

DNOI-4DRO: Deep 4D Radar Odometry with Differentiable Neural-Optimization Iterations

Shouyi Lu*, Huanyu Zhou*, Guirong Zhuo[†], Xiao Tang

School of Automotive Studies, Tongji University
{2210803, 2231634, zhuoguirong, 2433055}@tongji.edu.cn

Abstract

A novel learning-optimization-combined 4D radar odometry model, named DNOI-4DRO, is proposed in this paper. The proposed model seamlessly integrates traditional geometric optimization with end-to-end neural network training, leveraging an innovative differentiable neural-optimization iteration operator. In this framework, point-wise motion flow is first estimated using a neural network, followed by the construction of a cost function based on the relationship between point motion and pose in 3D space. The radar pose is then refined using Gauss-Newton updates. Additionally, we design a dual-stream 4D radar backbone that integrates multi-scale geometric features and clustering-based class-aware features to enhance the representation of sparse 4D radar point clouds. Extensive experiments on the VoD and Snail-Radar datasets demonstrate the superior performance of our model, which outperforms recent classical and learning-based approaches. Notably, our method even achieves results comparable to A-LOAM with mapping optimization using LiDAR point clouds as input.

Introduction

Odometry is critical to autonomous driving systems, particularly in GPS-denied environments. It involves using consecutive images or point clouds to estimate the relative pose transformation between two frames. Most existing odometry methods focus primarily on 2D visual odometry (Teed and Deng 2021a; Teed, Lipson, and Deng 2023; Lipson and Deng 2024; Wang et al. 2024; Chen et al. 2024) or 3D LiDAR (Wang et al. 2021b; Li et al. 2019; Wang et al. 2022; Liu et al. 2023; Ali et al. 2023; Deng et al. 2023; Zhang et al. 2024). However, the inherent characteristics of cameras and LiDARs make these methods vulnerable to challenging weather conditions such as rain, snow, or fog, compromising their robustness in these scenarios.

4D millimeter-wave radar, as an emerging automotive sensor, presents distinct advantages, including robustness under challenging weather and illumination conditions, the ability to measure object velocities and cost-effectiveness (Pan et al. 2024; Ding et al. 2024a,b; Lin et al. 2024; Peng

et al. 2024; Guan et al. 2024). These advantages have driven extensive research into 4D Radar Odometry (4DRO). Previous 4D radar odometry works can be classified into two categories: classical methods (Zhang et al. 2023; Li, Zhang, and Chen 2023; Wu et al. 2024) and learning-based methods (Zhuo et al. 2024; Lu et al. 2024). Classical 4DRO systems typically consist of point cloud association and nonlinear optimization. Nonlinear optimization is critical in enhancing accuracy by integrating continuous sensor observations to optimize long-range trajectories. However, these methods have limited robustness because of the extracted features' poor quality and low resolution (Zhang et al. 2023; Li, Zhang, and Chen 2023). With the development of deep learning, some methods (Lu et al. 2024; Li et al. 2025) attempt to utilize learning-based methods for 4D radar pose estimation. These methods extract features from two point clouds using a feature extraction network, perform point cloud matching through a feature association network, and finally estimate the pose using a decoding network. Although this direct pose regression method, which does not rely on explicit optimization, can sometimes be more robust, its accuracy often shows limitations across various environments.

To address these problems, we propose DNOI-4DRO, a novel learning-optimization combined 4D radar odometry model. DNOI-4DRO integrates traditional geometric optimization into end-to-end neural network training through an innovatively designed differentiable neural-optimization iteration operator, leveraging the strengths of both classical methods and deep networks. Specifically, we estimate the point-wise motion flow using a neural network, construct the cost function based on the motion and pose relationships of points in 3D space, and update the radar pose using the Gauss-Newton to maximize its compatibility with the current point motion flow estimate. Furthermore, to extract robust 4D radar point features, we propose a novel feature extraction network that enhances the representation of radar point clouds by integrating multi-scale geometric features with clustering-based class-aware features, thereby improving motion estimation in complex environments. We perform extensive evaluation across two datasets, demonstrating state-of-the-art performance in all cases. To summarize, our main contributions are as follows:

- (1) We propose DNOI-4DRO, a novel framework for

*These authors contributed equally.

[†]Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

4D radar odometry. By exploiting the relationship between point motion and pose, DNOI-4DRO couples a geometric optimizer with a neural network to construct a fully end-to-end architecture. (2) We specially design an efficient radar feature extractor comprising a dual-stream 4D radar backbone, which extracts radar features with two representations. (3) Finally, our method is demonstrated on VoD (Palffy et al. 2022) and Snail-Radar (Huai et al. 2024) datasets. Our method outperforms all classical and learning-based 4D radar odometry methods, achieving competitive results with A-LOAM, which uses LiDAR point clouds as input.

Related Work

Classical 4D Radar Odometry

Due to the similarity in spatial representation between 4D radar and LiDAR point clouds, most existing classical 4D radar odometry methods either directly adopt LiDAR odometry methods or modify them accordingly. Iterative Closest Point (ICP) is the most widely used LiDAR odometry method, computing relative pose transformations by minimizing the Euclidean distance between two point clouds. Based on error measurement, ICP can be categorized into ICP-point2point (Besl and McKay 1992) and ICP-point2plane (Chen and Medioni 1992), aiming to shorten point-to-point and point-to-plane distances. Generalized ICP (GICP) (Segal, Haehnel, and Thrun 2009) further intends to combine the advantages of both. Building on GICP, 4DRadarSLAM (Zhang et al. 2023) models the spatial probabilistic distribution of radar points based on uncertainties in range and azimuth angle measurements. It proposes an Adaptive Probability Distribution-GICP (APDG-ICP) method based on the modeling results. RIV-SLAM (Wang, May, and Nuechter 2024) further refines this by considering the impact of the angle of arrival on radar measurement uncertainty to achieve a more accurate spatial probabilistic distribution of radar points. In addition to ICP, the Normal Distribution Transform (NDT) (Stoyanov et al. 2012) is a widely used LiDAR odometry method. NDT represents point clouds with Gaussian distributions, converting pose estimation into an optimization problem that minimizes the Gaussian distribution error between the source and target point clouds. Expanding upon NDT, Li et al. (Li, Zhang, and Chen 2023) integrate the measurement uncertainties of 4D radar points into the calculations for the mean and variance of the normal distribution, aiming to reduce degradation effects caused by sparse 4D radar point clouds. Additionally, 4D iRIOM (Zhuang et al. 2023) introduces a point cloud registration method based on a distribution-to-multi-distribution distance metric, effectively addressing the sparsity issue of 4D radar point clouds and enhancing pose estimation robustness.

Our method draws on the cost function modeling approach of the ICP-point2point (Besl and McKay 1992), a simpler optimization problem that avoids extracting complex representations like line and plane features. Instead of relying on nearest-neighbor queries to find matching points, our method uses a neural network to estimate the point motion field. This approach mitigates the challenging non-

bijection correspondence between two point clouds caused by the sparsity and noise inherent in 4D radar point clouds.

Deep 4D Radar Odometry

Deep learning has made significant strides in visual and LiDAR odometry, yet 4D radar odometry remains a challenging problem. 4DRONet (Lu et al. 2024) proposes a sliding-window-based hierarchical optimization method to estimate and refine poses in a coarse-to-fine manner. SelfRO (Zhou, Lu, and Zhuo 2023) introduces a self-supervised 4D radar odometry method that uses radar point velocity information to construct a novel velocity-aware loss function, effectively guiding network training. CMFlow (Ding et al. 2023) introduces a cross-modal supervision approach for estimating 4D radar scene flow, simultaneously providing pose estimation as an intermediate output. CAO-RONet (Li et al. 2025) introduces a sliding-window-based optimizer that leverages historical priors to enable coupled state estimation and correct inter-frame matching errors. Although these methods show promising results, they often suffer from limited accuracy across diverse environments, and their interpretability remains constrained.

Moreover, we are inspired by the “Differentiable Recurrent Optimization-Inspired Design” proposed by DROID-SLAM (Teed and Deng 2021a) for visual odometry. The work combines iterative visual correspondence updates with differentiable bundle adjustments to optimize pose estimation. This concept has also been applied to optimizing pixel-level 3D motion (Teed and Deng 2021b). However, both works are based on RAFT (Teed and Deng 2020), which performs optical flow updates and optimizes variables in 2D space. This paper introduces the first neural-optimization module for 4D radar-based 3D point cloud odometry by tightly coupling RAFT/DROID-style differentiable iterations with Bundle Adjustment.

DNOI-4DRO

We propose a backbone network for pose estimation between two 4D radar point clouds and then construct a 4DRO system upon it. We provide an overview in Figure 1. The backbone network firstly extracts robust point and context features for each 4D radar point cloud. Then, the feature correlation volume is constructed using point features. Next, the differentiable neural-optimization iteration operator is introduced for pose estimation and refinement. Finally, the network outputs the pose $\mathbf{T}^1 \in \mathbb{R}^{4 \times 4}$ and $\mathbf{T}^2 \in \mathbb{R}^{4 \times 4}$ of the two 4D radar point clouds in the world coordinate system.

Feature Extraction

We employ two radar feature extraction networks with identical architectures but distinct functions to extract the point and context features from the 4D radar point cloud. The context features are provided as input to the recurrent iteration operator, whereas the point features are used to evaluate the point similarity using a dot product. To obtain fine-grained 4D radar point features, we propose a dual-stream 4D radar encoding network that integrates multi-scale geometric feature extraction, clustering-based class-aware feature extrac-

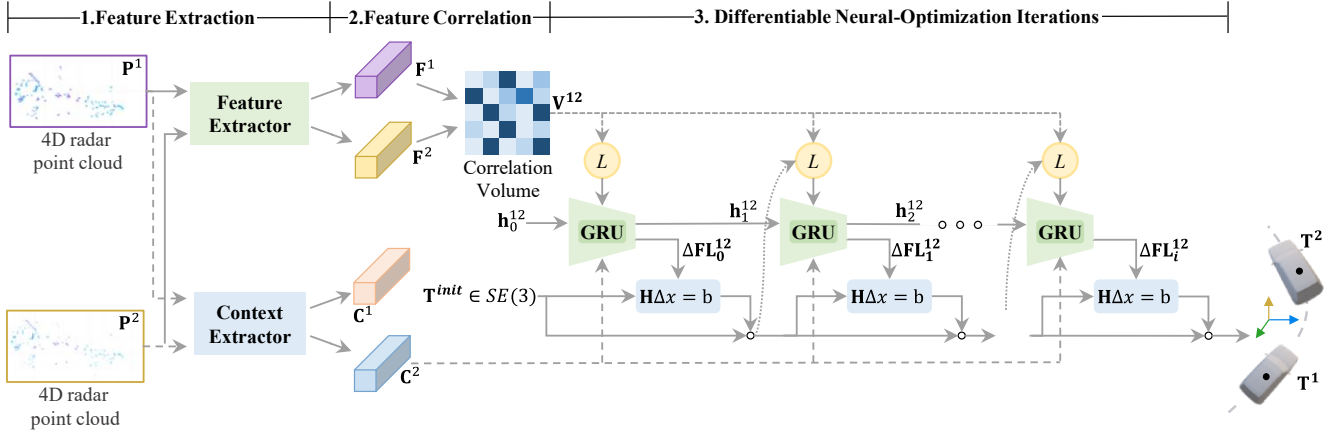


Figure 1: Overview of our backbone. (1) The feature and context extractors encode the point and context features of the input point cloud, respectively. (2) The feature correlation module constructs an all-pair correlation volume by calculating the matrix dot product of two-point features. (3) In each iteration, the differentiable neural-optimization iteration operator uses the pose estimated in the previous iteration to look up correlation features from the correlation volume, which are then processed through a GRU (Cho 2014) to generate a point motion field. The point motion field is fed into a least-squares-based optimization layer, where the pose is updated based on geometric constraints. After multiple iterations, the network outputs the predicted pose.

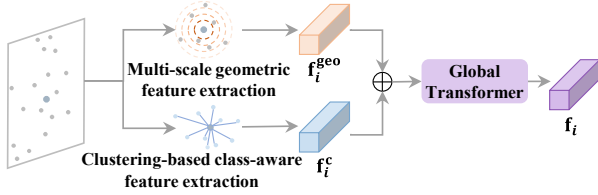


Figure 2: The structure of dual-stream radar feature extraction network.

tion, and a global Transformer module, as illustrated in Figure 2. This network offers a more comprehensive feature representation for each 4D radar point, thereby significantly improving the robustness of odometry across diverse environments.

Multi-scale Geometric Feature Extraction Given the 4D radar point clouds $\mathbf{P} = \{\mathbf{p}_i | \mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$, we employ four parallel Set Abstraction (SA) layers (Qi et al. 2017b) to encode the multi-scale geometric features. Each SA layer uses a different grouping radius to address the issue of non-uniform density in radar point clouds. At each scale s , each radar point \mathbf{p}_i generates a local geometric feature $\mathbf{f}_{i,s}^{\text{geo}}$. The geometric features across all scales are concatenated to form the multi-scale geometric feature representation of the 4D radar point, $\mathbf{f}_i^{\text{geo}} = \text{concat}(\{\mathbf{f}_{i,s}^{\text{geo}}\}_{s=0}^3)$.

Clustering-based Class-aware Feature Extraction To address the inherently non-uniform spatial distribution of 4D radar point clouds, we propose a novel clustering-based class-aware 4D radar point feature extraction method, consisting primarily of three components: 4D radar point clustering, class-aware feature aggregation, and feature dispatching.

4D Radar Point Clustering: We first use PointNet (Qi et al. 2017a) to map the 4D radar points to a high-dimensional feature space \mathbf{f}_i^p for similarity computation. Next, we apply Farthest Point Sampling (FPS) (Qi et al. 2017b) to evenly select C center points in space, and the center feature \mathbf{f}^{ct} is computed by averaging its K nearest points. We then calculate the pairwise cosine similarity matrix $\mathbf{S} \in \mathbb{R}^{C \times N}$ between each point and the set of center points. After that, we allocate each point to the most similar center, resulting in C clusters.

Class-aware Feature Aggregation: Following (Ma et al. 2023), we dynamically aggregate all radar point features within the same cluster based on the similarities to the cluster center. Given the cluster which contains M radar points around the j -th cluster center, the aggregated feature \mathbf{f}_j^a is calculated by:

$$\mathbf{f}_j^a = \frac{1}{\mathbf{A}} \left(\mathbf{f}_j^{\text{ct}} + \sum_{m=1}^M \text{sigmoid}(\alpha s_{jm} + \beta) \cdot \mathbf{f}_m^p \right), \quad (1)$$

$$\mathbf{A} = 1 + \sum_{m=1}^M \text{sigmoid}(\alpha s_{jm} + \beta), \quad (2)$$

where \mathbf{f}_j^{ct} represents the feature of the j -th cluster center, and s_{jm} denotes the similarity score between the m -th radar point and the j -th cluster center. α and β are learnable scalars used to scale and shift the similarity, while $\text{sigmoid}(\cdot)$ is the sigmoid function that rescales the similarity to (0,1). \mathbf{A} is the normalization factor.

Feature Dispatching: After obtaining the class-aware aggregated features, we employ an adaptive process to assign each point within a cluster based on its similarity, resulting in a more coherent and representative understanding of the overall structure and context within the cluster. For each point embedding \mathbf{f}_m^p , the updated point embedding \mathbf{f}_m^c is

computed using the following formula:

$$\mathbf{f}_m^c = \mathbf{f}_m^p + \text{sigmoid}(\alpha s_{jm} + \beta) \cdot \mathbf{f}_j^a, \quad (3)$$

We concatenate the multi-scale geometric features of radar points with the class-aware features to construct the geometric-class joint feature representation \mathbf{f}_i^1 .

Global Transformer The global transformer employs an attention mechanism (Vaswani et al. 2017) across the entire radar point cloud to capture the long-range correlation of points. For each point $\mathbf{p}_i \in \mathbf{P}$, the global transformer applies an attention mechanism to all other points $\mathbf{p}_j \in \mathbf{P}$:

$$\mathbf{f}_i = \sum_{\mathbf{f}_j^1 \in \mathcal{X}^1} \langle \alpha^g(\mathbf{f}_i^1), \beta^g(\mathbf{f}_j^1) \rangle \gamma^l(\mathbf{f}_j^1), \quad (4)$$

where \mathcal{X}^1 represents the geometric-class joint feature set, $\alpha^g(\cdot)$, $\beta^g(\cdot)$, and $\gamma^g(\cdot)$ are shared learnable linear transformations. $\langle \cdot \rangle$ denotes a scalar product. Linear layer, layer norm, and skip connection are further applied to complete the global transformer module.

Feature Correlation

We construct a feature correlation volume by evaluating feature similarity between all point pairs. Given point features $\mathbf{F}^1 = \{\mathbf{f}_i^1 | \mathbf{f}_i^1 \in \mathbb{R}^D\}_{i=1}^N$ and $\mathbf{F}^2 = \{\mathbf{f}_j^2 | \mathbf{f}_j^2 \in \mathbb{R}^D\}_{j=1}^N$, where D is the feature dimension, the correlation volume $\mathbf{V}^{12} \in \mathbb{R}^{N \times N}$ can be computed through matrix dot product:

$$\mathbf{V}^{12} = \mathbf{F}^1 \cdot (\mathbf{F}^2)^T. \quad (5)$$

Differentiable Neural-Optimization Iteration Operator

The core of our approach is the differentiable neural-optimization iteration operator, which contains three key parts, as illustrated in Figure. 3: (1) an adaptive patch-to-patch correlation lookup method that retrieves correlation features from the precomputed correlation volume; (2) a GRU (Cho 2014) that predicts updates to the point motion flow along with associated confidence weights; and (3) an All-point Motion-only Bundle Adjustment layer (AMBA) computes pose that aligns with the newly predicted point motion flow and confidence.

Adaptive Patch-to-Patch Correlation Look-up Layer

At the start of each iteration, we use the pose estimates from the previous iteration to calculate the correspondences. Specifically, for the i -th iteration, given the point clouds \mathbf{P}^1 and \mathbf{P}^2 , along with their respective poses \mathbf{T}_i^1 and \mathbf{T}_i^2 , we compute the all-point correspondence field $\mathbf{P}_i^{12} \in \mathbb{R}^{N \times 3}$:

$$\mathbf{P}_i^{12} = \mathbf{T}_i^{12} \cdot \mathbf{P}^1, \quad \mathbf{T}_i^{12} = \mathbf{T}_i^2 \cdot (\mathbf{T}_i^1)^{-1}, \quad (6)$$

\mathbf{P}_i^{12} represents the coordinates of points \mathbf{P}^1 mapped into \mathbf{P}^2 using the estimated pose. Then, we incorporate the Patch-to-Patch correlation concept from HALFlow (Wang et al. 2021a) into the point correlation look-up mechanism of PV-RAFT (Wei et al. 2021) to learn fine-grained correlation features of sparse 4D radar points, thereby guiding the right flowing direction of point motion. Specifically, for each

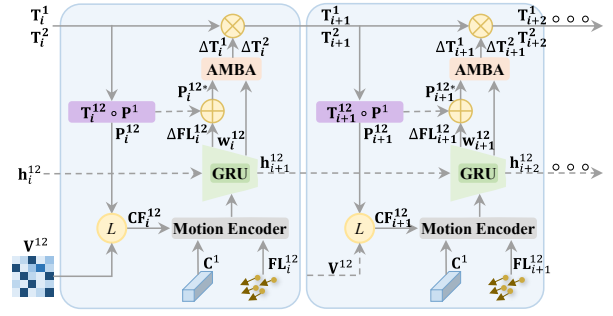


Figure 3: Illustration of the differentiable neural-optimization iteration operator, which predicts point motion flow revisions and maps them to pose updates through the AMBA layer.

point $\mathbf{p}_{i,j}^{12}$ in \mathbf{P}_i^{12} , we first retrieve the initial correlation features from the correlation volume using nearest neighbor queries. Subsequently, based on the 3D local geometric structure of $\mathbf{p}_{i,j}^{12}$ and its neighboring points in \mathbf{P}_i^{12} , we adaptively weight the initial correlation features of these neighbors to obtain the final correlation feature $\mathbf{cf}_{i,j}^{12}$.

Point Motion Flow Update and Confidence Weight Prediction We encode motion features using the correlation features $\mathbf{CF}_i^{12} = \{\mathbf{cf}_{i,j}^{12} | \mathbf{cf}_{i,j}^{12} \in \mathbb{R}^D\}_{j=1}^N$, the context features \mathbf{C}^1 , and the flow estimate $\mathbf{FL}_i^{12} \in \mathbb{R}^{N \times 3}$ from the previous iteration:

$$\mathbf{MF}_i^{12} = \text{concat}(\mathbf{CF}_i^{12}, \mathbf{C}^1, \text{MLP}(\mathbf{FL}_i^{12})). \quad (7)$$

The motion features \mathbf{MF}_i^{12} is used as the input to the GRU (Cho 2014) to update the current hidden state \mathbf{h}_i^{12} . Instead of predicting updates to the pose directly, we predict updates in the space of point motion flow fields. We map the updated hidden state through two additional MLP layers to produce two outputs: (1) a revision flow field $\Delta \mathbf{FL}_i^{12} \in \mathbb{R}^{N \times 3}$ and (2) associated confidence $\mathbf{w}_i^{12} \in \mathbb{R}_+^{N \times 3}$. The revision $\Delta \mathbf{FL}_i^{12}$ is a correction term predicted by the network to correct errors in the all-point correspondence field. We denote the corrected correspondence as $\mathbf{P}_i^{12*} = \mathbf{P}_i^{12} + \Delta \mathbf{FL}_i^{12}$. The confidence term is further processed with a sigmoid operation to constrain its values within the (0, 1). This term mitigates the adverse impact of inaccurate point motion flow estimation caused by occlusion, moving objects and noisy points on pose estimation accuracy.

All-point Motion-only Bundle Adjustment Layer The All-point Motion-only Bundle Adjustment Layer (AMBA) maps flow revisions into pose updates. The primary objective of the AMBA layer is to minimize the discrepancy between the transformed point cloud and the corrected correspondence \mathbf{P}_i^{12*} . Specifically, we define the cost function as:

$$E(\mathbf{T}^1, \mathbf{T}^2) = \|\mathbf{P}_i^{12*} - \mathbf{T}^2 \cdot (\mathbf{T}^1)^{-1} \cdot \mathbf{P}^1\|_{\Sigma_{12}}^2, \quad (8)$$

where $\|\cdot\|_{\Sigma}^2$ is the Mahalanobis distance which weights the error terms based on the confidence weights \mathbf{w}_i^{12} . In this

paper, we optimize the pose using the Gauss-Newton algorithm, based on the special Euclidean Lie algebra $se(3) = \left\{ \xi = \begin{pmatrix} \rho \\ \phi \end{pmatrix}, \rho \in \mathbb{R}^3, \phi \in so(3) \right\}$. First, we compute the Jacobian matrices $\mathbf{J}^1 \in \mathbb{R}^{N \times 3 \times 6}$ and $\mathbf{J}^2 \in \mathbb{R}^{N \times 3 \times 6}$ of the cost function with respect to \mathbf{T}^1 and \mathbf{T}^2 using the perturbation model.

$$\mathbf{J}^1 = \begin{bmatrix} \mathbf{I} & -(\mathbf{R}^{12} \circ \mathbf{P}^1 + \mathbf{t}^{12})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \circ \text{Adj}_{\mathbf{T}^{12}}, \quad (9)$$

$$\mathbf{J}^2 = - \begin{bmatrix} \mathbf{I} & -(\mathbf{R}^{12} \circ \mathbf{P}^1 + \mathbf{t}^{12})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}, \quad (10)$$

where $(\cdot)^\wedge$ denotes the antisymmetric matrix operation and $\text{Adj}_{(\cdot)}$ denotes the adjoint operator.

Then, we linearize the cost function, and solve for the update:

$$\mathbf{J}^T \text{diag}(\mathbf{w}_i^{12}) \mathbf{J} \Delta \xi = -\mathbf{J}^T E(\mathbf{T}^1, \mathbf{T}^2), \quad (11)$$

where $\Delta \xi = (\Delta \xi^1, \Delta \xi^2)$ represents the pose update in the Lie algebra. Eqn.11 can be rewritten as a linear system:

$$\mathbf{H} \Delta \xi = \mathbf{b}, \quad \mathbf{H} = \mathbf{J}^T \text{diag}(\mathbf{w}_i^{12}) \mathbf{J}, \quad \mathbf{b} = -\mathbf{J}^T E(\mathbf{T}^1, \mathbf{T}^2). \quad (12)$$

This linear system is solved using Cholesky decomposition to obtain the updates $\Delta \xi^1$ and $\Delta \xi^2$. The updates refine the pose estimates from the previous iteration, and the updated poses serve as input for the next neural-optimization iteration module:

$$\mathbf{T}_{i+1}^1 = \text{Exp}(\Delta \xi^1) \cdot \mathbf{T}_i^1, \quad \mathbf{T}_{i+1}^2 = \text{Exp}(\Delta \xi^2) \cdot \mathbf{T}_i^2. \quad (13)$$

Based on the properties of the Jacobian matrix and Cholesky decomposition, gradients are backpropagated using the chain rule. Consequently, the AMBA layer functions as a differentiable optimization layer, enabling end-to-end training of the entire architecture during the training process.

Training and Implementation

DNOI-4DRO is implemented in PyTorch, with the LieTorch extension (Teed and Deng 2021c) used to perform backpropagation in the tangent space of all group elements.

Constructing Frame Graph

Each training example consists of a 7-frame 4D radar point cloud sequence. For each point cloud frame in the training samples, edges are constructed with its three nearest neighboring frames to form a frame graph. In this graph, any two frames connected by an edge undergo pose estimation using the backbone network. Additionally, to remove the 6-dof gauge freedom, the first pose in each training sample is fixed to the ground truth.

Supervision

We supervise our network using a pose loss. Given a set of ground truth poses $\{\tilde{\mathbf{T}}\}_{i=1}^L$ and predicted poses $\{\mathbf{T}\}_{i=1}^L$, the pose loss is taken to be the distance between the ground truth and predicted poses:

$$L_{pose} = \sum_{i=1}^L \left\| \text{Log}_{SE3} \left((\tilde{\mathbf{T}}^i)^{-1} \cdot \mathbf{T}^i \right) \right\|^2, \quad (14)$$

where $\text{Log}_{SE3}(\cdot)$ represents the logarithmic map from Lie group to Lie algebra.

Training Details

We train for 30 epochs on an A100 GPU with a batch size of 4. We use the Adam optimizer and start with an initial learning rate of $2e^{-4}$, which decays by a factor of 0.1 every 10 epochs. N is set to be 512 in the proposed network. The radar point heights are constrained within the range of $[-2m, 10m]$ to retain reliable points. During training, random transformation matrices are applied to the points and the ground truth pose to increase data diversity. We select 8 cluster centers. The iteration operator is unrolled for 15 iterations during training, with two bundle adjustment steps performed within each iteration. The ground truth pose of the first frame in each training sequence serves as the initial pose for all radar frames in that sequence, with the initial point motion flow set to zero.

4DRO System

We apply the backbone network during inference to construct a fully 4DRO system that processes a 4D radar point cloud stream for real-time localization.

Initialization: We initialize with 8 frames, continuously adding new frames until 8 frames are accumulated, after which we perform 12 iterations of the iteration operator.

Tracking: We maintain a frame graph. After initialization, when a new frame is added, we extract features and add the new frame to the frame graph, establishing edges with all frames within an index distance of 2. The pose is initialized using a linear motion model. We then run four iterations of the iteration operator, performing two bundle adjustment iterations within each iteration. We fix the initial frame's pose in the frame graph to remove gauge freedom. After the new frame is tracked, we remove the oldest frame.

Experiments

We evaluate the method on two 4D radar odometry benchmarks, with dataset details in the appendix. We follow protocols of DVLO (Liu et al. 2024) to evaluate our method with two metrics: (1) Average sequence translational RMSE (%). (2) Average sequence rotational RMSE ($^\circ/100m$).

Performance Evaluation

VoD Results: Quantitative results are listed in Table 1. ICP-point2point (ICP-po2po) (Besl and McKay 1992), ICP-point2plane (ICP-po2pl) (Chen and Medioni 1992), GICP (Segal, Haehnel, and Thrun 2009), NDT (Stoyanov et al. 2012) are several classical point cloud odometry methods. TransLO (Liu et al. 2023) and EfficientLO (Wang et al. 2022) rely on the imaging principle of LiDAR to project point clouds onto a 2D cylindrical surface as network input. However, 4D radar operates with a different imaging mechanism and produces only about 512 sparse points per frame, making projection-based approaches unsuitable for such data. Therefore, we primarily compare our method with leading open-source LiDAR odometry techniques that directly process raw 3D point clouds, such as PWCLO-Net

Method		03		04		09		17		19		22		24		Mean	
		t_{rel}	r_{rel}	t_{rel}	r_{rel}	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 methods	ICP-po2po	0.39	1.00	0.21	1.14	0.15	0.72	0.16	0.53	1.40	4.70	0.44	0.76	0.24	0.77	0.427	1.374
	ICP-po2pl	0.42	2.19	0.37	1.83	0.50	1.32	0.23	0.68	3.04	5.62	0.42	1.20	0.35	0.67	0.761	1.930
	GICP	0.46	0.68	0.30	0.39	0.51	0.32	0.40	0.10	0.51	1.23	0.34	0.57	0.15	0.30	0.381	0.513
	NDT	0.55	1.60	0.47	0.91	0.46	0.56	0.44	0.40	1.33	2.58	0.47	1.10	0.36	1.84	0.583	1.284
LiDAR-based methods	Full A-LOAM	NA	NA	0.03	0.09	0.04	0.19	0.02	0.04	0.38	1.35	0.06	0.18	0.06	0.20	0.098	0.342
	A-LOAM w/o mapping	NA	NA	0.14	0.35	0.16	1.23	0.09	0.26	1.17	4.63	0.27	0.92	0.16	0.81	0.332	1.370
	PWCLO-Net	0.26	0.37	0.31	0.40	0.38	0.55	0.27	0.39	1.23	0.91	0.23	0.35	0.46	0.82	0.449	0.541
4D Radar-based methods	RaFlow	0.87	2.09	0.07	0.44	0.11	0.09	0.13	0.03	1.22	4.09	0.72	1.34	0.25	1.14	0.481	1.317
	CMFlow	0.06	0.10	0.05	0.09	0.09	0.14	0.06	0.03	0.28	0.94	0.14	0.29	0.12	0.58	0.114	0.310
	4DRONet	0.08	0.10	0.04	0.07	0.13	0.38	0.09	0.10	0.91	0.62	0.23	0.32	0.28	1.20	0.251	0.398
	CAO-RONet	0.049	0.034	0.042	0.040	0.064	0.089	0.102	0.013	0.024	0.045	0.082	0.063	0.135	0.076	0.071	0.051
Ours		0.021	0.018	0.022	0.026	0.017	0.016	0.018	0.021	0.021	0.038	0.025	0.024	0.034	0.048	0.023	0.027

Table 1: The 4D radar odometry experiment results on the VoD dataset (Palffy et al. 2022). t_{rel} and r_{rel} denote the average translational RMSE (m/m) and rotational RMSE ($^{\circ}$ /m), respectively, on all possible subsequences in the length of 20, 40, ..., 160m. All methods listed use 4D radar point clouds as input. Full A-LOAM is a complete SLAM system and others only include odometry. The best results are bold.

Method		Input	03		04		09		17		19		22		24		Mean	
			t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Full A-LOAM	L	0.05	0.11	0.03	0.10	0.03	0.08	0.02	0.02	0.05	0.14	0.03	0.09	0.03	0.05	0.034	0.084	
A-LOAM w/o mapping	L	0.06	0.10	0.03	0.12	0.06	0.06	0.03	0.02	0.14	0.19	0.06	0.11	0.06	0.08	0.063	0.097	
4DRONet	C+R	0.02	0.02	0.01	0.02	0.03	0.05	0.11	0.03	0.26	0.08	0.06	0.09	0.09	0.18	0.083	0.067	
Ours	R	<u>0.021</u>	0.018	<u>0.022</u>	<u>0.026</u>	0.017	0.016	0.018	0.021	0.021	0.038	0.025	0.024	0.034	0.048	0.023	0.027	

Table 2: The experiment results on the VoD dataset (Palffy et al. 2022) using methods with LiDAR or 4D radar-camera inputs. ‘L’, ‘C’, and ‘R’ represent LiDAR, camera, and 4D radar, respectively. The best results are bold.

(Wang et al. 2021b). A-LOAM (Zhang and Singh 2017) is a conventional LiDAR odometry method that achieves state-of-the-art performance on the KITTI Odometry benchmark (Geiger et al. 2013). We use 4D radar point clouds as input for these methods. The experimental results indicate that, while these methods perform effectively on LiDAR point clouds, their performance deteriorates significantly on 4D radar point clouds due to the sparse, noisy, and non-panoramic characteristics of 4D radar data. RaFlow (Ding et al. 2022) and CMFlow (Ding et al. 2023) are 4D radar-based scene flow estimation networks incorporating odometry estimation as an intermediate task. 4DRONet (Lu et al. 2024) and CAO-RONet (Li et al. 2025) are odometry networks specifically tailored for processing 4D radar data. Our method outperforms all previous 4D radar odometry methods due to its innovative architecture and robust point cloud feature extraction. Notably, our method reduces mean translation and rotation errors by 67.6% and 47.1%, respectively, compared to the previous best learning-based method.

Table 2 shows the evaluation results of Full A-LOAM and A-LOAM without mapping using 64-beam LiDAR point clouds as input, along with the results of the 4D radar-camera fusion odometry method (4DRVONet) (Zhuo et al. 2024). It can be seen that our method achieves competitive performance with Full A-LOAM in short-range localization. Compared with 4DRVONet, our method achieves a reduction of 72.3% in mean translation error and 59.7% in mean rotation error on the test sequences. The experiment results prove the effectiveness of our proposed neural-optimization iteration module and demonstrate its great potential in 4D radar odometry tasks.

Snail-Radar Results: Compared to VoD (Palffy et al.

Method		Input	if		iaf		st		Mean	
			t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Full A-LOAM	L	0.92	0.41	1.54	0.54	1.93	2.54	1.47	1.16	
	L	<u>14.74</u>	<u>4.10</u>	<u>14.75</u>	<u>3.88</u>	<u>12.26</u>	<u>9.62</u>	<u>13.92</u>	<u>5.87</u>	
ICP-po2po	R	35.54	13.98	37.09	10.18	26.31	60.07	32.98	28.08	
	R	27.16	11.17	30.81	11.35	20.70	39.96	26.23	20.83	
	R	31.20	11.37	31.79	9.7	26.36	51.22	29.79	24.10	
4DRadarSLAM	R	<u>13.32</u>	<u>6.11</u>	<u>16.13</u>	<u>6.62</u>	<u>10.70</u>	<u>15.43</u>	<u>13.39</u>	<u>9.39</u>	
Ours	R	<u>3.92</u>	<u>1.85</u>	<u>6.10</u>	<u>2.21</u>	<u>3.76</u>	<u>3.94</u>	<u>4.60</u>	<u>2.67</u>	

Table 3: The experiment results on the Snail-Radar dataset (Huai et al. 2024). t_{rel} and r_{rel} denote the average translational RMSE (%) and rotational RMSE ($^{\circ}$ /100m), respectively, on all possible subsequences in the length of 100, 200, ..., 800m. The best results are shown in bold, and the second-best results are underlined.

2022), the Snail-Radar Dataset (Huai et al. 2024) presents greater challenges with longer, complex trajectories. We benchmark against ICP methods, 4DRadarSLAM (Zhang et al. 2023) (loop closure disabled), and LiDAR-based A-LOAM. Table 3 shows our method outperforms A-LOAM (w/o mapping) and narrows the gap with Full A-LOAM. Notably, it reduces translation and rotation errors by 65.7% and 71.6% compared to 4DRadarSLAM.

Ablation Study

We perform ablation experiments on the VoD Dataset to evaluate the effectiveness of each module in DNOI-4DRO.

Feature Extraction: We evaluate the effectiveness of each component in the proposed feature extraction network. As shown in Table 4(a), removing any individual module leads

Method	03		04		09		17		19		22		24		Mean	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
(a) Ours w/o multi-scale geometric feature	0.040	0.067	0.026	0.033	0.030	0.012	0.017	0.023	0.033	0.150	0.030	0.049	0.033	0.033	0.030	0.052
Ours w/o clustering-based class-aware feature	0.043	0.072	0.025	0.030	0.019	0.012	0.017	0.021	0.050	0.053	0.023	0.038	0.031	0.030	0.030	0.037
Ours w/o global transformer	0.032	0.037	0.027	0.031	0.016	0.014	0.018	0.021	0.045	0.071	0.027	0.039	0.027	0.026	0.027	0.034
Ours (full)	0.035	0.056	0.021	0.031	0.016	0.018	0.015	0.015	0.039	0.021	0.016	0.028	0.027	0.034	0.024	0.029
(b) Ours w/o AMBA layer	0.053	0.115	0.024	0.085	0.020	0.053	0.020	0.048	0.021	0.159	0.037	0.076	0.033	0.089	0.030	0.089
Ours w/o iteration operator	0.470	0.814	0.338	0.701	0.455	0.704	0.390	0.442	0.133	0.021	0.389	0.600	0.259	0.873	0.348	0.594
Ours (full)	0.035	0.056	0.021	0.031	0.016	0.018	0.015	0.015	0.039	0.021	0.016	0.028	0.027	0.034	0.024	0.029
(c) Ours w/o confidence weight	0.039	0.041	0.027	0.040	0.033	0.027	0.032	0.025	0.189	0.259	0.048	0.043	0.049	0.109	0.060	0.078
Ours (one confidence weight)	0.056	0.092	0.031	0.039	0.032	0.049	0.097	0.015	0.065	0.075	0.024	0.037	0.037	0.041	0.049	0.050
Ours (full)	0.035	0.056	0.021	0.031	0.016	0.018	0.015	0.015	0.039	0.021	0.016	0.028	0.027	0.034	0.024	0.029

Table 4: The ablation study results of 4D Radar odometry on the VoD dataset (Palffy et al. 2022). The best results are bold.

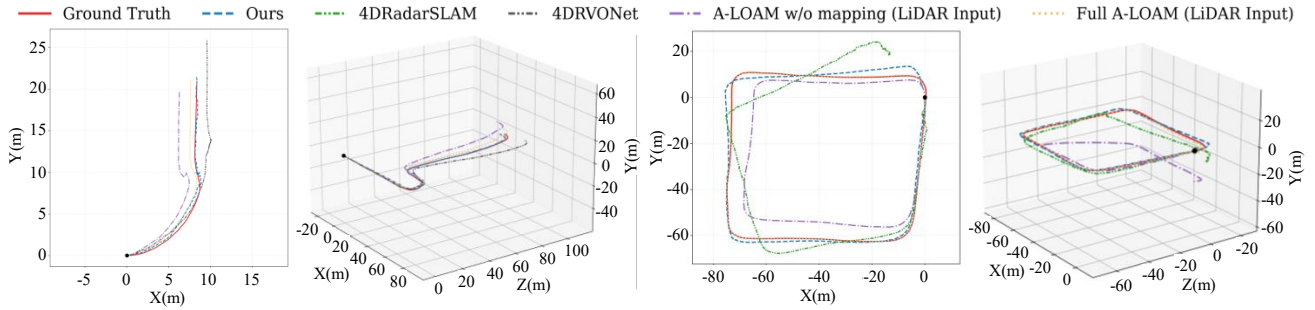


Figure 4: 3D and 2D trajectory results for VoD test sequences 19 and 22, and Snail-Radar test sequences st. Our method obtains the most accurate trajectory.

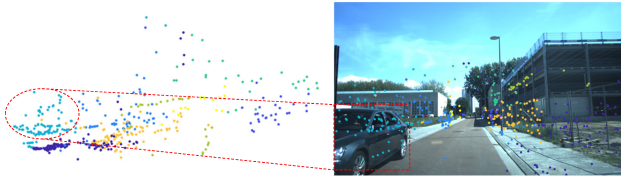


Figure 5: Visualization of clustering results. 4D radar points are color-coded based on their assigned categories.

to a noticeable drop, highlighting each module’s critical role in capturing fine-grained 4D radar features and enhancing the overall 4DRO performance.

Iteration Operator: We remove the AMBA layer during training, relying on point motion flow supervision to train the network. Then, we eliminate the entire iteration operator and directly regress poses from correlation features, which means that completely discarding the geometric optimization process. As shown in Table 4(b), removing either component significantly reduces pose estimation accuracy.

Confidence Weight: First, we remove the confidence weight by setting all weights to 1. Then, we calculate a confidence value for each point, assuming that the point has equal importance in pose estimation across different directions. Table 4(c) show that the proposed confidence weight and fine-grained estimation across different directions contribute to better results.

Visualization

We visualize 3D and 2D trajectories based on our estimated pose on the VoD and Snail-Radar datasets, as shown in Figure 4. These figures show that our odometry can track the trajectory of the ground truth fairly well. Although we do not have the mapping procedure, our odometry achieves trajectory accuracy comparable to that of Full A-LOAM in short-range localization. Furthermore, our approach significantly reduces odometry drift along the Z-axis compared to other methods. Figure 5 illustrates the clustering results. The clustering-based method effectively mitigates the constraints of the receptive field, facilitating extracting features that belong to the same category (e.g., vehicles, curbs, walls). This approach enables point features to exhibit class awareness, improving their representational capacity.

Conclusion

We introduce DNOI-4DRO, an end-to-end neural architecture for 4D radar odometry, which combines the strengths of both classical approaches and deep networks through a differentiable neural-optimization iteration module. We compare against existing approaches and show strong performance across several datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 52325212.

References

- Ali, S. A.; Aouada, D.; Reis, G.; and Stricker, D. 2023. Delo: deep evidential lidar odometry using partial optimal transport. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4517–4526.
- Besl, P. J.; and McKay, N. D. 1992. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, 586–606. Spie.
- Chen, W.; Chen, L.; Wang, R.; and Pollefeys, M. 2024. Leap-vo: Long-term effective any point tracking for visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19844–19853.
- Chen, Y.; and Medioni, G. 1992. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3): 145–155.
- Cho, K. 2014. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. *arXiv preprint arXiv:1409.1259*.
- Deng, J.; Wu, Q.; Chen, X.; Xia, S.; Sun, Z.; Liu, G.; Yu, W.; and Pei, L. 2023. Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8218–8227.
- Ding, F.; Luo, Z.; Zhao, P.; and Lu, C. X. 2024a. milliflow: Scene flow estimation on mmwave radar point cloud for human motion sensing. In *European Conference on Computer Vision*, 202–221. Springer.
- Ding, F.; Palffy, A.; Gavrilu, D. M.; and Lu, C. X. 2023. Hidden gems: 4d radar scene flow learning using cross-modal supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9340–9349.
- Ding, F.; Pan, Z.; Deng, Y.; Deng, J.; and Lu, C. X. 2022. Self-supervised scene flow estimation with 4-d automotive radar. *IEEE Robotics and Automation Letters*, 7(3): 8233–8240.
- Ding, F.; Wen, X.; Zhu, Y.; Li, Y.; and Lu, C. X. 2024b. Radarocc: Robust 3d occupancy prediction with 4d imaging radar. *Advances in Neural Information Processing Systems*, 37: 101589–101617.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11): 1231–1237.
- Guan, R.; Zhang, R.; Ouyang, N.; Liu, J.; Man, K. L.; Cai, X.; Xu, M.; Smith, J.; Lim, E. G.; Yue, Y.; et al. 2024. Talk2Radar: Bridging natural language with 4d mmwave radar for 3d referring expression comprehension. *arXiv preprint arXiv:2405.12821*.
- Huai, J.; Wang, B.; Zhuang, Y.; Chen, Y.; Li, Q.; and Han, Y. 2024. SNAIL Radar: A large-scale diverse benchmark for evaluating 4D-radar-based SLAM. *arXiv preprint arXiv:2407.11705*.
- Li, Q.; Chen, S.; Wang, C.; Li, X.; Wen, C.; Cheng, M.; and Li, J. 2019. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8473–8482.
- Li, X.; Zhang, H.; and Chen, W. 2023. 4d radar-based pose graph slam with ego-velocity pre-integration factor. *IEEE Robotics and Automation Letters*, 8(8): 5124–5131.
- Li, Z.; Cui, Y.; Huang, N.; Pang, C.; and Fang, Z. 2025. CAO-RONet: A Robust 4D Radar Odometry with Exploring More Information from Low-Quality Points. *arXiv preprint arXiv:2503.01438*.
- Lin, Z.; Liu, Z.; Xia, Z.; Wang, X.; Wang, Y.; Qi, S.; Dong, Y.; Dong, N.; Zhang, L.; and Zhu, C. 2024. RCBEVDet: radar-camera fusion in bird’s eye view for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14928–14937.
- Lipson, L.; and Deng, J. 2024. Multi-session slam with differentiable wide-baseline pose optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19626–19635.
- Liu, J.; Wang, G.; Jiang, C.; Liu, Z.; and Wang, H. 2023. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1683–1691.
- Liu, J.; Zhuo, D.; Feng, Z.; Zhu, S.; Peng, C.; Liu, Z.; and Wang, H. 2024. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment. In *European Conference on Computer Vision*, 475–493. Springer.
- Lu, S.; Zhuo, G.; Xiong, L.; Zhu, X.; Zheng, L.; He, Z.; Zhou, M.; Lu, X.; and Bai, J. 2024. Efficient Deep-Learning 4D Automotive Radar Odometry Method. *IEEE Transactions on Intelligent Vehicles*, 9(1): 879–892.
- Ma, X.; Zhou, Y.; Wang, H.; Qin, C.; Sun, B.; Liu, C.; and Fu, Y. 2023. Image as set of points. *arXiv preprint arXiv:2303.01494*.
- Palffy, A.; Pool, E.; Baratam, S.; Kooij, J. F.; and Gavrilu, D. M. 2022. Multi-class road user detection with 3+ 1D radar in the View-of-Delft dataset. *IEEE Robotics and Automation Letters*, 7(2): 4961–4968.
- Pan, Z.; Ding, F.; Zhong, H.; and Lu, C. X. 2024. Ratrack: moving object detection and tracking with 4d radar point cloud. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 4480–4487. IEEE.
- Peng, G.; Li, H.; Zhao, Y.; Zhang, J.; Wu, Z.; Zheng, P.; and Wang, D. 2024. Transloc4d: Transformer-based 4d radar place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17595–17605.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.
- Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-icp. In *Robotics: science and systems*, volume 2, 435. Seattle, WA.

- Stoyanov, T.; Magnusson, M.; Andreasson, H.; and Lilienthal, A. J. 2012. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *The International Journal of Robotics Research*, 31(12): 1377–1393.
- Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419. Springer.
- Teed, Z.; and Deng, J. 2021a. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34: 16558–16569.
- Teed, Z.; and Deng, J. 2021b. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8375–8384.
- Teed, Z.; and Deng, J. 2021c. Tangent space backpropagation for 3d transformation groups. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10338–10347.
- Teed, Z.; Lipson, L.; and Deng, J. 2023. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36: 39033–39051.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, D.; May, S.; and Nuechter, A. 2024. RIV-SLAM: Radar-Inertial-Velocity optimization based graph SLAM. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 774–781. IEEE.
- Wang, G.; Wu, X.; Jiang, S.; Liu, Z.; and Wang, H. 2022. Efficient 3d deep lidar odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5749–5765.
- Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021a. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30: 5168–5181.
- Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021b. Pwclonet: 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*, 15910–15919.
- Wang, S.; Li, W.; Wang, Y.; Fan, Z.; Huang, Z.; Cai, X.; Zhao, J.; and Li, D. 2024. MambaVO: Deep Visual Odometry Based on Sequential Matching Refinement and Training Smoothing. *arXiv preprint arXiv:2412.20082*.
- Wei, Y.; Wang, Z.; Rao, Y.; Lu, J.; and Zhou, J. 2021. Pvraft: Point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6954–6963.
- Wu, X.; Chen, Y.; Li, Z.; Hong, Z.; and Hu, L. 2024. EFEAR-4D: Ego-Velocity Filtering for Efficient and Accurate 4D radar Odometry. *IEEE Robotics and Automation Letters*.
- Zhang, H.; Wang, G.; Wu, X.; Xu, C.; Ding, M.; Tomizuka, M.; Zhan, W.; and Wang, H. 2024. DSLO: Deep Sequence LiDAR Odometry Based on Inconsistent Spatio-temporal Propagation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10672–10677. IEEE.
- Zhang, J.; and Singh, S. 2017. Low-drift and real-time lidar odometry and mapping. *Autonomous robots*, 41: 401–416.
- Zhang, J.; Zhuge, H.; Wu, Z.; Peng, G.; Wen, M.; Liu, Y.; and Wang, D. 2023. 4dradar slam: 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)*, 8333–8340. IEEE.
- Zhou, H.; Lu, S.; and Zhuo, G. 2023. Self-supervised 4-D Radar Odometry for Autonomous Vehicles. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 764–769.
- Zhuang, Y.; Wang, B.; Huai, J.; and Li, M. 2023. 4d iriom: 4d imaging radar inertial odometry and mapping. *IEEE Robotics and Automation Letters*, 8(6): 3246–3253.
- Zhuo, G.; Lu, S.; Zhou, H.; Zheng, L.; Zhou, M.; and Xiong, L. 2024. 4DRVO-Net: Deep 4D Radar–Visual Odometry Using Multi-Modal and Multi-Scale Adaptive Fusion. *IEEE Transactions on Intelligent Vehicles*, 9(6): 5065–5079.