the Ceres solver, using a trust-region (Dogleg) method. The sliding window spans the most recent $L = 10$ radar frames. The problem is rebuilt and solved at every radar frame, without keyframe thinning, and warm-started from the previous window. Inertial preintegration uses a continuous-integration preintegrator [34]. In simulation, the radar runs at 10 Hz and the IMU at 200 Hz, with a fixed extrinsic calibration. Each sequence contains multiple independently moving objects, and the simulator exposes the ground-truth ego trajectory and per-point flow. On HeRCULES [32], the radar is a Continental ARS548 (20 Hz) and the inertial sensor is an XSens MTi-300 (100 Hz).

B. Evaluation Metrics

1) Ego-motion: Let $\mathbf{p}_k$ and $\hat{\mathbf{p}}_k$ be the ground-truth and estimated positions and $\mathbf{R}_k, \hat{\mathbf{R}}_k$ the corresponding orientations. We report the absolute trajectory error (ATE) as the RMSE of $\|\hat{\mathbf{p}}_k - \mathbf{p}_k\|$ over a sequence, the final ATE (F-ATE) at the last frame, and the rotation RMSE: $\mathrm{RMSE}_k\|\mathrm{Log}(\mathbf{R}_k^\top \hat{\mathbf{R}}_k)\|$ in degrees. On the real dataset, we also report the per-frame relative pose error (RPE) for translation, isolating local drift from accumulated global error.

TABLE I: **Scene-flow Benchmark on the HeRCULES dataset.**

Method	Overall			Moving			Static		
	EPE↓	Acc_5↑	Acc_{10}↑	EPE↓	Acc_5↑	Acc_{10}↑	EPE↓	Acc_5↑	Acc_{10}↑
ICP[37]	0.497	0.004	0.029	0.660	0.003	0.020	0.490	0.004	0.030
SceneFF[38]	0.295	0.036	0.058	0.486	0.008	0.031	0.287	0.037	0.059
RaFlow[8]	0.181	0.027	0.184	0.443	0.020	0.131	0.170	0.027	0.186
Ours	**0.148**	**0.373**	**0.605**	**0.439**	**0.064**	**0.310**	**0.136**	**0.386**	**0.618**

TABLE II: **Ego-motion benchmark on the HeRCULES dataset.**

Method	Sports Complex			Parking Lot		
	RTE [m]	RRE [°]	ATE [m]	RTE [m]	RRE [°]	ATE [m]
Odometry						
Radar-ICP	0.049	**0.071**	7.10	0.058	**0.089**	3.40
Go-RIO [39]	0.233	1.190	29.28	0.459	0.221	5.74
Ours (no dyn.)	0.038	0.380	9.98	**0.023**	0.370	1.94
Ours	**0.029**	0.200	**5.44**	**0.023**	0.370	**1.90**
SLAM						
4DRadarSLAM [40]	1.680	2.760	183.84	2.900	8.630	163.26
RaI-SLAM [16]	0.020	0.023	3.90	0.022	0.033	2.20

2) Scene flow: For each dynamic point i with estimated flow $\hat{\mathbf{s}}_i$ and ground-truth flow $\mathbf{s}_i$, the end-point error (EPE) is $\|\hat{\mathbf{s}}_i - \mathbf{s}_i\|$. We report its RMSE over all dynamic points. Following the scene-flow convention, we also report two accuracy rates: Acc_{10} and Acc_5, the fraction of points whose EPE is below 0.10 m and 0.05 m, respectively.

C. Real-World HeRCULES Dataset

1) Scene-flow benchmark: Table I reports the scene-flow accuracy of RIFlow and the baseline methods on the HeRCULES dataset. Among all compared methods, RIFlow achieves the best overall performance, reducing the EPE to 0.148 while improving Acc5 and Acc10 to 0.373 and 0.605, respectively. On moving points, RIFlow attains the lowest EPE of 0.439 and the highest Acc5 and Acc10 of 0.064 and 0.310, showing clear advantages in estimating dynamic motions. On static points, RIFlow also achieves the best results, with an EPE of 0.136, Acc5 of 0.386, and Acc10 of 0.618. These results indicate that the proposed world-centric cluster-level formulation improves both dynamic and static scene-flow estimation compared with ICP, SceneFF, and RaFlow.

Fig. 5 provides a qualitative comparison through point-wise error maps. Compared with the baselines, RIFlow produces more low-error regions and fewer large-error outliers across different scenes, especially on moving objects, which is consistent with the quantitative improvements reported in Table I.

2) Qualitative scene flow and mapping: Figs. 1 and 6 further show that the per-object scene flow is recovered consistently across road-user classes (vehicle, pedestrian, motorcycle) and across turning, high-speed, low-speed, and static-ego conditions. The estimated object tracks remain coherent on the accumulated static radar map. Fig. 7 further shows that the reconstructed static maps are sharp and the estimated trajectory

TABLE III: **Ego-motion ablation on five HeRCULES sequences,** ablating the frontend dynamic-clustering mode. Best ATE per row in **bold**.

Sequence	Dur.	ATE (m)↓			RPE_t	RRE
	(s)	No-dyn.	Per-frame	Accum.	(Ours)↓	(Ours)↓
ParkingLot	125	1.94	1.93	**1.90**	0.023	0.369
SportsComplex	320	9.97	8.31	**5.43**	0.016	0.461
Library	150	9.72	9.61	**6.82**	0.025	0.296
RiverIsland	175	13.72	32.64	**7.69**	0.032	0.396
Mountain	150	**7.07**	7.42	7.32	0.030	0.422

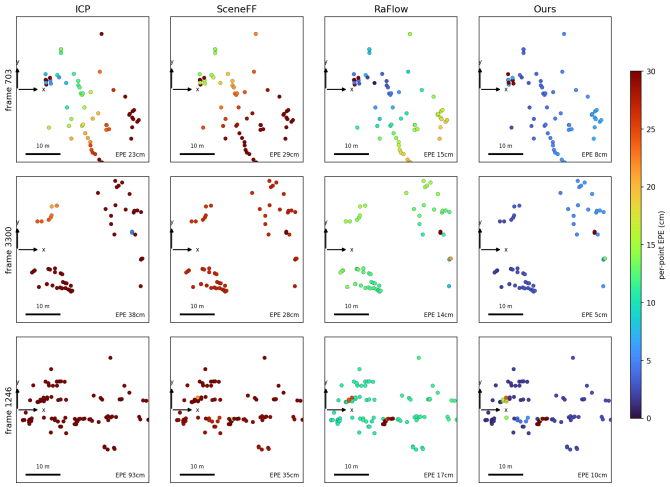


Fig. 5: **Scene flow error benchmark on HeRCULES.**

follows the ground truth on both the structured ParkingLot and the larger SportsComplex environment.

3) Ego-motion benchmark: Table II compares RIFlow with representative radar odometry and radar SLAM methods on two HeRCULES sequences. Among the odometry methods, RIFlow achieves the best ATE on both Sports Complex and Parking Lot, reducing the error from 7.10 m to 5.44 m and from 3.40 m to 1.90 m compared with Radar-ICP, respectively. Compared with Ours (no dyn.), the full model further reduces the ATE on both sequences, indicating that the dynamic-object residuals provide useful constraints for ego-motion estimation. Although RaI-SLAM achieves lower errors by using a SLAM formulation with loop closure, RIFlow remains competitive among odometry-only methods.

4) Ego-motion ablation: Table III evaluates RIFlow under different frontend dynamic-clustering modes on five HeRCULES sequences. Compared with No-dyn., the Accum. setting reduces ATE in four out of five sequences, with clear improvements on ParkingLot and SportsComplex from 1.94,m to 1.90,m and from 9.97,m to 5.43,m, respectively. Accum. consistently outperforms Per-frame clustering, showing that temporally accumulated dynamic clusters provide more stable Doppler constraints. The Mountain sequence is the only exception because of the large vertical drift in this sequence, which dominates the ATE and weakens the apparent benefit of the dynamic factors.

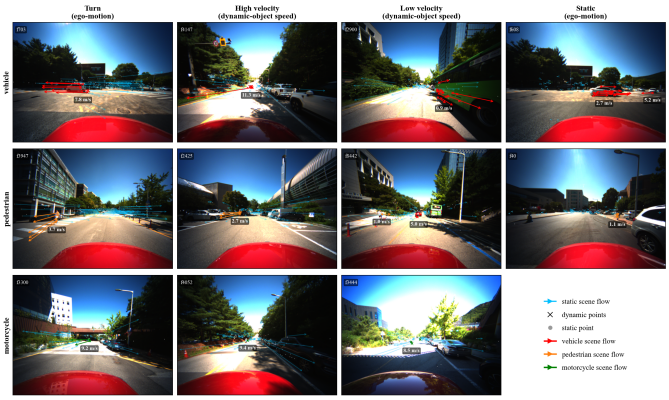
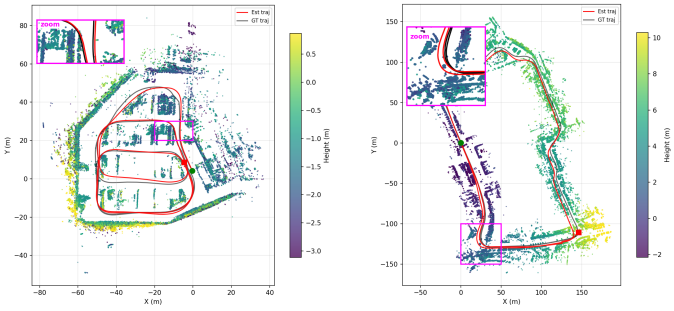


Fig. 6: **Qualitative scene flow on HeRCULES.** Rows correspond to road-user classes. Columns correspond to turning ego-motion, high-speed and low-speed targets, and static scenes.



(a) ParkingLot (b) SportsComplex

Fig. 7: **Reconstructed static radar maps and estimated trajectorieson two HeRCULES sequences.**

D. Ablation study in simulation

We evaluate the proposed factors on three simulated trajectories: Default, Wide Loop, and Rectangle. All configurations retain the IMU preintegration and static-Doppler residuals, while different dynamic scene-flow and scan-registration factors are removed to analyse their contributions.

1) Scene-flow accuracy: Table IV evaluates the contribution of the scene-flow factors. The setting w/o $\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}$ keeps only the dynamic Doppler factor $\mathbf{r}^{DDf}$ for object-motion estimation, and therefore represents the Doppler-only scene-flow baseline. This setting gives limited accuracy and fails severely in the Rectangle trajectory, indicating that Doppler alone is insufficient to recover stable cluster-level scene flow. The setting w/o $\mathbf{r}^{Tsf}$ retains both $\mathbf{r}^{DDf}$ and $\mathbf{r}^{Spf}$, but removes the temporal smoothness constraint; it improves over the Doppler-only setting but still underperforms the full formulation. With all scene-flow factors retained, i.e., $\mathbf{r}^{DDf}+\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}$, the full coupled formulation achieves the best or comparable EPE across all trajectories, with 0.091 m, 0.074 m, and 0.093 m, and obtains the highest Acc_5 values. These results show that $\mathbf{r}^{Spf}$ provides geometric consistency, while $\mathbf{r}^{Tsf}$ supplies the key temporal regularisation needed for stable cluster-level scene-flow estimation.

TABLE IV: **Scene-flow factor ablation on three simulation trajectories.**

Configuration	Default			Wide Loop			Rectangle		
	EPE↓	Acc_{10} ↑	Acc_5 ↑	EPE↓	Acc_{10} ↑	Acc_5 ↑	EPE↓	Acc_{10} ↑	Acc_5 ↑
w/o $\mathbf{r}^{DDf}$ (no flow)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
w/o $\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}$ (dec.)	0.113	67.1	30.5	0.276	81.5	50.1	2.454	60.5	25.2
w/o $\mathbf{r}^{Tsf}$ (dec.)	0.106	67.6	30.5	0.264	82.1	50.4	0.129	63.4	25.6
Full (dec.)	**0.091**	**75.2**	33.8	0.075	86.8	54.5	0.097	70.3	27.9
w/o $\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}$ (cou.)	0.113	67.1	30.5	0.277	81.8	51.0	3.339	60.7	25.8
w/o $\mathbf{r}^{Tsf}$ (cou.)	0.105	67.1	31.0	0.264	82.9	51.5	0.127	64.1	27.1
Full (cou.)	**0.091**	74.8	**34.3**	**0.074**	**87.1**	**55.1**	**0.093**	**71.4**	**30.0**

TABLE V: **Ego-odometry factor ablation on three simulation trajectories.** ATE and F-ATE are reported in meters, and Rot. is reported in degrees.

Configuration	Default			Wide Loop			Rectangle		
	ATE↓	F-ATE↓	Rot.↓	ATE↓	F-ATE↓	Rot.↓	ATE↓	F-ATE↓	Rot.↓
w/o $\mathbf{r}^{DDf}+\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$	0.6045	1.4135	1.054	0.3574	0.9413	1.004	1.4473	2.6887	2.195
w/o $\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$ (dec.)	0.6370	1.4899	1.106	0.3782	0.9993	1.050	1.4197	2.5544	2.105
w/o $\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$ (dec.)	0.6384	1.4946	1.110	0.3783	0.9994	1.050	1.4197	2.5544	2.105
w/o $\mathbf{r}^{G}+\mathbf{r}^{M}$ (dec.)	0.6267	1.4599	1.083	0.3783	0.9994	1.050	1.4197	2.5544	2.105
w/o $\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$ (cou.)	0.6236	1.4546	1.080	0.3782	0.9994	1.050	1.3810	2.5176	1.997
w/o $\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$ (cou.)	0.7011	1.6288	1.194	0.3678	0.9918	1.064	1.1996	2.2212	1.682
w/o $\mathbf{r}^{G}+\mathbf{r}^{M}$ (cou.)	0.6826	1.5964	1.183	0.3065	0.8614	0.964	**1.0376**	1.9642	**1.440**
w/o $\mathbf{r}^{M}$	0.6728	1.5753	1.168	**0.2170**	**0.6618**	**0.829**	1.0987	2.0337	1.571
Full	**0.5808**	**1.3599**	**1.019**	0.2287	0.6756	0.830	1.0554	**1.9286**	1.503

2) Ego-motion accuracy: Table V evaluates the contribution of each factor to ego-motion accuracy. The setting w/o $\mathbf{r}^{DDf}+\mathbf{r}^{Spf}+\mathbf{r}^{Tsf}+\mathbf{r}^{G}+\mathbf{r}^{M}$ uses only the IMU preintegration and static-Doppler residuals, i.e., $\mathbf{r}_{\mathcal{I}} + \mathbf{r}^{SDf}$, and therefore serves as the static-only baseline. The decoupled dynamic-Doppler settings, denoted as dec., introduce $\mathbf{r}^{DDf}$ but do not directly couple the object-motion variables with the ego-motion state, and they do not consistently improve ego-motion accuracy. In contrast, the coupled settings, denoted as cou., make $\mathbf{r}^{DDf}$ depend jointly on the ego-motion and object-motion variables. When $\mathbf{r}^{Spf}$ and $\mathbf{r}^{Tsf}$ are retained but both scan-registration factors $\mathbf{r}^{G}$ and $\mathbf{r}^{M}$ are removed, the coupled setting reduces ATE in the Wide Loop and Rectangle trajectories, showing that the dynamic-object constraints can directly contribute to ego-motion estimation. With all factors retained, the full model achieves the best Default ATE and the best Rectangle F-ATE, while remaining competitive on Wide Loop and Rectangle. These results confirm that the proposed coupled dynamic factors provide complementary constraints to the static-scene registration terms $\mathbf{r}^{G}$ and $\mathbf{r}^{M}$.

IV. Conclusion

We propose RIFlow, a radar-inertial framework that jointly estimates ego-motion and scene flow from 4D radar point clouds. The proposed factor graph explicitly models dynamic objects using object-motion vectors and incorporates radar Doppler measurements as motion constraints, allowing dynamic returns to support ego-motion estimation rather than being discarded as outliers. An accumulated dynamic-object tracking front-end further improves the stability of cluster-level motion estimation in sparse real-world radar data. Experiments in simulation and on the HeRCULES dataset show that RIFlow estimates scene flow and improves radar-inertial odometry without relying on LiDAR or GNSS supervision.

References

[1] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 03, pp. 475–480, 2005.

[2] F. Ding, A. Palffy, D. M. Gavrila, and C. X. Lu, "Hidden gems: 4d radar scene flow learning using cross-modal supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9340–9349.

[3] J. Fu, Z. Xiang, and N. Zhao, "Raliflow: Scene flow estimation with 4d radar and lidar point clouds," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 5, 2026, pp. 4012–4021.

[4] J. Fu, Z. Xiang, and N. Zhao, "Weakly supervised cross-modal learning for 4d radar scene flow estimation," 2026. [Online]. Available: https://arxiv.org/abs/2605.18507

[5] K. Liu, Y. Zhou, M. Chen, J. He, J. Xu, 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)*, 2025, pp. 6765–6770.

[6] M. Zhai, Y. Wang, H. Hu, C.-M. Pun, and H. Gao, "Fgrflow: Learning fine-grained rigidity scene flow from 4d radar point cloud," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 267–276.

[7] A. Khoche, Q. Zhang, Y. Cai, S. S. Mansouri, and P. Jensfelt, "Dogflow: Self-supervised lidar scene flow via cross-modal doppler guidance," *IEEE Robotics and Automation Letters*, 2026.

[8] 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 (RA-L)*, vol. 7, no. 3, pp. 8233–8240, 2022.

[9] Y. Liu, X. Chen, N. Wang, S. Andreev, A. Dvorkovich, R. Fan, and H. Lu, "Self-supervised diffusion-based scene flow estimation and motion segmentation with 4d radar," *IEEE Robotics and Automation Letters*, 2025.

[10] J. Wu, M. Braun, D. Spata, and M. Rottmann, "Tars: Traffic-aware radar scene flow estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 26 075–26 084.

[11] N. J. Abu-Alrub and N. A. Rawashdeh, "Radar odometry for autonomous ground vehicles: A survey of methods and datasets," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4275–4291, 2023.

[12] S. Yao, R. Guan, Z. Peng, C. Xu, Y. Shi, W. Ding, E. G. Lim, Y. Yue, H. Seo, K. L. Man *et al.*, "Exploring radar data representations in autonomous driving: A comprehensive review," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[13] H. Zhen, Z. Gao, R. Jin, X. Guo, and F. Lin, "Far-rio: A fast and robust radar-inertial odometry with isotropic

uncertainty model and dual-observation update pipeline," *IEEE Robotics and Automation Letters*, 2026.

[14] J. Xiang, X. He, Z. Chen, L. Zhang, X. Luo, and J. Mao, "Vgc-rio: A tightly integrated radar-inertial odometry with spatial weighted doppler velocity and local geometric constrained rcs histograms," *IEEE Robotics and Automation Letters*, 2025.

[15] E. Furuno, P. Megarajan, C. Kowalski, T. C. Stratmann, M. Pfingsthorn, and A. Hein, "Claro: A cluster-based method for radar odometry," *IEEE Robotics and Automation Letters*, 2026.

[16] D. C. Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss, "Rai-slam: Radar-inertial slam for autonomous vehicles," *IEEE Robotics and Automation Letters*, 2025.

[17] D. C. Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss, "Multi-session mapping and long-term localization for autonomous vehicles using radar," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 2522–2529, 2026.

[18] J. Morris, Y. Wang, M. Kliniewski, and V. Ila, "Dynosam: Open-source smoothing and mapping framework for dynamic slam," *IEEE Transactions on Robotics*, 2025.

[19] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.

[20] J. Huang, S. Yang, Z. Zhao, Y.-K. Lai, and S.-M. Hu, "Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5875–5884.

[21] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2168–2177.

[22] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "Dynaslam ii: Tightly-coupled multi-object tracking and slam," *IEEE robotics and automation letters*, vol. 6, no. 3, pp. 5191–5198, 2021.

[23] I. Ballester, A. Fontán, J. Civera, K. H. Strobl, and R. Triebel, "Dot: Dynamic object tracking for visual slam," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 11 705–11 711.

[24] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[25] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[26] J. Wang, M. Rünz, and L. Agapito, "Dsp-slam: Object oriented slam with deep shape priors," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1362–1371.

[27] K. M. Judd, J. D. Gammell, and P. Newman, "Multimotion visual odometry (mvo): Simultaneous estimation of camera and third-party motions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3949–3956.

[28] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: A visual dynamic object-aware SLAM system," *arXiv preprint arXiv:2005.11052*, 2020.

[29] J. Zhang, M. Henein, R. Mahony, and V. Ila, "Robust ego and object 6-dof motion estimation and tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5017–5023.

[30] Y. Qiu, C. Wang, W. Wang, M. Henein, and S. Scherer, "Airdos: Dynamic slam benefits from articulated objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8047–8053.

[31] J. Morris, Y. Wang, and V. Ila, "The importance of coordinate frames in dynamic slam," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 755–13 761.

[32] H. Kim, M. Jung, C. Noh, S. Jung, H. Song, W. Yang, H. Jang, and A. Kim, "Hercules: Heterogeneous radar dataset in complex urban environment for multi-session radar slam," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 4649–4656.

[33] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics (T-RO)*, vol. 33, no. 1, pp. 1–21, 2017.

[34] C. Le Gentil and T. Vidal-Calleja, "Continuous latent state preintegration for inertial-aided systems," in *The International Journal of Robotics Research (IJRR)*, 2023.

[35] X. Li, X. Zhang *et al.*, "4D radar-based pose graph SLAM with ego-velocity pre-integration factor," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 8, pp. 5124–5131, 2023.

[36] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems (RSS)*, 2009.

[37] P. J. Best, "A method for registration of 3-d shapes," *IEEE Trans Pattern Anal Mach Vision*, vol. 14, pp. 239–256, 1992.

[38] J. K. Pontes, J. Hays, and S. Lucey, "Scene flow from point clouds with or without learning," in *2020 international conference on 3D vision (3DV)*. IEEE, 2020, pp. 261–270.

[39] W. Yang *et al.*, "Ground-optimized 4D radar-inertial odometry via continuous velocity integration using gaussian process," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[40] J. Zhang, H. Zhuge, Z. Wu *et al.*, "4DRadarSLAM: A 4D imaging radar SLAM system for large-scale environments based on pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.