CAO-RONet: A Robust 4D Radar Odometry with Exploring More Information from Low-Quality Points

Zhiheng Li, Yubo Cui, Ningyuan Huang, Chenglin Pang and Zheng Fang*

Abstract— **Recently, 4D millimetre-wave radar exhibits more stable perception ability than LiDAR and camera under adverse conditions (e.g. rain and fog). However, low-quality radar points hinder its application, especially the odometry task that requires a dense and accurate matching. To fully explore the potential of 4D radar, we introduce a learning-based odometry framework, enabling robust ego-motion estimation from finite and uncertain geometry information. First, for sparse radar points, we propose a local completion to supplement missing structures and provide denser guideline for aligning two frames. Then, a context-aware association with a hierarchical structure flexibly matches points of different scales aided by feature similarity, and improves local matching consistency through correlation balancing. Finally, we present a window-based optimizer that uses historical priors to establish a coupling state estimation and correct errors of inter-frame matching. The superiority of our algorithm is confirmed on View-of-Delft dataset, achieving around a 50% performance improvement over previous approaches and delivering accuracy on par with LiDAR odometry. Our code will be available.**

I. INTRODUCTION

Odometry estimation is an important issue for autonomous driving and mobile robots, aiming to supply precise location information for navigation through correlating sensor data at various times. In recent years, some algorithms [1]–[5] have devoted to applying supervised or self-supervised learning to address odometry estimation problem, achieving results that approach or surpass traditional geometry-based methods [6]–[8]. However, most odometry methods often rely on camera or LiDAR that are easily affected by lighting and weather, making them difficult to handle complex application scenes, such as rain, fog and smoke-filled underground mines.

Due to its high penetration and long-range sensing ability, 4D radar has garnered significant attention and is widely used in perception tasks like 3D detection and tracking [9]–[15]. Similarly, several works [16]–[19] also have tried to achieve end-to-end 4D radar odometry to deal with harsh conditions. However, most of them adhere to LiDAR-based paradigm [1] without fully considering the distinctive features of radar. For example, they usually match adjacent raw radar frames based on geometric distance relationships. Subsequently, the inter-frame registration results, which do not account for long-term motion pattern, are directly used as the predicted ego-motion.

Specifically, most algorithms neglect the following issues: (1) *Sparsity*: Each radar frame generally contains only a few hundred points (about 1% of LiDAR), thus it provides limited geometric information for matching. (2) *Noise*: The position of radar points suffers from noise owing to multipath effects, leading to a "hard" distance-based matching struggle to learn reliable data association. (3) *Continuity*: The state estimation viewed as mere inter-frame matching disrupts the continuity of ego-motion and aggravates the impact of error in degraded scenarios, such as being occluded, where finding enough and effective matching pairs is harder for radar than LiDAR.

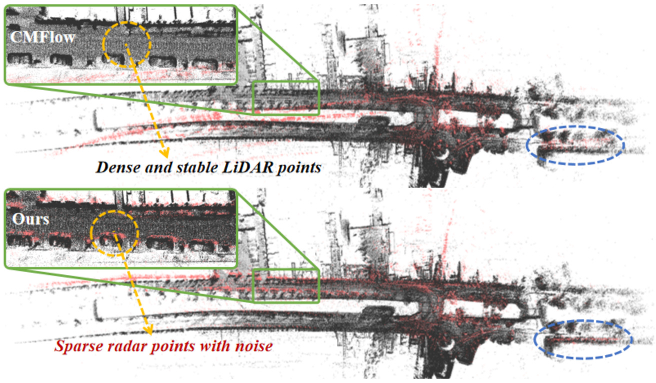


Fig. 1: Comparison of odometry accuracy. The black points are LiDAR map constructed using ground-truth poses, while red points denote radar map assembled from predicted poses.

To solve the above challenges, we propose a deep-learning odometry network named CAO-RONet, designed to be compatible with the unique properties of 4D radar, which consists of three essential ideas: **(1)** ***Local completion***: For sparse and incomplete radar points, an intuitive strategy is to fill empty spaces. Thus, we create many synthetic points that align with regional structure and supply more geometric information for matching process (e.g., yielding 64 synthetic points from 256 raw points supplies 25% additional data). By doing so, denser point pairs can be used to reduce odometry errors, especially in turning. **(2)** ***Context-aware association***: Instead of directly matching points based on distance, we extra consider feature similarity that implicitly compares partial structure and radial velocity. Then, it is softly combined with the distance weight to achieve a resilience registration and alleviate the influence of noise in point positions. Moreover, a sequential modeling is utilized to balance matching information along the multiple directions to suppress outliers and ensure correlation consistency in local area. **(3)** ***Clip-window optimization***: Following a core principle that odometry is continuous state estimation, we unite the notion of window optimizer with the state space model (SSM) to analyze motion patterns cross state sequence and allow historical priors to restrict the pose estimation. This forms a coupling relationship that rectifies poor matching and

This work was supported by the National Natural Science Foundation of China under Grants 62073066, the Fundamental Research Funds for Central Universities under Grant N2226001, and 111 Project under Grant B16009. (Corresponding author: Zheng Fang, e-mail: fangzheng@mail.neu.edu.cn)

The authors are all with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China;

smooths trajectory in scenes with insufficient matched points.

In sum, the contributions of our paper are as follows:

- We design a 4D radar odometry network, named CAO-RONet, to unleash the power of low-quality radar points and implement robust ego-motion estimation over time.
- We first introduce a local Completion to provide denser constraint for matching. A context-aware Association is adopted to flexibly match points with noise and suppress outliers. We also present a clip-window Optimization to couple multiple states across time to correct pose errors.
- Extensive experiments on View-of-Delft dataset demonstrate that the proposed method achieves state-of-the-art performance with around a 50% reduction in root mean square error against previous works, running at 50 FPS.

II. Related Work

A. LiDAR-based Odometry Methods

As a classic algorithm, ICP is widely utilized in traditional LiDAR odometry. It strives to align the points of two frames by searching their corresponding relationship and minimizing the distance errors. Based on error measurement, ICP can be categorized into P2P-ICP [20] and P2Pl-ICP [21], aiming to shorten point-to-point and point-to-plane distances. GICP [7] further intends to combine the advantages of both. Compared to ICP with static assumptions, NDT [8] converts point cloud into probability distributions, exhibiting stronger adaptability in dynamic scenes, but determining all points associations is too time-consuming. Thus, to represent raw points with fewer elements, LOAM [6] selects keypoints from sharp edges and planar surfaces based on curvature, and exploits them to align edge lines and planar patches. Then, Lego-LOAM [22] uses a segmentation module to discard unreliable points and apply planes derived from stable ground points for matching.

Thanks to powerful data encoding and association abilities of neural networks, end-to-end LiDAR odometry has rapidly developed. LONet [23] eases the effect of dynamic objects by a probability mask and constrains network learning through differences in normal vectors between two frames. After that, LodoNet [24] uses image-based feature descriptors to extract keypoint pairs from LiDAR images and match them for pose estimation. Drawing on the idea of ICP iterative optimization, PWCLO-Net [1] proposes a coarse-to-fine strategy to achieve ego-motion refinement utilizing multiple warps. TransLO [3] further applies a window-based Transformer to extract global embeddings for large-scale matching consistency. While the above works get encouraging results with LiDAR in standard conditions, they still face challenges when point degradation occurs due to heavy smoke or adverse weather.

B. 4D Radar-based Odometry Methods

Due to all-weather operational characteristics, some methods try to use 4D radar to implement odometry. For example, regarding the issue of radar frames losing obvious geometric structure, 4DRadarSLAM [25] integrates GICP with spatial probability distribution of each point and develops APDGICP algorithm for scan-to-scan matching. Besides, to alleviate the sparsity of points, [26] employs sliding window to construct a dense radar submap with rich structural information, which is aligned with current frame by NDT [8] for scan-to-submap registration. As the initial learning-based 4D radar odometry, SelfRO [19] presents a self-supervised method that employs consistency losses based on velocity, geometry and distribution to minimize the gap between two frames. 4DRONet [17] decouples radar information encoding with different natures to avoid mutual interference and adopts a velocity-aware cost volume to enable stable matching, even with moving objects. To diminish reliance on costly labels, CMFlow [16] proposes an elaborate cross-modal method that utilizes complementary supervision signals from multi-sensor and other pre-trained models to guide network training. However, these end-to-end works disregard the sparsity and noise of 4D radar, thus the potential of radar odometry has not been fully explored.

III. Methodology

A. Overview

Fig. 2 illustrates the framework of our CAO-RONet. Two sampled adjacent frames are first input into PointNet++ [27] to encode radar information (3D position, radar cross section (RCS), and radial relative velocity (RRV)), resulting in $P_1 = \{p_i = \{x_i, f_i\}\}_{i=1}^{N}$ and $P_2 = \{p_j = \{x_j, f_j\}\}_{j=1}^{N}$, where $x \in \mathbb{R}^3$ denotes 3D coordinates, $f \in \mathbb{R}^C$ is point features. To enhance sparse points, a Local Completion Module (LCM) is proposed to offset M anchor points and generate artificial points, which are combined with P to get densified sets $Q_1 = \{q_i\}_{i=1}^{M+N}$ and $Q_2 = \{q_j\}_{j=1}^{M+N}$. Thereafter, the point sets are split into two groups and separately undergo Context-aware Association Module (CAM) to match point pairs by feature-assisted aligning and correlation balancing, yielding a feature $G \in \mathbb{R}^{(M+N+W)\times C}$ with multi-scale matching information. Finally, we use a Clip-window Optimization Module (COM) to establish a coupling odometry, which adopts bi-directional SSM to optimize state feature g_t derived from G and estimate a quaternion $q \in \mathbb{R}^4$ and translation vector $t \in \mathbb{R}^3$.

B. Offset-based Local Completion

Although some algorithms [28]–[31] are dedicated to point cloud completion, they often focus on repairing missing parts of objects. Thus, directly applying them to sparse radar points from large-scale scenes may cause uncontrolled positions of artificial points and hinder stable local matching. Rather than completing the scene globally, we employ a local scheme that encodes local information of raw points to predict coordinate and feature offsets, creating new ones with confined positions that conform to regional shape and facilitate denser matching.

Specifically, for point clouds $P \in \{P_1, P_2\}$, we adopt the farthest point sampling (FPS) to determine M anchor points $S = \{s_i = \{x_i, f_i\}\}_{i=1}^{M}$. Based on each s_i, we use ball query to search for K neighbors from P and produce a local region $\mathcal{N}_i$, where the point coordinates and features are $X_i \in \mathbb{R}^{K\times 3}$ and $F_i \in \mathbb{R}^{K\times C}$. Then, the max-pooling is applied to process F_i and generate a local feature $\overline{f}_i \in \mathbb{R}^C$, which is adopted to calculate the difference with f_i and mapped to feature offset Δf_i through multi-layer perceptron (MLP):

$$\overline{f}_i = \mathrm{MaxPool}(F_i),\ \Delta f_i = \mathrm{MLP}(f_i - \overline{f}_i) \tag{1}$$

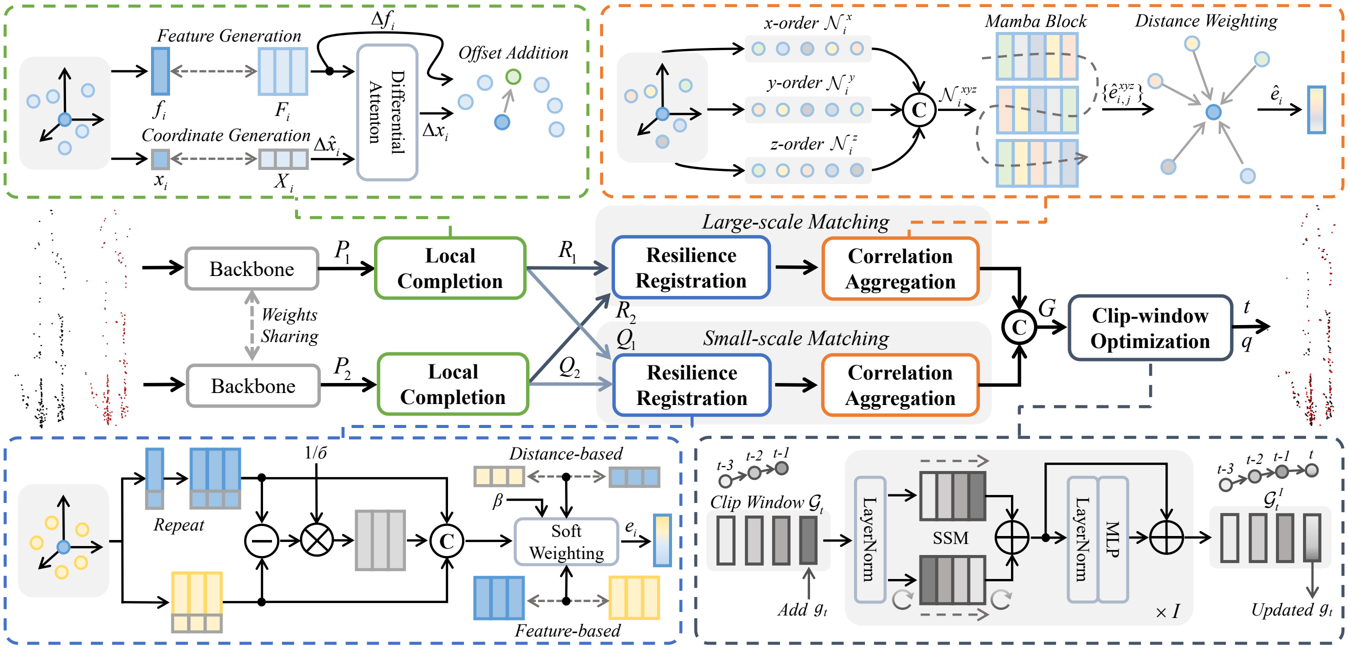


Fig. 2: The overview of our proposed CAO-RONet. At first, the two frames of radar features derived from backbone are fed into LCM to densify sparse points. Then, CAM implements feature-assisted registration to associate point pairs in different scales, followed by correlation balancing to suppress outliers. Finally, COM with sequential state modeling applies historical prior from clip window to constraint the current ego-motion prediction and smooth trajectory.

Similar to the above procedure, we average the coordinates X_i within local region $\mathcal{N}_i$ to get a geometric center $\overline{x}_i$. Next, the initial positional offset $\Delta\hat{x}_i$ is derived by calculating the difference between x_i and $\overline{x}_i$:

$$\overline{x}_i = \text{AvgPool}(X_i),\ \Delta\hat{x}_i = (x_i - \overline{x}_i) \tag{2}$$

To further correct initial offset $\Delta\hat{x}_i$ in a learnable manner, we introduce a differential attention mechanism in which the feature $(\Delta f_i, f_i)$ is normalized and projected into *query* and *key* embeddings. Then, attention weight is computed by dot-product and exerted on *value* $\Delta\hat{x}_i$ to get updated offset Δx_i:

$$\Delta x_i = \text{Softmax}(\frac{(\text{LN}(\Delta f_i)W_q)(\text{LN}(f_i)W_k)^T}{\sqrt{d_k}})\Delta\hat{x}_i \tag{3}$$

where W_q, W_k are linear projection matrices, and LN is layer normalization. The Δf_i and Δx_i are added to anchor point s_i to produce a new point $q_i = \{x_i + \Delta x_i, f_i + \Delta f_i\}$. Finally, these new point are combined with the original P to generate $Q_1 = \{q_i\}_{i=1}^{M+N}$ and $Q_2 = \{q_j\}_{j=1}^{M+N}$ for a denser matching.

C. Hierarchical Context-aware Association

To avoid excessive reliance on local similarity and neglect of large-scale consistency in inter-frame matching, we adopt a hierarchical strategy that divides matching process into two groups: a small-scale ($Q_1 \rightarrow Q_2$) and large-scale ($R_1 \rightarrow R_2$) registration, where R_1 and R_2 are sampled from Q_1 and Q_2 using FPS. Subsequently, we apply resilience registration and correlation propagation to each group independently. Due to the same process, we only introduce ($Q_1 \rightarrow Q_2$) for brevity.

Resilience Registration. For the point $q_i = \{x_i, f_i\} \in \mathbb{R}^{3+C}$ in Q_1, we find K neighbors $\mathcal{N}_i = \{q_{i,j} = \{x_{i,j}, f_{i,j}\}\}_{j=1}^{K} \in \mathbb{R}^{K\times(3+C)}$ in Q_2. Later, we determine the difference between q_i and $q_{i,j}$ and obtain a normalized feature $d_{i,j}$ by scalar σ, which stands for feature deviations across channels and local groups, thereby reducing the influence of excessive deviation.

$$d_{i,j} = \frac{q_i - q_{i,j}}{\sigma + \epsilon}, \sigma = \sqrt{\frac{1}{D}\sum_{i=1}^{M+N}\sum_{j=1}^{K}(q_i - q_{i,j})^2} \tag{4}$$

where $D = (M+N)\times K\times(3+C)$, and ϵ is a small constant used to ensure numerical stability. Then, the $d_{i,j}$, q_i and $q_{i,j}$ are merged and processed through MLP to get a contrastive feature $h_{i,j}$ that measures the relationship between q_i and its neighbor $q_{i,j}$. To assemble discrete set $\{h_{i,j}\}_{j=1}^{K}$ into a local correlation vector $e_i \in \mathbb{R}^C$, we produce two types of weights $(w_{i,j}^d \in \mathbb{R}^C, w_{i,j}^f \in \mathbb{R}^C)$ and then apply soft weighted sum to $h_{i,j}$, using a learnable parameter β to adjust the confidence in both the spatial distance and feature similarity, as follows:

$$h_{i,j} = \text{MLP}(d_{i,j} \oplus q_i \oplus q_{i,j}) \tag{5}$$

$$w_{i,j}^d = \text{MLP}(x_i - x_{i,j}), w_{i,j}^f = \text{MLP}\langle f_i, f_{i,j}\rangle \tag{6}$$

$$e_i = \beta\sum_{j=1}^{K} h_{i,j} \odot w_{i,j}^f + (1-\beta)\sum_{j=1}^{K} h_{i,j} \odot w_{i,j}^d \tag{7}$$

where $\oplus$ and $\odot$ mean channel concatenation and dot product. $\langle\cdot\rangle$ denotes similarity calculation. As a result, after combining vector e_i of each point, we obtain the correlation embedding $E = \{e_i\}_{i=1}^{M+N}$ for Q_1, which accounts for feature difference to enable more resilient matching compared to relying solely on rigid distance affected by random noise of point position.

Correlation Aggregation. It is worth noting that some points may have insufficient matching due to occlusion or isolation, giving rise to outliers in correlation embedding E. A simple way to suppress them is to aggregate the correlations within a local region $\mathcal{N}_i = \{q_{i,j} = \{x_{i,j}, e_{i,j}\}\}_{j=1}^{K}$ surrounding each point $q_i = \{x_i, e_i\}$ of Q_1 and then update e_i to improve local consistency. However, this approach may still be affected by outliers in $\mathcal{N}_i$. To tackle this matter, we advocate sorting the points within $\mathcal{N}_i$ through coordinates $\{x_{i,j}\}$ and applying an RNN-like sequential modeling to adjust each embedding $e_{i,j}$ by neighbors, which is based on the fact that adjacent points exhibit similar matching situations in most cases.

Specifically, we first arrange points $\mathcal{N}_i$ along the x, y and z axes through their coordinates and generate three sequences $(\mathcal{N}_i^x, \mathcal{N}_i^y, \mathcal{N}_i^z)$, which are concatenated to obtain $\mathcal{N}_i^{xyz}$ with coordinates $\{x_{i,j}^{xyz}\}_{j=1}^{3K}$ and embeddings $\{e_{i,j}^{xyz}\}_{j=1}^{3K}$. Then, a Mamba block [32] with a global receptive field and parallel processing is used to encode $\{e_{i,j}^{xyz}\}$ sequentially and produce balanced embeddings $\{\hat{e}_{i,j}^{xyz}\}$. Later, to aggregate $\{\hat{e}_{i,j}^{xyz}\}$ and refine correlation e_i, we calculate Euclidean distance weights $w_{i,j}$ between q_i and its neighbors $\mathcal{N}_i^{xyz}$, which are assigned to $\{\hat{e}_{i,j}^{xyz}\}$ to get a updated correlation embedding $\hat{e}_i$ in Eq. 8.

$$w_{i,j} = \frac{1}{||x_i - x_{i,j}^{xyz}||_2}, \hat{e}_i = \mathrm{MLP}(\sum_{j=1}^{3K}(w_{i,j} \odot \hat{e}_{i,j}^{xyz}) \oplus e_i) \quad (8)$$

Finally, we combine the correlation embeddings from both $(Q_1 \rightarrow Q_2)$ and $(R_1 \rightarrow R_2)$ into feature $G \in \mathbb{R}^{(M+N+W) \times C}$ that contains multi-scale alignment. W is the number of R_1.

D. Bi-directional Clip-window Optimization

Different from previous method [17] that pools correlation embedding G to obtain state quantity $g_t \in \mathbb{R}^{1 \times C}$ and directly predict ego-motion, we argue that it is necessary to introduce historical states as constraints to handle transient degradation issue. Thus, we construct a clip window of maximum length L to store states. Its update mechanism is denoted as follows:

$$\mathcal{G}_t = \begin{cases} \{g_t\}, & \text{if } t \bmod L = 0 \\ \{g_{t-(t \bmod L)}, ..., g_t\}, & \text{otherwise} \end{cases} \quad (9)$$

Specifically, after obtaining the raw current state g_t, it is first added to $\mathcal{G}_t$. If clip window is full, the past states are cleared, and the window is refilled to prevent the prolonged influence of low-quality states. To leverage historical priors to optimize g_t, we define a discretized state space model (SSM) [33] as in Eq. 10, where the $\overline{\mathbf{A}} \in \mathbb{R}^{C \times C}$, $\overline{\mathbf{B}} \in \mathbb{R}^{C \times 1}$ and $\mathbf{C} \in \mathbb{R}^{1 \times C}$ denote learnable parameters. Note that the hidden state h_{t-1} that implicitly represents motion pattern is derived from past states of $\mathcal{G}_t$ by the same way, and this process is performed in parallel with a convolution kernel [34] rather than recursion.

$$h_t = \overline{\mathbf{A}} h_{t-1} + \overline{\mathbf{B}} g_t, \;\; \hat{g}_t = \mathbf{C} h_t \quad (10)$$

Based on SSM, we further introduce a bidirectional modeling block as shown in Eq. 11 and Eq. 12, where ordered sequence $\mathcal{G}_t$ is fed into the SSM in both forward and reverse directions to enable network to learn a broader range of motion patterns.

$$\hat{\mathcal{G}}_t^i = \mathrm{SSM}(\mathrm{LN}(\mathcal{G}_t^{i-1})) + \mathcal{F}(\mathrm{SSM}(\mathcal{F}(\mathrm{LN}(\mathcal{G}_t^{i-1})))) \quad (11)$$

$$\mathcal{G}_t^i = \mathrm{MLP}(\mathrm{LN}(\hat{\mathcal{G}}_t^i)) + \hat{\mathcal{G}}_t^i \quad (12)$$

where i and $\mathcal{F}$ denote i-th block ($i \in \{1, ..., I\}$) and reverse sorting. To the end, we separate updated g_t from final output $\mathcal{G}_t^I$ and use two MLPs to estimate the quaternion $q \in \mathbb{R}^4$ and translation vector $t \in \mathbb{R}^3$.

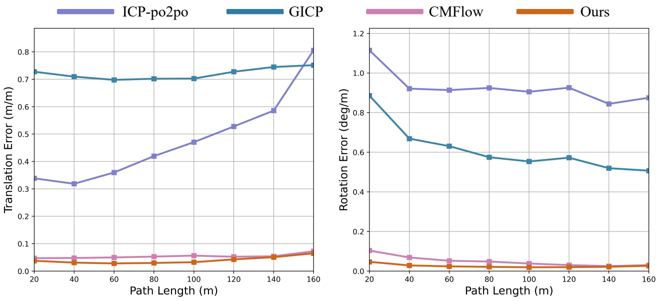


Fig. 3: Average translational and rotational errors on the test sequences of VoD in the length of 20, 40, ..., 160m.

E. Implementation Details

Loss Functions. The loss functions for rotation and translation are defined in Eq. 13, where q^{gt} and t^{gt} are the ground-truth quaternion and translation vector, respectively.

$$\mathcal{L}_q = ||q - q^{gt}||_2, \mathcal{L}_t = ||t - t^{gt}||_2 \quad (13)$$

Referring to the previous work [17], we use two learnable parameters, w_q and w_t, which adjust losses $\mathcal{L}_q$ and $\mathcal{L}_t$ during training to account for differences in scale and units between q and t. Consequently, the overall loss can be formulated as:

$$\mathcal{L} = \mathcal{L}_q \exp(-w_q) + w_q + \mathcal{L}_t \exp(-w_t) + w_t \quad (14)$$

Data Augmentation. In data processing, we clear out radar points outside the field of view of image and constrain them to the height range of [-3m, 3m] to retain reliable points. To increase data diversity, we flip training sequences to produce new trajectories with reverse ego-motion, and apply random offsets to points and ground-truth pose matrix during training

Training & Inference. The model is trained for 60 epochs on a NVIDIA RTX 4090 GPU using an Adam optimizer with a starting learning rate of 1×10^{-3} and a decay rate of 0.9 per epoch. Then, trainable parameters w_q and w_t are initialized to -2.5 and 0.0, and the clip window length is set to 5. The default numbers of sampled raw points N, completion points M and large-scale points W are 256, 64 and 64.

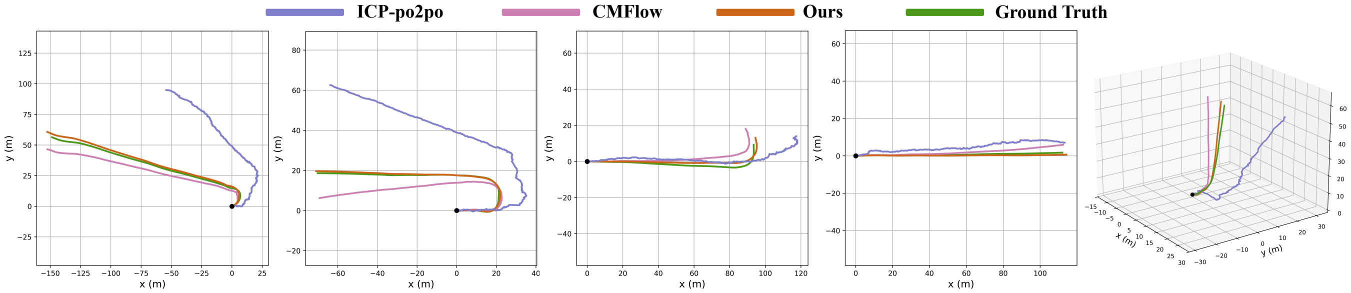
IV. EXPERIMENTS

Datasets. We conduct in-depth experiments on View-of-Delft dataset (VoD) [35], which contains 8,682 point cloud frames (captured by a Velodyne HDL-64 LiDAR and ZF FR-Gen21 4D radar) along with the corresponding extrinsic parameters and odometry information. Based on the continuity of frame, the VoD can be divided into 24 sequences, with 6,964 frames used as the training set and 1,718 frames from five sequences (00, 03, 04, 07 and 23) used as the test set.

Evaluation Metric. The relative pose error (RPE) is usually used to measure the difference between the ground truth and predicted poses over the specific intervals or distances. Based on it, we calculated the root mean square error (RMSE) for rotation (°/m) and translation (m/m) across test sequences with lengths ranging from 20m to 160m in 20m increments.

TABLE I: 4D radar odometry experiment results on View-of-Delft (VoD) dataset. Like previous works, we keep two decimal places for odometry metric in this table, but three decimal places for ablation studies to better illustrate performance changes.

Method		00		03		04		07		23		Mean		Time (ms)
		t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	
Classical-based	ICP-po2po	0.57	1.23	0.38	0.98	0.21	1.15	0.30	1.75	0.18	0.49	0.33	1.12	3.80
	ICP-po2pl	0.69	1.67	0.41	2.16	0.39	1.86	0.74	2.77	1.38	1.07	0.72	1.91	1.11
	GICP	0.41	0.42	0.46	0.65	0.31	0.38	0.37	0.29	0.79	0.17	0.47	0.38	1.29
	NDT	0.52	0.63	0.56	1.52	0.47	0.91	0.69	0.51	0.52	0.37	0.55	0.79	1.02
LiDAR-based	A-LOAM w/o mapping	-	-	-	-	0.14	0.35	0.13	0.74	0.25	1.39	-	-	4.70
	LO-Net	0.81	0.81	1.12	1.89	0.23	0.46	0.19	0.21	0.53	1.07	0.58	0.89	11.6
4D Radar-based	RaFlow	0.61	0.84	0.87	1.98	0.07	0.45	0.07	0.04	0.42	1.16	0.41	0.90	36.3
	4DRO-Net	0.08	**0.03**	0.06	0.05	0.08	0.07	0.05	0.03	0.10	0.15	0.07	0.07	10.8
	CMFlow†	**0.04**	0.05	0.07	0.09	0.06	0.09	0.03	0.04	0.09	0.14	0.06	0.08	30.4
	Ours	0.05	**0.03**	**0.02**	**0.03**	**0.03**	**0.05**	**0.02**	**0.02**	**0.04**	**0.06**	**0.03**	**0.04**	20.2



Sequence 00 Sequence 03 Sequence 04 Sequence 07 Sequence 23

Fig. 4: The trajectory visualization of our CAO-RONet with other methods on sequences 00, 03, 04, 07 and 23, respectively.

TABLE II: Odometry experiments using different sensors on VoD dataset. Sequence division follows 4DRVO-Net [18].

Method	A-LOAM w/o mapping	4DRVO-Net	CMFlow	**Ours**
Sensor	LiDAR	Radar + Camera	Radar Only	Radar Only
Mean t_{rel}	**0.06**	0.08	0.11	0.07
Mean r_{rel}	0.10	0.07	0.31	**0.05**

A. Quantitative Results.

Consistent with [17], our method is compared with classic geometry-based algorithms, such as ICP-po2po, ICP-po2pl, GICP, NDT and LOAM in Tab. I. Besides, we also compare learning-based methods initially designed for LiDAR points (LO-Net) and 4D radar (RaFlow, 4DRO-Net, and CMFlow). Specifically, we retrained CMFlow† instead of directly using the pre-trained model like [17] to avoid unfairness caused by differences in the training sequence split. The results confirm that classic approaches effective for LiDAR, perform poorly or cannot complete all sequences due to the extremely sparse points of 4D radar. Compared to end-to-end methods, owing to improvements made to radar natures, we obtain the lowest mean t_{rel} and r_{rel}, and the errors of all sequences are more balanced. Moreover, Fig. 3 shows the average segment errors on test sequences, proving the advantages of our method over previous works. Finally, we validate that our low-cost radar-only method achieves competitive results compared to other methods that combine camera or use LiDAR in Tab. II.

B. Qualitative Results

To provide a more intuitive comparison, Fig. 4 visualizes the trajectories of several methods across different sequences. Despite ICP-po2po getting the best t_{rel} among classic methods, it suffers from obvious deviations due to the challenges in building stable matching caused by noise and the sparsity of radar points. While CMFlow, as a state-of-the-art method, displays considerable improvement, it still produces notable error during turn in sequences 00 and 03. By contrast, owing to a denser matching and coupling optimization, our method exhibits smaller error during linear and rotational movement, particularly in sequences 03 and 07. Since the quality of map constructed by odometry can manifest ego-motion accuracy, we apply the predicted poses to LiDAR points to build dense maps that allow for easier comparison. As shown in Fig. 6, CMFlow produces many ghost effects, whereas our maps are clearer, validating our odometry is more stable and accurate.

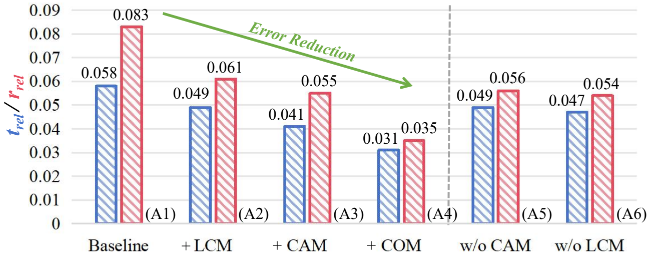


Fig. 5: The effect of different modules on VoD dataset.

C. Ablation Studies

To thoroughly analyse the impact of each module and its design strategy on our method's performance (mean t_{rel} and r_{rel}), we conduct a series of ablation studies on VoD dataset. **Model Components.** In Fig. 5, we can observe that adopting LCM for local completion (A2) results in a moderate reduction in t_{rel} and r_{rel}. We attribute this to enhanced geometric information between two frames, which helps network form more effective matching pairs to constrain motion estimation, particularly during turns. Then, the elastic matching in CAM reduces the impact of noise, while the correlation aggregation balances the matching information of each point to mitigate

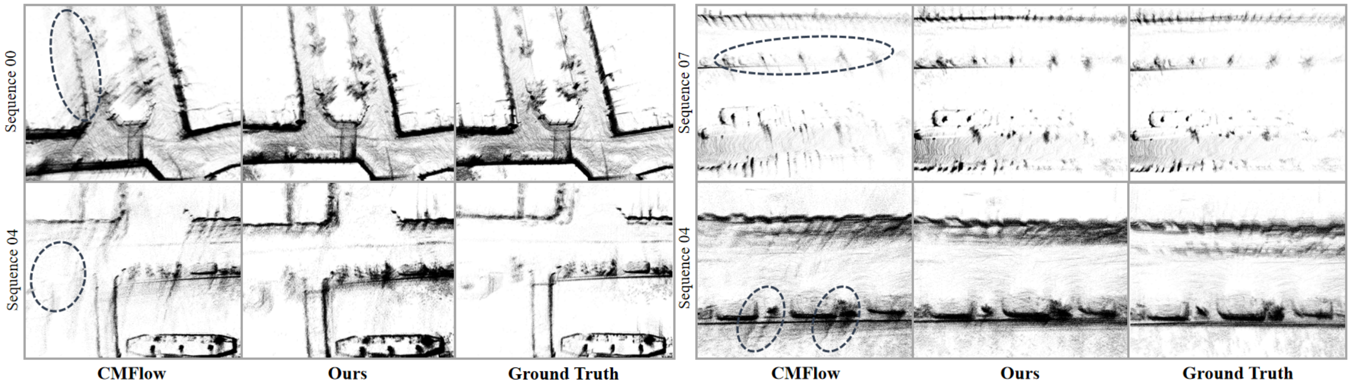


Fig. 6: Comparison of local maps that are constructed from LiDAR points based on predicted radar odometry.

TABLE III: Ablation studies on Context-aware Association.

	Hierarchy	Matching		Aggregation	t_{rel}	r_{rel}
		Distance	Feature			
B1		✓			0.049	0.056
B2	✓	✓			0.038	0.055
B3		✓	✓		0.039	0.046
B4	✓	✓	✓		0.035	0.041
B5	✓	✓	✓	✓	**0.031**	**0.035**

TABLE IV: Ablation studies on Clip-window Optimization. Uni- and bi- mean uni-directional and bi-directional SSM.

	Cross-Attn	SSM		Slide	Clip	t_{rel}	r_{rel}
		Uni-	Bi-				
C1	✓				✓	0.038	0.044
C2		✓			✓	0.036	0.038
C3			✓	✓		0.032	0.037
C4			✓		✓	**0.031**	**0.035**

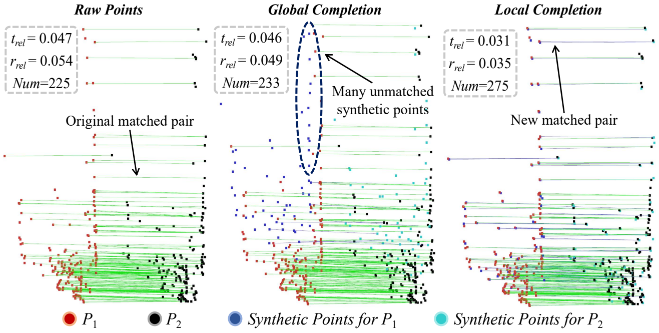


Fig. 7: Comparison of matching results within a fixed range ($1.5m$) with different completion methods. *Num* is the number of matched pairs. t_{rel} and r_{rel} are mean error of test set.

outlier effects, thereby improving odometry accuracy in A3. When COM is employed for coupling state optimization, the best result is obtained in A4. Thus, we think that considering historical states is a key factor for the robustness of odometry, especially in degraded situations, where historical priors can help suppress severe errors. Finally, in A5 and A6, removing CAM or LCM will lead to performance degradation, proving that each module is complementary and indispensable.

Global Completion vs. Local Completion. Previous completion methods [28], [29] usually complete the entire object or scene globally. Although this approach can introduce some additional geometric information, it will result in an unstable distribution of generated points, making it difficult to search correspondence between two frames (in the middle of Fig. 7). In contrast, our local completion restricts point generation to specific areas while adhering to local structural characteristic. Thus, it ensures more consistent created points across frames, enabling denser matching (in the right of Fig. 7).

Context-aware Association. We conduct more detailed ablation studies of CAM in Tab. III. As displayed in B1 and B3, integrating feature-based matching, which takes into account comprehensive information, outperforms the purely distance-based matching due to the unsteady positions of radar points. Then, B2 shows that hierarchical matching encourages model to perceive correspondences at larger scales and contributes to the reduction of t_{rel}. Furthermore, B4 and B5 reflect that sequential modeling can ensure adjacent points within a local area share similar correlation information, thereby preventing outliers from misleading ego-motion estimation.

Clip-window Optimization. In order to capture continuous ego-motion, we introduce a state optimizer to save historical state quantities in clip window and use them to update current state. Ablation studies on the design strategy of optimizer are presented in Tab. IV. In C1, we first try to use cross-attention to establish relationship among multiple states and renew the current one. However, this feature-similarity-based method is challenging to capture the causal and temporal dependencies between states, thus limiting accuracy improvement. Inspired by global sequence modeling of the state space model (SSM), we adopt it to optimize the current state based on prior states and surpass C1. Later, a bidirectional SSM is designed and achieves a better result in C4 since it models richer motion patterns by flipping state order. Finally, owing to clearing the accumulated states within fixed intervals, clip window avoids the sustained influence of inferior states and has advantages over the slide window in our task, as verfied in C3 and C4.

V. CONCLUSIONS

In this article, we analyze the unique characteristics of 4D radar and present an odometry network customized for low-quality radar points. It can not only supply denser constraints for matching by local completion but also utilize the feature-assisted registration and correlation balancing to alleviate the impact of noise and outlier. Finally, an ego-motion that aligns with motion trends is estimated by window-based optimizer. The experiments show that our method achieves state-of-the-art results against past traditional and learning-based works.

References

[1] G. Wang, X. Wu, Z. Liu, and H. Wang, "Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15910–15919, 2021.

[2] Y. Almalioglu, A. Santamaria-Navarro, B. Morrell, and A.-A. Agha-Mohammadi, "Unsupervised deep persistent monocular visual odometry and depth estimation in extreme environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3534–3541, IEEE, 2021.

[3] J. Liu, G. Wang, C. Jiang, Z. Liu, and H. Wang, "Translo: A window-based masked point transformer framework for large-scale lidar odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1683–1691, 2023.

[4] J. Dai, X. Gong, Y. Li, J. Wang, and M. Wei, "Self-supervised deep visual odometry based on geometric attention model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3157–3166, 2022.

[5] J. Nubert, S. Khattak, and M. Hutter, "Self-supervised learning of lidar odometry for robotic applications," in *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 9601–9607, IEEE, 2021.

[6] J. Zhang, S. Singh, *et al.*, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.

[7] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *Robotics: science and systems*, vol. 2, p. 435, Seattle, WA, 2009.

[8] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3, pp. 2743–2748, IEEE, 2003.

[9] Y. Chae, H. Kim, and K.-J. Yoon, "Towards robust 3d object detection with lidar and 4d radar fusion in various weather conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15162–15172, 2024.

[10] W. Xiong, J. Liu, T. Huang, Q.-L. Han, Y. Xia, and B. Zhu, "Lxl: Lidar excluded lean 3d object detection with 4d imaging radar and camera fusion," *IEEE Transactions on Intelligent Vehicles*, 2023.

[11] M. Zeller, D. C. Herraez, J. Behley, M. Heidingsfeld, and C. Stachniss, "Radar tracker: Moving instance tracking in sparse and noisy radar point clouds," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[12] K. Yoneda, R. Shiraki, K. Hariya, H. Inoshita, R. Yanase, and N. Suganuma, "Fast 3d object detection for 4d imaging radar integrating image map features using semi-supervised learning," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1367–1372, IEEE, 2024.

[13] J. Liu, G. Ding, Y. Xia, J. Sun, T. Huang, L. Xie, and B. Zhu, "Which framework is suitable for online 3d multi-object tracking for autonomous driving with automotive 4d imaging radar?," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1258–1265, IEEE, 2024.

[14] L. Zheng, S. Li, B. Tan, L. Yang, S. Chen, L. Huang, J. Bai, X. Zhu, and Z. Ma, "Rcfusion: Fusing 4-d radar and camera with bird's-eye view features for 3-d object detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.

[15] J. Liu, Q. Zhao, W. Xiong, T. Huang, Q.-L. Han, and B. Zhu, "Smurf: Spatial multi-representation fusion for 3d object detection with 4d imaging radar," *IEEE Transactions on Intelligent Vehicles*, 2023.

[16] 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*, pp. 9340–9349, 2023.

[17] S. Lu, G. Zhuo, L. Xiong, X. Zhu, L. Zheng, Z. He, M. Zhou, X. Lu, and J. Bai, "Efficient deep-learning 4d automotive radar odometry method," *IEEE Transactions on Intelligent Vehicles*, 2023.

[18] G. Zhuoins, S. Lu, L. Xiong, H. Zhouins, L. Zheng, and M. Zhou, "4drvo-net: Deep 4d radar–visual odometry using multi-modal and multi-scale adaptive fusion," *IEEE Transactions on Intelligent Vehicles*, 2023.

[19] H. Zhou, S. Lu, and G. Zhuo, "Self-supervised 4-d radar odometry for autonomous vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 764–769, 2023.

[20] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, Spie, 1992.

[21] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[22] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, IEEE, 2018.

[23] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "Lo-net: Deep real-time lidar odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8473–8482, 2019.

[24] C. Zheng, Y. Lyu, M. Li, and Z. Zhang, "Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation," in *Proceedings of the 28th ACM international conference on multimedia*, pp. 2391–2399, 2020.

[25] J. Zhang, H. Zhuge, Z. Wu, G. Peng, M. Wen, Y. Liu, and D. Wang, "4dradarslam: A 4d imaging radar slam system for large-scale environments based on pose graph optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8333–8340, IEEE, 2023.

[26] X. Li, H. Zhang, and W. Chen, "4d radar-based pose graph slam with ego-velocity pre-integration factor," *IEEE Robotics and Automation Letters*, 2023.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[28] Z. Chen, F. Long, Z. Qiu, T. Yao, W. Zhou, J. Luo, and T. Mei, "Anchorformer: Point cloud completion from discriminative nodes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13581–13590, 2023.

[29] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *European conference on computer vision*, pp. 416–432, Springer, 2022.

[30] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1939–1948, 2020.

[31] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12498–12507, 2021.

[32] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[33] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[34] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré, "Combining recurrent, convolutional, and continuous-time models with linear state space layers," *Advances in neural information processing systems*, vol. 34, pp. 572–585, 2021.

[35] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.

Manuscript 1999 submitted to 2025 IEEE International Conference on
Robotics and Automation (ICRA). Received September 15, 2024.