

# Learning Point Correspondences In Radar 3D Point Clouds For Radar-Inertial Odometry

Jan Michalczyk<sup>1</sup>, Stephan Weiss<sup>1</sup>, and Jan Steinbrener<sup>1</sup>

**Abstract**—Using 3D point clouds in odometry estimation in robotics often requires finding a set of correspondences between points in subsequent scans. While there are established methods for point clouds of sufficient quality, state-of-the-art still struggles when this quality drops. Thus, this paper presents a novel learning-based framework for predicting robust point correspondences between pairs of *noisy, sparse and unstructured 3D point clouds* from a light-weight, low-power, inexpensive, consumer-grade System-on-Chip (SoC) Frequency Modulated Continuous Wave (FMCW) radar sensor. Our network is based on the transformer architecture which allows leveraging the attention mechanism to discover pairs of points in consecutive scans with the greatest mutual affinity. The proposed network is trained in a self-supervised way using set-based multi-label classification cross-entropy loss, where the ground-truth set of matches is found by solving the Linear Sum Assignment (LSA) optimization problem, which avoids tedious hand annotation of the training data. Additionally, posing the loss calculation as multi-label classification permits supervising on point correspondences directly instead of on odometry error, which is not feasible for sparse and noisy data from the SoC radar we use. We evaluate our method with an open-source state-of-the-art Radar-Inertial Odometry (RIO) framework in real-world Unmanned Aerial Vehicle (UAV) flights and with the widely used public Coloradar dataset. Evaluation shows that the proposed method improves the position estimation accuracy by over 14 % and 19 % on average, respectively. The open source code and datasets can be found here: [https://github.com/aaucns/radar\\_transformer](https://github.com/aaucns/radar_transformer).

## I. INTRODUCTION

Using radar sensors for robot localization has recently been gaining significant attention in robotics mostly owing to the advances achieved in the FMCW millimeter-wave (mmWave) radar technology and its widespread use in the automotive industry [1]. Using millimeter wavelengths makes the radar largely unaffected by air obscuration and extreme illumination, allowing it to operate in adverse conditions including snow, fog, dust or darkness [2]. Thus, combining an Inertial Measurement Unit (IMU) with a radar into RIO offers an accurate odometry estimator robust in environments where other sensors are prone to fail [3]. Nonetheless, the light-weight and low-power SoC variant of the mmWave FMCW radar which is of greatest interest [4] in applications involving small-sized robots with limited payload like wheeled robots or UAVs is known for its notoriously noisy and sparse measurements [5].

<sup>1</sup>All authors are with the Control of Networked Systems Group, University of Klagenfurt, Austria {firstname.lastname}@ieee.org. This research received funding from the Austrian Ministry of Climate Action and Energy (BMK) under the grant agreement 880057 (CARNIVAL).

Pre-print version, accepted June/2025 (IROS), DOI follows ASAP ©IEEE.

Approaches employing FMCW SoC radar data in RIO estimation could generally be divided into methods relying only on the Doppler velocity information from the current measurement [6], [7], [8], [9], [10], [11] and methods performing some form of 3D point matching using past and current measurements in addition to the instantaneous Doppler velocity information [12], [13], [14], [15], [16]. Methods using bulky and expensive scanning radars for Radar Odometry (RO) often rely on keypoints extraction and scan registration owing to the high-quality and density of the 2D 360° scans they produce [17], [18], [19]. In this work we are focused on the low-cost, lightweight and low-power FMCW SoC radar sensors as our primary application are small-sized UAVs.

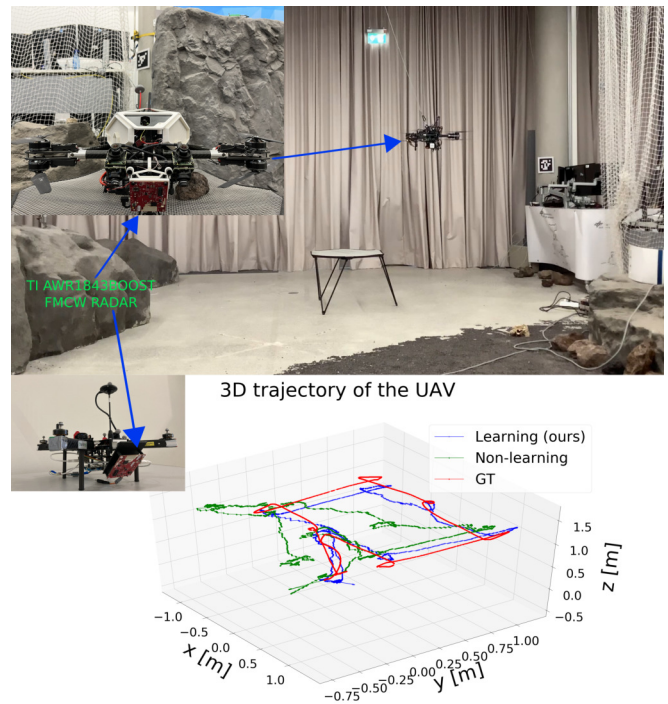


Fig. 1. ARDEA-X [20] and CNS-UAV platforms used in this work with the mounted consumer-grade FMCW SoC radar sensor. The radar chip that we use outputs highly noisy and sparse 4D point clouds (3D points and Doppler velocities). In the lower part of the figure we plot the estimated position for one of the validation flights using the EKF-based RIO framework from [12] (open-sourced with [21]) when switching between matching methods and using *solely* 3D point matches in the update step. Note how the proposed method allows tracking the position of the UAV making *no* use of the Doppler information, while the non-learning approach drifts considerably.

Using the 3D point information in state estimation usually involves finding correspondences between points in sub-

sequent point clouds. FMCW SoC radar point clouds are much different from their LiDAR counterparts in that they are orders of magnitude sparser which makes it impossible to sense structures in the environment [5]. Moreover, they exhibit temporal and viewpoint variations, that is, point clouds of the same object measured in different time instants as well as point clouds of the same object but measured from slightly different viewpoints may differ very significantly [22]. Additionally, they contain ghost reflections caused by speckle noise and multi-path reflections [23]. Hence, classic techniques developed for LiDAR data, like Iterative Closest Point (ICP) or Normal Distribution Transform (NDT) cannot readily be applied to radar point clouds [14]. Because of the challenging nature of the radar 3D point clouds there is a need for new methods to process them in state estimation pipelines.

In this work we present a novel learning framework for predicting robust point correspondences between FMCW SoC radar 3D point clouds in RIO estimation. Our framework is inspired by recent advances in deep learning for dense 3D point clouds processing in [24], [25] and tailored for the sparse and noisy radar measurements. In particular, our framework consists of the following steps (Fig. 2):

- 1) Calculate the input embeddings in a *per-point* manner for each of the two consecutive input point clouds using the *PointNet* architecture [26].
- 2) Use two transformer [27] sub-networks to predict a matrix whose rows and columns correspond to the points in the first and second input point clouds, respectively, and whose entries express the degree of likelihood that the corresponding row and column form a match.
- 3) Solve the LSA optimization problem on the predicted matrix to find the set of point correspondences, where each item contains the index into the first (row) and second (column) point cloud.
- 4) During training, cast the problem into a multi-label classification setting by considering the column index from the LSA solution the class label of a point in the second point cloud, which allows leveraging the cross-entropy loss function.
- 5) During inference, apply acceptance and Field Of View (FOV) thresholds to the set of matches found in step 3 to form the output.

We train and test our network on a real-world, self-collected dataset consisting of 13 manually and autonomously flown UAV trajectories from which we select 8 for training and 5 for testing. To make the validation more thorough and comparable, we also test our approach with the public Coloradar dataset [28]. Evaluation of our method in an open-source state-of-the-art EKF-based RIO framework from [21] (which does not use a learning-based matching algorithm to find 3D point matches) shows an increase in estimation accuracy by over 14% for the self-collected dataset and by 19% for the Coloradar dataset, in terms of position RMSE norm. We also note that, when deactivating the Doppler information and

only keeping the 3D point matches as correction information for the IMU integration in RIO, we note a difference in accuracy of more than 70% when using our method on the self-collected dataset. To our knowledge, this is the first framework for learning point correspondences in sparse and noisy 3D point clouds as available from inexpensive SoC radar sensors. Our main contributions are:

- Deep learning framework for predicting robust correspondences in sparse and noisy FMCW SoC radar 3D point clouds.
- Efficient, self-supervised method not requiring hand-annotated ground-truth data.
- Formulation of the learning problem as multi-label classification which allows training on the sparse and noisy 3D point clouds yet results in unambiguous matches.
- Evaluation of the method with real-world data (our own dataset and a public benchmark Coloradar dataset [28]) in a state-of-the-art open-source RIO estimation framework.
- Open-source implementation of the presented architecture together with the used dataset for the benefit of the research community.

This paper is organized as follows. Section II reviews the recent related work in the domain of finding point correspondences in radar point clouds. In section III, we describe our learning framework. Subsection III-A outlines the architecture of the presented network. In subsection III-B we describe how training and inference are performed. In section IV, we explain the experiments (subsection IV-A) and evaluations (subsection IV-B) conducted in order to demonstrate and validate the proposed method. Finally, we present conclusions in section V.

## II. RELATED WORK

Within the radar data association approaches employed in state estimation, we can distinguish methods suitable for the dense 2D 360° scans generated by mechanically rotating radar and 3D point clouds from SoC radar. Within the latter, we can also differentiate between methods using industry-grade and consumer-grade sensors. With our framework, we aim at inexpensive, consumer-grade radar sensors where the quality of the sensed point clouds is considerably lower in terms of number of points and noise than of those measured using industry-grade ones. Among the methods using the industry-grade SoC sensors, in [15] the authors integrate the Radar Cross Section (RCS) into the nearest-neighbor search for correspondences in the euclidean space to make it more resilient against the noise. In [29], the same authors augment the method from [15] by precisely modeling the uncertainty of radar measurements and incorporating it into the matching algorithm, which considerably improves the accuracy. Authors in [16] perform distribution-to-multi-distribution geometric scan-to-submap matching by introducing spatial covariances of clusters of points. In [13], the LSA problem is solved to construct a similarity matrix on which a search guided by a local geometric coherence is used to

match subsequent 3D point clouds from a consumer-grade SoC radar.

Approaches developed for scanning radars differ considerably from those for their SoC counterparts, due to the different nature of the measurements they collect. In [18], authors use a learning approach supervised on the odometry error to find salient keypoints in the 2D radar scans. Found keypoints are matched using the cosine similarities. In [17], a matching method is presented which leverages local coherence among points forming a scan, that is, an assumption that local geometries between points are preserved across consecutive scans. This assumption is also used in [13], [14]. The work in [19] presents an unsupervised learning approach to RO in which features from 2D radar scans are matched using a differentiable softmax matcher within their proposed network architecture.

Interestingly, authors in [30] argue that when industry-grade sensors are used and only odometry is needed, scan matching is no longer needed and the Doppler measurements suffice. This statement is disputed in [15], [29] where the authors claim the vital importance of the point matching in their RIO system despite using automotive-grade radar. At any rate, still in many systems the price, weight and power consumption requirements mandate the use of consumer-grade radar chips as the one we use in the present paper where using point matches boosts estimation accuracy.

### III. LEARNING 3D POINT CORRESPONDENCES IN RADAR 3D POINT CLOUDS

We base our network architecture on the one defined in [25] for registration of dense and noiseless 3D point clouds of shapes, and adapt it to our setting of learning correspondences in variable-length, sparse and noisy SoC radar 3D point clouds.

#### A. Network Architecture

As seen in Fig. 2, the first step in our network applies the *PointNet* sub-network to two consecutive input point clouds to embed them in a higher-dimensional space. We obtain the input point clouds by finding the length  $N$  of the longest point cloud in our whole dataset and padding all point clouds to that length with zero vectors of size  $1 \times 3$ . We also append a zero vector to the beginning of each point cloud which is necessary to train our network as a multi-label classifier (see subsection III-B). That way, each of the resulting input point clouds has the shape  $(N + 1) \times 3$ . We apply the embedding sub-network on a *per-point* basis, which means that for a single point in the input represented by three coordinates  $\{x, y, z\}$  we obtain an embedding vector of size  $1 \times E$ , where  $E$  is the chosen embeddings size. *PointNet* parameters are learned and shared among all points in the input. In the next step, we forward the embedded points in each point cloud to two transformer sub-networks. The role of each of the transformers is to compute new embeddings of each point cloud using the contextual information of both point clouds jointly. That way, the network can leverage the attention mechanism on both point clouds together which permits finding the

embeddings tailored to the specific kind of point clouds, thus making the embeddings task-specific [25]. Specifically, in each transformer block apart from passing to the decoder the output embeddings of the encoder, we also pass in the other point cloud input embeddings. The final embeddings are calculated by summing the transformer output, which encodes the mutual information about the point clouds, with the initial embeddings. The output of the network is obtained by calculating the dot product of the final embeddings of each point in the first point cloud with the final embeddings of each point in the second point cloud. This operation yields an output matrix of shape  $(N + 1) \times (N + 1)$ . Entries in the output matrix express the affinity between points in each input point cloud, that is, the likelihood that a pair of points form a correspondence.

#### B. Network Training And Inference

In the case of noisy, sparse and variable-length SoC radar point clouds, we cannot conveniently train the network on the odometry error using the Singular Value Decomposition (SVD) as in [18] and [25]. We thus propose a different approach to calculating the loss in our network. Namely, we treat the index of every point in the point cloud as its class label and reserve the class label (and the index) "0" for any non-matched points. The output matrix of the network is structured as follows,

$$\mathbf{G} = \underbrace{\begin{pmatrix} \bullet & \bullet & \cdots & \cdots & \cdots & \bullet & \bullet \\ \bullet & c_{11} & c_{12} & \cdots & c_{1K} & \cdots & \bullet \\ \vdots & c_{21} & c_{22} & \cdots & c_{2K} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ \vdots & c_{L1} & c_{L2} & \cdots & c_{LK} & \cdots & \vdots \\ \bullet & \vdots & \vdots & \vdots & \vdots & \ddots & \bullet \\ \bullet & \bullet & \cdots & \cdots & \cdots & \bullet & \bullet \end{pmatrix}}_{N+1} \left. \vphantom{\begin{pmatrix} \bullet & \bullet & \cdots & \cdots & \cdots & \bullet & \bullet \\ \bullet & c_{11} & c_{12} & \cdots & c_{1K} & \cdots & \bullet \\ \vdots & c_{21} & c_{22} & \cdots & c_{2K} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ \vdots & c_{L1} & c_{L2} & \cdots & c_{LK} & \cdots & \vdots \\ \bullet & \vdots & \vdots & \vdots & \vdots & \ddots & \bullet \\ \bullet & \bullet & \cdots & \cdots & \cdots & \bullet & \bullet \end{pmatrix}} \right\} N+1 \quad (1)$$

and is obtained by taking the dot product of the final embeddings  $\Sigma_1$  and  $\Sigma_2$  of the two consecutive input point clouds as shown in the Fig. 2. Entries in row  $i$  express the likelihood that point  $i$  in the first input point cloud is a correspondence to point  $j$  in the second input point cloud, where  $i = 1 \dots L$ ,  $j = 1 \dots K$  and  $L$ ,  $K$  are lengths of the respective point clouds. Only the green sub-matrix in the output matrix in Eq. 1 carries useful information. All other elements result from adding the "0" (non-matched) class label and from padding the point clouds to equal length with zeros. In particular, appending the zero vector at the beginning of each input point cloud creates the 0-th row and column in  $\mathbf{G}$ . This is crucial during training, since we assign a "0" class (0-th index) in the ground-truth for every point in the first point cloud which does not have a match in the second point cloud.

During inference, since point clouds are usually of different lengths, we solve the LSA problem on the green sub-

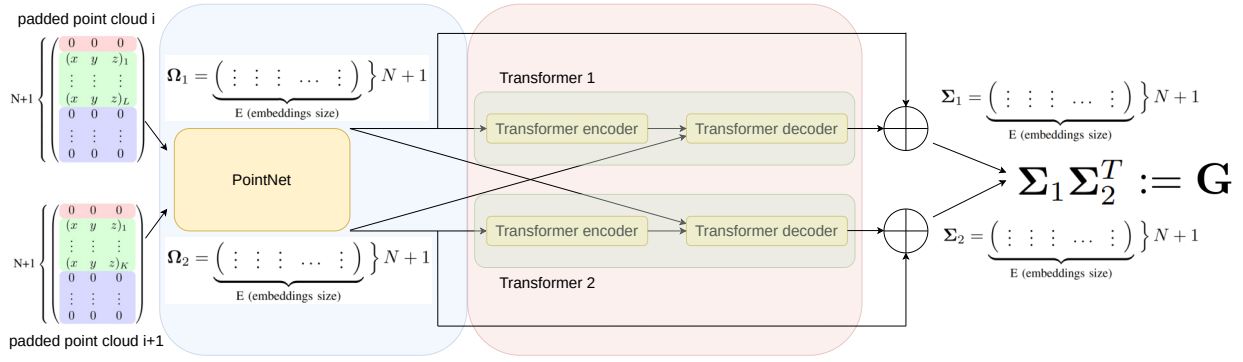


Fig. 2. Our learning framework is based on the architecture proposed in [25] for dense and structured point clouds registration. We adapted it to our scenario with highly noisy and sparse consumer-grade SoC radar 3D point clouds. Note the zero padding of the input point clouds (subsec. III-B) and the pre-pended "0", which account for the variable-length input and the class label attributed to any point with no match, respectively. Input point clouds are passed to the embedding *PointNet* sub-network (light blue block). Individual embeddings enter the transformer sub-networks (light red block) where the self and the reciprocal attention is computed for each point cloud. Output matrix  $\mathbf{G}$  representing the mutual affinity between points in the input point clouds is obtained by calculating the dot product of the final embeddings. Within the output we search for the set of matches by solving the LSA (see subsec. III-B).

matrix to find the optimal assignment,

$$\max \sum_{i=1}^L \sum_{j=1}^K \mathbf{C}_{i,j} \mathbf{X}_{i,j} \quad (2)$$

Where  $\mathbf{X}$  is a boolean matrix where  $\mathbf{X}_{i,j} = 1$  iff row  $i$  is assigned to column  $j$  and  $L, K$  are lengths of the input point clouds.  $\mathbf{C}$  is the green sub-matrix from Eq. 1. Constraints of the problem are such that each row is assigned to at most one column and each column to at most one row. For each entry in the solution we apply an experimentally chosen threshold to decide whether it is a match or not. LSA is usually solved using Munkres algorithm [31].

During training, the structuring of the network output shown in Eq. 1 allows us to compute the cross-entropy loss between each row (the index of which is the index of a point in the first point cloud) and the ground-truth label (index of the matched point in the second point cloud or the "0" index for a non-match), as follows,

$$l_n = -\frac{1}{M} \sum_{i=1}^M \log \left( \frac{\exp(\mathbf{G}(p_i, q_i))}{\sum_{j=1}^{N+1} \exp(\mathbf{G}(p_i, q_j))} \right) \quad (3)$$

where  $n = 1 \dots B$  and  $B$  is the mini-batch size, and  $(p_i, q_i)$ ,  $i = 1 \dots M$  are the indices of the ground-truth matches in the first and second point cloud, respectively.  $N+1$  is the length of each row (and column) of the  $\mathbf{G}$  matrix.

Preparing the input data and ground-truth labels for training requires pre-processing. In order to generate the ground-truth point correspondences, we use the spatial transformation from the motion capture system between radar frames of every two consecutive radar measurements. Using the spatial information, we transform the 3D points from the first point cloud to the frame of the second point cloud and perform geometric matching by solving the LSA optimization (this time minimization) problem as in Eq. 2 but this time on a matrix whose entries are euclidean distances between points in both point clouds expressed in the second point cloud

frame, as in,

$$\mathbf{C}_{i,j} = \|\mathcal{R}_c \mathbf{p}_{\mathcal{P}_i}^c - (\mathcal{R}_c \mathbf{R}_{\mathcal{R}_p} \mathcal{R}_p \mathbf{p}_{\mathcal{P}_j}^p + \mathcal{R}_c \mathbf{p}_{\mathcal{R}_p})\| \quad (4)$$

where  $\mathcal{R}_{\{p,c\}} \mathbf{p}^{\{p,c\}}_{\mathcal{P}}$  are all points from the previous radar scan at time instance  $t_p$  and from the current radar scan at  $t_c$ , in the previous and current radar frames, respectively.  $\mathcal{R}_c \mathbf{R}_{\mathcal{R}_p}$  and  $\mathcal{R}_c \mathbf{p}_{\mathcal{R}_p}$  are rotation and translation parts of the spatial transform between the current and previous radar frames obtained from the motion capture system. That way, we obtain the ground-truth class labels (indices of matched points in each point cloud). We shift the obtained labels by one to account for the "0" class for every non-matched point. Pre-processing the input radar data consists only of removing the points outside of the FOV of the sensor and aforementioned zero padding.

## IV. RESULTS

### A. Experiments

In order to train and validate our learning framework, we collect a dataset consisting of 13 UAV trajectories with two different platforms (ARDEA-X and CNS-UAV) described in [21] (see Fig. 1). Both platforms use the same consumer-grade TI AWR1843BOOST FMCW SoC radar chip mounted and configured in the same way, as well as the same pixhawk IMU sensor. We record radar and IMU sensor measurements as well as the ground truth pose of the UAV using a motion capture system. We divide this dataset into 8 training and 5 validation trajectories. The training dataset contains trajectories between 150 m - 180 m in length flown manually. The validation dataset contains shorter trajectories between 11 m - 38 m, among which some are manually flown while others are pre-planned, executed using specified waypoints. We also validate our method on five sequences from the public open-source Coloradar dataset [28]. Coloradar sequences are collected using hand-held sensor rig containing the same TI radar chip as ARDEA-X and CNS-UAV platforms. Coloradar sequences contain much

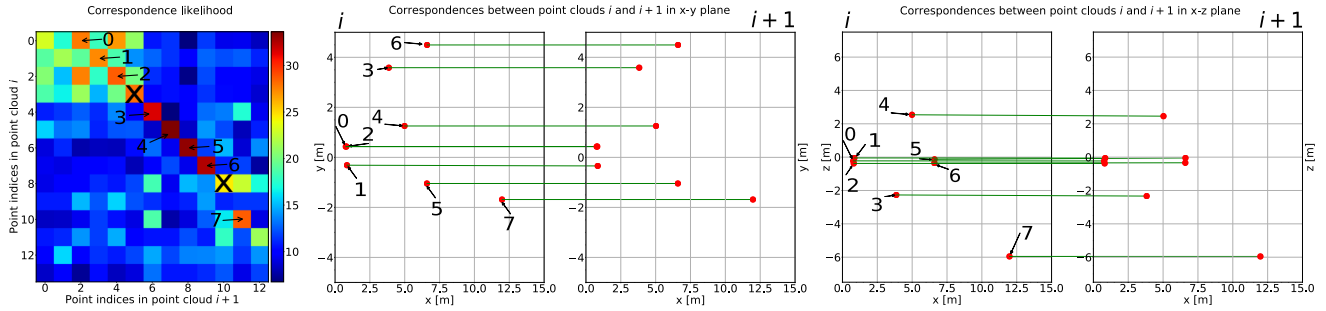


Fig. 3. Leftmost part of the figure shows the correspondence likelihood matrix (the green part of the  $\mathbf{G}$  matrix in the Eq. 1) inferred from the learned proposed model for two consecutive point clouds  $i$  and  $i + 1$ . Rightmost and the middle parts show the resulting 3D point correspondences for the same point clouds. Correspondences are shown on  $xy$  (middle sub-plot) and  $xz$  (right sub-plot) planes. Numbers marking the entries in the matrix are consistent with the indices of the matches in sub-plots. During inference, we prune the matches outside the FOV of the radar and with correspondence likelihoods below the empirically found acceptance threshold (shown as "x" in the leftmost plot). Thus, some entries in the matrix have no corresponding matches despite having relatively hot values. Only matched points are shown to not clutter the figure.

more aggressive motion than the self-collected dataset, thus being more challenging. While in three sequences a motion capture system is used as the ground-truth, the other two contain high-precision LiDAR-Inertial Odometry (LIO) data. We train our network using PyTorch open-source package. We assess our 3D point matching framework qualitatively in an indirect way by plugging it into an open-source RIO framework from [12], [21] and comparing the accuracy of the obtained estimates to the case when the default (non-learning) matching algorithm is used. Between executions of the RIO estimator, we only exchange the matching algorithm, all other parameters and settings remain the same. The RIO which we use for validation is EKF-based and in the update step uses three sources of information: 3D point matches, Doppler velocities and persistent features. For the self-collected dataset, we execute the RIO in two configurations: in the default configuration with both Doppler and point matches, and with only point matches enabled in the update step. For the Coloradar we only use default configuration (point matches and Doppler). In all cases, we compile the RIO framework without persistent features in the update. In the case of our learning-based framework, we execute the inference on the learned model in a Python node before feeding it to the RIO. The inference with a non-optimized model takes on average 0.0273 s, which means the optimized implementation would lend itself to real-time use. The non-learning matching algorithm is implemented within the RIO estimator as its default matching algorithm and described in [13]. Both RIO and inference node are executed offline on the recorded sensor data on an Intel Core i7-10850H vPRO laptop with 16 GB RAM.

## B. Evaluation

For each trajectory from both validation datasets (self-collected and Coloradar), we compute the norm of RMSE of the position and attitude estimates along with the mean and the standard deviation of the obtained values when switching the matching algorithm inside the RIO estimator (see Tab. I and Tab. II). In the case of the self-collected dataset, we compute the norm of RMSE values in the case when both

point matches and Doppler are used, and additionally, when only point matches residuals are used in the update step of the EKF in the RIO. Our comparisons show that when only point matches are used in the update step, which is the most direct way of assessing the performance of our learning-based matching framework, we obtain a striking 70.38 % improvement in the position estimate accuracy on average. When compared to the state-of-the-art configuration of the RIO, that is, with both Doppler and point matches residuals enabled, we obtain a 14.28 % improvement in position accuracy on average. With the Coloradar dataset, we only execute the full configuration containing both Doppler and point matches residuals and obtain 19.01 % improvement in position accuracy on average. The motion in the Coloradar dataset is too aggressive for the configuration using only point matches to work properly (for both learning and non-learning). For our recorded dataset, as can be seen in Fig. 5, when point matches are used as the sole source of information for measurement updates in the EKF RIO, the presented method allows for much more accurate estimation than the non-learning approach and when combined with the Doppler velocity measurements, greatly reduces the final error.

In Fig. 3, we can see how the trained network infers the correspondences. Note how matches 0 and 2 lie very close geometrically and hence points involved in them have all high mutual affinities, nevertheless, the network still makes correct distinction between them. Points involved in match 1, which also lie close to points involved in matches 0 and 2, do not have high affinity with points in 0, 2. This can be explained by looking at the the middle plot and observing that on the  $xy$ -plane, points in match 1 are offset from points in 0 and 2. For points in matches 5 and 6, despite all of them being close on the  $xz$ -plane, our network correctly assigns the mutual affinities, since on the  $xy$ -plane, the points are clearly separated resulting in unambiguous matches. Points in matches 3, 4, 7 are significantly spaced in both planes, thus all have strong unambiguous mutual affinity values. Points (3, 5) are not considered a match despite their high mutual affinity because at least one of them is outside of

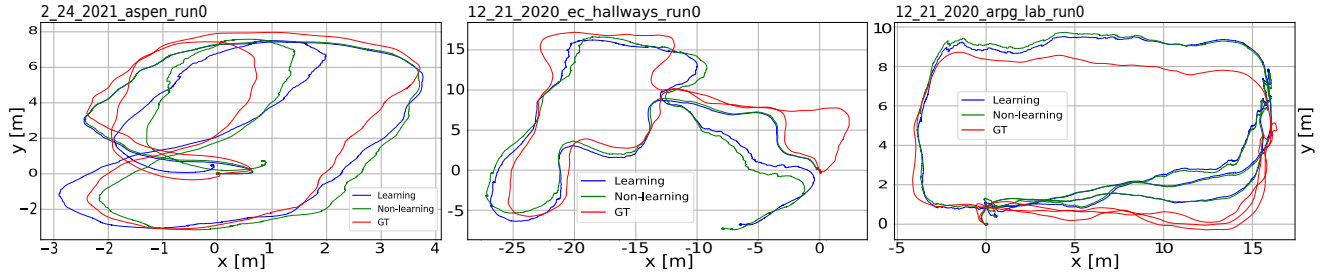


Fig. 4. Three illustrative sequences (also used in [29]) out of five chosen from the Coloradar dataset for evaluation. From left to right: "2\_24\_2021\_aspen\_run0", "12\_21\_2020\_ec\_hallways\_run0", "12\_21\_2020\_arpg\_lab\_run0". In red we mark the ground-truth, and in green the non-learning and in blue the learning (proposed) approaches, respectively. Across all used Coloradar sequences, using our learning-based matching method results in a decrease of the norm of position RMSE by 19%.

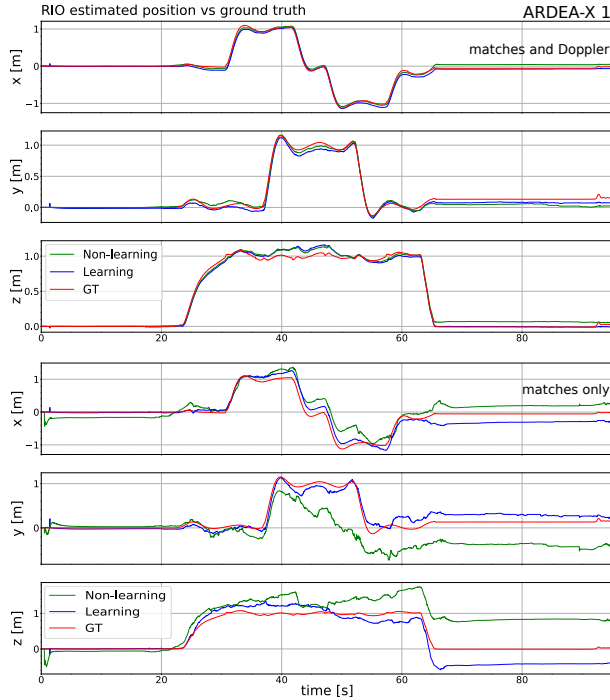


Fig. 5. Estimated position of the ARDEA-X UAV for the flight 1 from the Tab. II and I. We plot the  $x, y, z$  coordinates of the estimate against the ground-truth for the configuration with matches and Doppler, and only matches used in the update step of the EKF RIO framework used for validation. Each configuration is executed with the proposed learning-based and the default non-learning matching algorithm. In red the ground-truth, in green and blue non-learning and learning approaches, respectively.

the FOV. Similarly, the points (8, 10) are not counted as a match since their correspondence likelihood value is below the empirically determined acceptance threshold.

In Fig. 4 we plot the estimation results for three out of chosen five Coloradar sequences for both used matching methods. From the five sequences, "12\_21\_2020\_ec\_hallways\_run0", "2\_24\_2021\_aspen\_run0" and "12\_21\_2020\_arpg\_lab\_run0" are also chosen in the latest state-of-the-art work on RIO presented in [29] where the authors also provide the norm of RMSE of position and attitude estimate for their method. This allows us to note that for sequences "2\_24\_2021\_aspen\_run0" and

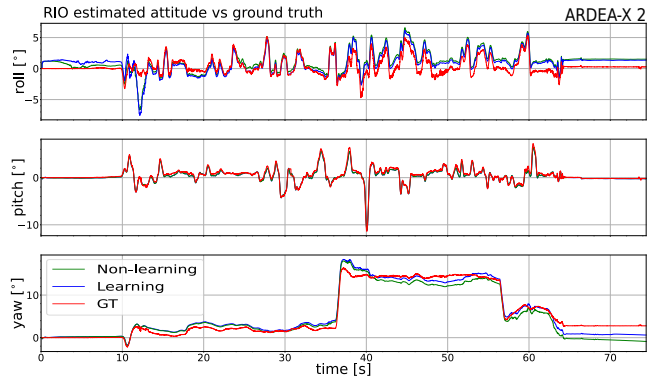


Fig. 6. Estimated attitude of the ARDEA-X UAV for the flight 2 from the Tab. II and I. We plot the roll, pitch and yaw angles of the estimate against the ground-truth for the configuration with matches and Doppler used in the update step of the EKF in RIO framework used for validation, when executed with the proposed learning-based (in blue) and the default non-learning matching algorithm (in green). In red we plot the ground-truth attitude.

"12\_21\_2020\_arpg\_lab\_run0" our method outperforms [29] 3.044 m (ours) to 3.820 m and 5.388 m (ours) to 6.101 m, respectively in position and  $13.372^\circ$  (ours) to  $30.905^\circ$ , and  $8.287^\circ$  (ours) to  $12.640^\circ$ , respectively in attitude. For the sequence "12\_21\_2020\_ec\_hallways\_run0" our method performs worse, 9.927 m (ours) to 5.223 m in position as well as in attitude  $18.357^\circ$  (ours) to  $16.070^\circ$ .

Note, however, that in [29] the underlying estimator is different to the one we use here. Thus, we compare in Tab. I and Tab. II more rigorously the specific benefit of the learning-based matching. For this, we implement the RIO framework of [12], [21] with Doppler and point matches as well as only point matches from our learning based approach and with the original non-learning approach. Tab. I clearly shows that our learning based approach improves the performance in the position estimation in almost all runs. When using both Doppler and point matches, the improvement using our approach is in average over 14% or a bit over 5 cm on our own datasets, and 19% or 8 cm on the Coloradar datasets. With a standard deviation much higher than the improvements, these results have limited statistical relevance. *However, when using only point matches, the improvement is much higher with over 70% or nearly 1.5 m on our*

TABLE I

RMSE NORM VALUES OF POSITION ESTIMATE FOR BOTH MATCHING METHODS ACROSS SELF-COLLECTED UAV FLIGHTS AND SEQUENCES FROM OPEN-SOURCE COLORADAR DATASET.

Nr	Self-collected dataset   RMSE   of position [m]			
	Doppler and matches		Matches only	
	Learning (ours)	Non learning	Learning (ours)	Non learning
1	<b>0.083</b>	0.101	<b>0.351</b>	0.748
2	<b>0.212</b>	0.272	<b>0.793</b>	1.795
3	0.714	<b>0.704</b>	<b>0.745</b>	1.608
4	0.380	<b>0.338</b>	<b>0.431</b>	1.074
5	<b>0.230</b>	0.473	<b>0.736</b>	5.088
Average	0.324	0.378	0.611	2.063
Std. dev.	0.217	0.202	0.183	1.558
Sequence	Coloradar dataset   RMSE   of position [m]			
aspen_run0	<b>3.044</b>	5.327	x	x
arpg_lab_run0	<b>5.388</b>	6.080	x	x
ec_hallways_run0	<b>9.927</b>	11.523	x	x
aspen_run4	<b>5.315</b>	6.296	x	x
aspen_run5	<b>3.466</b>	4.280	x	x
Average	5.428	6.701	x	x
Std. dev.	2.728	2.808	x	x

datasets. With a standard deviation of a bit over 18 cm, this clearly underlines the estimation improvement due to our approach. The Coloradar dataset trajectories are too agile for the estimator to work properly when not including Doppler information, hence the 'x' in the lower right part of the table.

The benefit of our approach regarding the attitude estimation is less clear. Using both Doppler and point matches we observe in average a decrease in performance of 5% or roughly half a degree on both our and the Coloradar datasets. When only using point matches, we observe in average a bit more than 9% or nearly 1 degree performance drop on our datasets. Note, however, that these differences are barely statistically relevant since the standard deviation in all cases is higher than 5 degrees for our approach. Thus, we can conclude that while our method has a clearly positive impact on the position estimate, the attitude barely benefits from the new approach. The root cause of the reduced benefit in attitude is to be investigated further – we assume a connection to the bad angular resolution and high angular noise that comes with this type of sensors. We plot the attitude estimates for one of the trajectories from the self-collected dataset in the Fig. 6.

## V. CONCLUSIONS

In this paper, we presented a novel self-supervised learning framework for finding 3D point correspondences in sparse and noisy point clouds from a SoC FMCW radar. To our knowledge, this is the first learning approach addressing the problem of data association in the challenging setting of 3D point cloud measurements from low-cost, low-power, lightweight, consumer-grade radar chips. In our framework, we leverage the *PointNet* architecture to compute individual point embeddings in each of the two consecutive input point clouds. Subsequently, using transformer architecture and its

TABLE II

RMSE NORM VALUES OF ATTITUDE ESTIMATE FOR BOTH MATCHING METHODS ACROSS SELF-COLLECTED UAV FLIGHTS AND SEQUENCES FROM OPEN-SOURCE COLORADAR DATASET.

Nr	Self-collected dataset   RMSE   of attitude [°]			
	Matches and Doppler		Matches only	
	Learning (ours)	Non learning	Learning (ours)	Non learning
1	3.996	<b>3.203</b>	<b>0.966</b>	6.689
2	<b>1.642</b>	2.036	4.730	<b>3.597</b>
3	17.068	<b>16.945</b>	17.198	<b>17.067</b>
4	10.058	<b>9.712</b>	8.547	<b>6.234</b>
5	18.158	<b>16.551</b>	21.108	<b>14.305</b>
Average	10.184	9.689	10.510	9.578
Std. dev.	7.453	7.077	8.446	5.782
Sequence	Coloradar dataset   RMSE   of attitude [°]			
aspen_run0	13.372	<b>9.389</b>	x	x
arpg_lab_run0	8.287	<b>7.506</b>	x	x
ec_hallways_run0	<b>18.357</b>	21.666	x	x
aspen_run4	<b>7.776</b>	8.117	x	x
aspen_run5	9.363	<b>7.947</b>	x	x
Average	11.431	10.925	x	x
Std. dev.	6.045	4.451	x	x

attention mechanism, we augment the initial embeddings with the reciprocal information from both inputs, to finally form the matching likelihood matrix by calculating the dot product of the augmented embeddings. We provide a self-supervision method using set-based multi-label classification cross-entropy loss, where the ground-truth set of matches is calculated by solving the LSA optimization problem. Employing multi-label classification cross-entropy loss enables directly using correspondences in training. This is crucial since training on odometry error using e.g. SVD, as used in methods for scanning radars or dense point clouds, is not feasible with the sparse and noisy measurements from the SoC radar sensor that we use in this work. We applied our framework to the task of RIO estimation on a small-sized UAV and showed that it outperforms the default non-learning 3D point matching method. In particular, in an open-source state-of-the-art RIO framework we switched the 3D point matching algorithm from the default non-learning one to the one presented in this paper while keeping all other settings and parameters unchanged. The reduction in the norm of RMSE of position estimate calculated over the whole real-world validation dataset in both cases when only matches, and matches with Doppler velocity are used in the estimator reveals that our learning-based method surpasses the non-learning one. We make both our framework and the datasets open-source for the benefit of the research community.

## ACKNOWLEDGMENT

Authors would like to thank Florian Steidle, Julius Quell and Marcus G. Müller from the MAV Exploration Team at the Institute of Robotics and Mechatronics in the German Aerospace Center (DLR) for hosting the author and helping with the dataset acquisition.

## REFERENCES

- [1] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, 2012.
- [2] M. Nissov, N. Khedekar, and K. Alexis, "Degradation resilient lidar-radar-inertial odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8587–8594.
- [3] J. Michalczyk, M. Scheiber, R. Jung, and S. Weiss, "Radar-inertial odometry for closed-loop control of resource-constrained aerial platforms," in *2023 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2023, pp. 61–68.
- [4] K. Harlow, H. Jang, T. D. Barfoot, A. Kim, and C. Heckman, "A new wave in robotics: Survey on recent mmwave radar applications in robotics," *IEEE Transactions on Robotics*, vol. 40, pp. 4544–4560, 2024.
- [5] Y. Cheng, J. Su, M. Jiang, and Y. Liu, "A novel radar point cloud generation method for robot environment perception," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3754–3773, 2022.
- [6] C. Doer and G. F. Trommer, "Radar inertial odometry with online calibration," in *2020 European Navigation Conference (ENC)*. IEEE, 2020, pp. 1–10.
- [7] —, "An ekf based approach to radar inertial odometry," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2020, pp. 152–159.
- [8] —, "Yaw aided radar inertial odometry using manhattan world assumptions," in *2021 28th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 2021, pp. 1–9.
- [9] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Agha-Mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5739–5746.
- [10] C. Kim, G. Bae, W. Shin, S. Wang, and H. Oh, "EKF-based radar-inertial odometry with online temporal calibration," 2025. [Online]. Available: <https://arxiv.org/abs/2502.00661>
- [11] Y. S. Park, Y.-S. Shin, J. Kim, and A. Kim, "3d ego-motion estimation using low-cost mmwave radars via radar velocity factor for pose-graph slam," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7691–7698, 2021.
- [12] J. Michalczyk, R. Jung, C. Brommer, and S. Weiss, "Multi-state tightly-coupled ekf-based radar-inertial odometry with persistent landmarks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4011–4017.
- [13] J. Michalczyk, R. Jung, and S. Weiss, "Tightly-coupled ekf-based radar-inertial odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 12 336–12 343.
- [14] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni, and A. Markham, "Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3314–3323, 2020.
- [15] Q. Huang, Y. Liang, Z. Qiao, S. Shen, and H. Yin, "Less is more: Physical-enhanced radar-inertial odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 966–15 972.
- [16] Y. Zhuang, B. Wang, J. Huai, and M. Li, "4d iriom: 4d imaging radar inertial odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3246–3253, 2023.
- [17] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6045–6052.
- [18] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9484–9490.
- [19] K. Burnett, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Radar odometry combining probabilistic estimation and unsupervised feature learning," *arXiv preprint arXiv:2105.14152*, 2021.
- [20] P. Lutz, M. G. Müller, M. Maier, S. Stoneman, T. Tomić, I. von Bargen, M. J. Schuster, F. Steidle, A. Wedler, W. Stürzl, and R. Triebel, "Ardea—an mav with skills for future planetary missions," *Journal of Field Robotics*, vol. 37, no. 4, pp. 515–551, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21949>
- [21] J. Michalczyk, J. Quell, F. Steidle, M. G. Müller, and S. Weiss, "Tightly-coupled factor graph formulation for radar-inertial odometry," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3364–3370.
- [22] D. Brodeski, I. Bilik, and R. Giryes, "Deep radar detector," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.
- [23] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller, "Ghost target detection in 3d radar data using point cloud based deep neural network," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 10 398–10 403.
- [24] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187–199, Apr. 2021. [Online]. Available: <http://dx.doi.org/10.1007/s41095-021-0229-5>
- [25] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3522–3531.
- [26] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.
- [27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
- [28] A. Kramer, K. Harlow, C. Williams, and C. Heckman, "Coloradar: The direct 3d millimeter wave radar dataset," *The International Journal of Robotics Research*, vol. 41, no. 4, pp. 351–360, 2022. [Online]. Available: <https://doi.org/10.1177/02783649211068535>
- [29] Y. Xu, Q. Huang, S. Shen, and H. Yin, "Incorporating point uncertainty in radar slam," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2168–2175, 2025.
- [30] V. Kubelka, E. Fritz, and M. Magnusson, "Do we need scan-matching in radar odometry?" in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 710–13 716.
- [31] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.