TransLoc4D: Transformer-based 4D Radar Place Recognition

Guohao Peng[1], Heshan Li[1], Yangyang Zhao[2], Jun Zhang[1*],
Zhenyu Wu[1], Pengyu Zheng[2], Danwei Wang[1]
Nanyang Technological University, Singapore
[1]{guohao.peng, heshan.li, jun.zhangj, zhenyu.wu, edwwang}@ntu.edu.sg,
[2]{zhao0417, pzheng002}@e.ntu.edu.sg

Abstract

Place recognition is crucial for unmanned vehicles in terms of localization and mapping. Recent years have witnessed numerous explorations in the field, where 2D cameras and 3D LiDARs are mostly employed. Despite their admirable performance, they may encounter challenges in adverse weather such as rain and fog. Hopefully, 4D millimeter-wave radar emerges as a promising alternative, as its longer wavelength makes it virtually immune to interference from tiny particles of fog and rain. Therefore, in this work, we propose a novel 4D radar place recognition model, TransLoc4D, based on sparse convolutions and Transformer structures. Specifically, a MinkLoc4D backbone is first proposed to leverage the multi-modal information from 4D radar scans. Rather than merely capturing geometric structures of point clouds, MinkLoc4D additionally explores their intensity and velocity properties. After feature extraction, a Transformer layer is introduced to enhance local features before aggregation, where linear self-attention captures the long-range dependencies of the point cloud, alleviating its sparsity and noise. To validate TransLoc4D, we construct two datasets and set up benchmarks for 4D radar place recognition. Experiments validate the feasibility of TransLoc4D and demonstrate it can robustly deal with dynamic and adverse environments.

1. Introduction

Place recognition is a fundamental component of autonomous navigation systems, especially when operating in GPS-denied environments. It is typically handled as a retrieval task [3, 36, 50] to provide a good initial value for localization optimization [8, 40, 41] and navigation decision-making. Traditional methods relying on 2D images and 3D point clouds have been extensively studied and become the mainstream solutions for place recognition. However, the inherent characteristics of cameras and LiDARs make these methods vulnerable to challenging weather conditions such as rain, snow, or fog. This hinders their applicability for robust place recognition in inclement-weather scenarios.

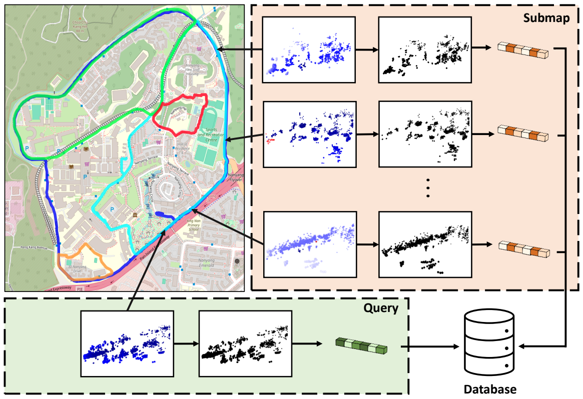

Figure 1. Schematic of the 4D Radar Place Recognition (4DRPR) pipeline. Each 4D radar scan of the trajectory is preprocessed and characterized into a novel 4D point cloud representation, which is then fed into the TransLoc4D encoding network to generate a global compact descriptor for instance retrieval.

Recently, 4D millimeter-wave radar [24, 51] has become a promising alternative and garnered growing interest. Since millimeter waves have longer wavelengths, they are effective at penetrating raindrops and fog particles. Therefore, compared with 2D cameras and 3D LiDAR, 4D radar can better adapt to harsh weather conditions and low-light environments. While existing 2D and 3D methods have to consider domain adaptation [1, 44, 45] or complementary sensor fusion [21, 22, 46] to cope with lighting and weather changes, 4D radar alone is sufficient for robust place recognition in diverse environmental conditions.

Despite the advantages, 4D radar produces sparse point cloud scans with lower spatial resolution than 3D LiDAR, as in Fig. 1. This leaves points of a 4D radar scan lacking context information. Besides, aliasing caused by multipath echoes may blur the shape of static architectures and dy-

[*] Corresponding author

namic objects. These factors diminish the precision of the 4D radar scene depiction. Specifically for place recognition, a paradigm is needed to exploit the multimodal information of intensity and Doppler velocity from 4D radar scans. Differences in observation distance and angle, as well as the external environment, can influence the intensity of the reflection detected from the same object. The radial velocity of the object is also affected by changes in its azimuth angle relative to the radar and the vehicle's own driving speed. Therefore, how to take advantage of the geometry, intensity, and velocity information of 4D radar scans, but ignoring the interference contained therein, is the major challenge of applying 4D radar to the place recognition task.

Taking these issues into account, we propose a novel 4D radar place recognition model, TransLoc4D. Its encoding architecture consists of three modules: sparse convolutions for feature extraction, Transformer for feature enhancement, and GeM pooling for descriptor generation. Firstly, point cloud processing and a MinkLoc4D backbone are proposed to exploit the multimodal information of geometry, intensity, and velocity from 4D radar scans. Rather than operating directly on raw point clouds, MinkLoc4D discretizes point clouds into sparse voxelized representation [26]. While most preliminary 3D methods [10, 19, 42] merely extract geometry information from binary voxelized features, MinkLoc4D proposes a novel paradigm that reforms additional intensity and radial velocity attributes of 4D radar scans into numerical features. Specifically, RANSAC-based ego-velocity estimation strategy is first employed to remove dynamic points and estimate vehicle ego-velocity. Then after preprocessing, sparse convolutions are applied to encode geometric, intensity, and motion patterns of the refined point clouds into latent features.

Following backbone feature extraction, a Transformer layer is introduced for spatial feature enhancement. The sparse convolutional backbone only extracts local patterns between points and their neighborhoods, while their long-range contextual association is unexplored. In this case, a Transformer with interpolation [6] and linear self-attention [37] is employed to play a complementary role. It interpolates sparse voxels, captures their long-range dependencies, and associates global context to enhance their feature representation. Finally, the global descriptor of a 4D radar scan is generated by aggregating the enhanced local features via Generalized-Mean pooling [32]. The overall TransLoc4D encoding architecture is differentiable and can be optimized through end-to-end training.

Since there has been no open-sourced dataset specifically for 4D radar place recognition, we create two benchmark datasets based on the open-source NTU4DRadLM [49] and SJTU4D [24] datasets, as well as supplementary data newly collected at NTU campus. Experimental results demonstrate the feasibility of our TransLoc4D and provide benchmarks for 4D radar place recognition. Overall, the contributions of this work are summarized as follows:

- A novel encoding architecture, TransLoc4D, is proposed. It is the first end-to-end network aimed at tackling place recognition based on 4D radar attributes.
- A MinkLoc4D backbone is proposed for feature extraction based on the multimodal information of geometry, intensity, and velocity from 4D radar scans.
- A linear Transformer is introduced to capture the long-range dependencies to enhance feature representation.
- Two datasets are constructed for 4DRPR, on which our TransLoc4D is validated and benchmarks are set.

2. Related Work

2D Visual Place Recognition. Compared with handcrafted local features [31, 43], learning-based architectures demonstrate excellent potential. Integrating specific structures [13, 30, 47] or attention mechanisms [18, 29, 30], NetVLAD [2] variants show superior performance. In addition to innovations in encoding architectures, the explorations of better training metrics [3,4,12,13,48] or matching strategies [23] have also been ongoing. CosPlace [3] and EigenPlace [4] propose to train the model for a categorization task. Through supervised training, better performance can be learned with a simple model structure.

3D LiDAR Place Recognition (3DLPR) is typically handled as a point cloud retrieval task via global or local descriptor matching. PointNetVLAD [36] is the seminal end-to-end architecture for 3DLPR. Rather than processing on raw point clouds, MinkLoc3D [19] pioneers the use of sparse voxelized representations, forwarding them to the 3D convolutional network. TransLoc3D [42] proposes an adaptive receptive field module that applies channel attention to multiscale features, which are then integrated by a Transformer and NetVLAD. PPT-Net [15], SVT-Net [10], and PTC-Net [6] exploit various Transformer modules to learn long-range contextual properties. From the perspective of learning metric, MinkLoc3Dv2 [20] proposes a Truncated Smooth-AP loss suitable for large training batches.

However, the aforementioned methods extract features from either raw point clouds or their voxelized representations, where only geometric patterns are captured. The intensity property of 3D LiDAR, proven effective in constructing non-learned local descriptors for 3D place recognition [14, 52], have not been fully utilized in global descriptor generation. In the limited literature, Intensity Scan Context [38] proposes the scan context identification based on intensity characteristics, which however is not learnable. MinkLoc3D-SI [52] inherits the spherical point representation and employs 3D convolutional architecture to utilize

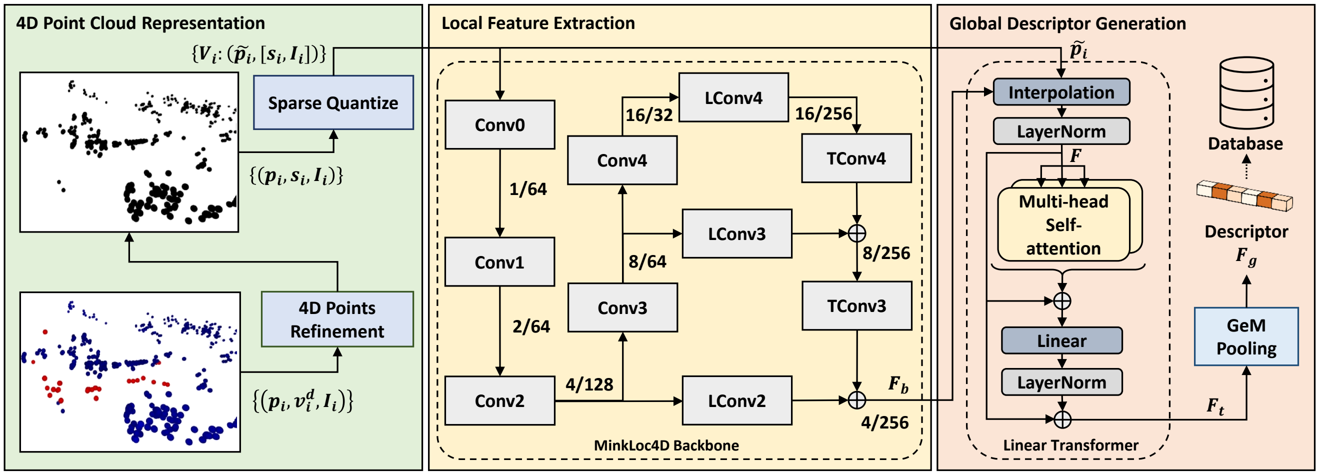

Figure 2. Overview of the TransLoc4D encoding architecture. First, preprocessing is performed to refine the 4D point cloud based on ego-velocity regression and RANSAC filtering. Geometric, velocity, and intensity attributes are reformed and integrated into a novel 4D point representation, which is taken as input by the 3D sparse convolutional backbone MinkLoc4D for local feature extraction. Finally, local features are enhanced by a Transformer via global context integration and aggregated as the final global descriptor by GeM pooling.

intensity information. Our TransLoc4D uses the Cartesian point representation and integrates the additional velocity characteristics besides geometric and intensity.

4D Radar Place Recognition (4DRPR). Among early attempts at radar place recognition, Kidnapped Radar [35] provides a rotation-invariant solution for spinning radars. AutoPlace [5] proposes the first solution for automotive radar, using a deep neural network for spatial-temporal feature embedding. With the advent of 4D millimeter-wave radar, 4DRPR is becoming an emerging solution robust against adverse weather conditions. However, there are currently no open-source datasets and methods specifically targeting 4DRPR. The two open-source datasets with 4D radar scans and sufficient closed loops are SJTU4D [24] proposed for autonomous driving, and NTU4DRadLM [49] proposed for 4D radar Simultaneous Localization and Mapping (SLAM) [7, 50]. We create benchmark datasets for 4DRPR based on their GPS and 4D radar data. While NTU4DRadLM proposes a loop closure detection module based on Intensity Scan Context [17, 38], it is handcrafted and relies on the rough position provided by front-end odometry. Our TransLoc4D is a differentiable architecture that can be fine-tuned by end-to-end training.

3. Proposed Method

This section expounds on the details of the TransLoc4D. Sec. 3.1 proposes point cloud preprocessing and a 4D representation. Sec. 3.2 introduces the MinkLoc4D backbone. It extracts features from multimodal characteristics of 4D radar scans. The Transformer and GeM pooling for global context integration and descriptor generation are elaborated in Sec. 3.3. Sec. 3.4 describes the training process. Fig. 2 illustrates the overall TransLoc4D framework.

3.1. 4D Point Preprocessing and Representation

4D millimeter-wave radar provides five primitive attributes of point clouds: range, azimuth, altitude, Doppler velocity, and intensity. Through the transformation of the first three attributes, the coordinates of the points in the Cartesian coordinate system can be obtained as $\{\boldsymbol{p}_i = (x_i, y_i, z_i)^T\}$. Compared with 3D LiDAR, velocity is a unique attribute of 4D radar. The radial relative velocity v_i^d of a point can be obtained by the Doppler effect. In TransLoc4D, the velocity information is utilized from two aspects: (1) remove noise points caused by multipath aliasing and dynamic points to reduce their interference; (2) regress ego-velocity and eliminate its influence in the data representation to obtain ego-velocity invariance.

Ego-velocity estimation for invalid points removal. In 4D radar scans, there is a proportion of invalid points, such as noise points caused by multipath aliasing or points from dynamic objects. They may degrade the radar scan representation. Therefore, we introduce data preprocessing [9] to denoise and remove dynamic points.

Given a set of N points $\{(\boldsymbol{p}_i, v_i^d)\}_{i=1}^N$ with Cartesian position $\boldsymbol{p}_i$ and radial relative velocity v_i^d, as in Eq. 1, the radial relative velocity v_i^d of each point is formally the product of its velocity relative to the radar $\boldsymbol{v}_i^r{=}\left[v_{x,i}^r, v_{y,i}^r, v_{z,i}^r\right]^T$ and its unit position vector $\hat{\boldsymbol{p}}_i{=}\frac{\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|}{=}[\hat{p}_{x,i}, \hat{p}_{y,i}, \hat{p}_{z,i}]^T$.

$$v_i^d = \hat{\boldsymbol{p}}_i^\intercal \boldsymbol{v}_i^r = \hat{p}_{x,i} v_{x,i}^r + \hat{p}_{y,i} v_{y,i}^r + \hat{p}_{z,i} v_{z,i}^r \quad (1)$$

Let $\boldsymbol{v}^e{=}\left[v_x^e, v_y^e, v_z^e\right]^T$ denote the ego-velocity of the 4D radar. The relative velocities of the static points to the radar are the same as $\boldsymbol{v}^r{=}{-}\boldsymbol{v}^e{=}\left[-v_x^e, -v_y^e, -v_z^e\right]^T$. Assuming N measured points are static and applying Eq. 1 to them respectively, a system of N linear equations with three variables can be expressed as matrix notation in Eq. 2 and Eq. 3:

$$\boldsymbol{v}^d = [\hat{\boldsymbol{p}}_i, \ldots, \hat{\boldsymbol{p}}_N]^\intercal \, \boldsymbol{v}_i^r = \hat{\boldsymbol{P}}\boldsymbol{v}^r = -\hat{\boldsymbol{P}}\boldsymbol{v}^e \tag{2}$$

$$\begin{bmatrix} v_1^d \\ \vdots \\ v_N^d \end{bmatrix} = \begin{bmatrix} \hat{p}_{x,1} & \hat{p}_{y,1} & \hat{p}_{z,1} \\ \vdots & \vdots & \vdots \\ \hat{p}_{x,N} & \hat{p}_{y,N} & \hat{p}_{z,N} \end{bmatrix} \begin{bmatrix} -v_x^e \\ -v_y^e \\ -v_z^e \end{bmatrix} \tag{3}$$

If only at least three static points have different unit position vector $\hat{\boldsymbol{p}}$, the matrix $\hat{\boldsymbol{P}} \in \mathbb{R}^{N\times 3}$ is full rank. According to the least squares method [27], the linear equations in Eq. 2 has the optimal analytical solution as Eq. 4, through which the ego-velocity $\boldsymbol{v}^e$ can be estimated.

$$\boldsymbol{v}^e = -\left(\hat{\boldsymbol{P}}^\intercal \hat{\boldsymbol{P}}\right)^{-1} \hat{\boldsymbol{P}}^\intercal \boldsymbol{v}^d \tag{4}$$

The necessary prerequisite for accurately regressing the analytical ego-velocity is the measured points are all static. To this end, RANSAC [9, 11] is employed to get rid of dynamic outliers through iterative sampling. The RANSAC inlier points $\boldsymbol{P'}$ obtained with the optimal $\boldsymbol{v}^e$ are most likely to be static and retained for subsequent processing. By this means, the velocity attribute of the 4D radar scan is leveraged to regress the radar ego-velocity $\boldsymbol{v}^e$ and refine the original data with RANSAC outlier points removed.

Preprocessing for 4D point representation. Considering the varying ego-velocities of a 4D radar when visiting the same place, directly incorporating radial relative velocity $\boldsymbol{v}^d$ into feature embedding may introduce bias, causing the model to learn harmful tricks. To eliminate the influence of ego-velocity on point representation, we decouple the speed and direction in the radial relative velocity $\boldsymbol{v}^d$ and the ego-velocity $\boldsymbol{v}^e$. Since their speeds are both determined by the motion of the radar rather than the static scene, we propose to combine only their directions into a new attribute representing the azimuth angle of the point relative to the direction of the radar motion.

Specifically, the radial velocity direction of a point can be expressed as the unit position vector $\hat{\boldsymbol{p}}_i$. The ego-velocity direction is formulated as $\hat{\boldsymbol{v}}^e = \frac{\boldsymbol{v}^e}{\|\boldsymbol{v}^e\|} = \left[\hat{v}_x^e, \hat{v}_y^e, \hat{v}_z^e\right]^T$. As in Eq. 5, the new attribute s that characterizes relative azimuth angle is defined as the cosine similarity (dot product) of the two unit direction vectors $\hat{\boldsymbol{p}}_i$ and $\hat{\boldsymbol{v}}^e$.

$$s_i = \langle \hat{\boldsymbol{p}}_i, \hat{\boldsymbol{v}}^e \rangle = \hat{\boldsymbol{p}}_i^\intercal \hat{\boldsymbol{v}}^e = \hat{p}_{x,i}\hat{v}_x^e + \hat{p}_{y,i}\hat{v}_y^e + \hat{p}_{z,i}\hat{v}_z^e \tag{5}$$

Essentially, the radial relative velocity attribute v^d of the original point cloud $\mathcal{PC}$=$\{(\boldsymbol{p}_i, v_i^d, I_i)\}$ is utilized to remove dynamic points and generate a new attribute s of the refined 4D point cloud $\mathcal{PC}'$=$\{(\boldsymbol{p}_i, s_i, I_i)\}$. That is, the relative azimuth angle s independent of radar motion. With the normalized intensity attribute I of point clouds, the refined point cloud $\mathcal{PC}'$ is the 4D point representation for subsequent feature embedding and descriptor generation.

3.2. MinkLoc4D Backbone for Feature Extraction

While raw point processing lacks the capacity to capture local geometric patterns, voxelization and 3D sparse convolution [20, 42, 52] have proven superior in the 3DLPR task. The State-Of-The-Arts (SOTAs) MinkLoc3Dv2 [20] uses the binary voxelized representation to indicate the presence of 3D points and MinkLoc3D-SI [52] associates each point with a LiDAR intensity feature. Inspired by them, we propose the first feature extraction backbone for 4DRPR, MinkLoc4D. Based on voxelized points with numerical features of intensity and relative azimuth angle, it exploits the multi-modal information of geometry, intensity, and velocity from 4D radar scans via 3D sparse convolutions.

Specifically, the refined point cloud $\mathcal{PC}'$=$\{(\boldsymbol{p}_i, s_i, I_i)\}$ is first quantized into sparse tensors $\tilde{\mathcal{PC}}$=$\{V_i : (\tilde{\boldsymbol{p}}_i, [s_i, I_i])\}$. Each V_i represents a non-empty voxel with 3D coordinates $\tilde{\boldsymbol{p}}_i$ and two-dimensional features $[s_i, I_i]$. Then the voxelized point cloud $\tilde{\mathcal{PC}}$ is fed to a Feature Pyramid Network (FPN) for local feature extraction. As in Fig. 2, the FPN consists of two trunks bridged at different scales by lateral convolutional blocks (LConv2∼LConv4). The bottom-up trunk contains four convolutional blocks (Conv0∼Conv4) with increasing receptive fields. Incorporating Efficient Channel Attention (ECA) [39], they produce sparse 3D feature maps with decreasing spatial resolution. The top-down trunk contains two transposed convolutional blocks (TConv3 and TConv4). They generate upsampled feature maps, which are then merged with the skipped features from the corresponding layers in the bottom-up trunk via lateral convolutions. Overall, the FPN takes as input the two-dimensional features of the voxelized point cloud, and generates sparse local features $F_b \in \mathbb{R}^{N_b\times 256}$ for non-empty voxels.

3.3. Transformer for Feature Enhancement

Due to the sparseness, unevenness, and disorder of point clouds, there may also be correlations between points that are far apart. The MinkLoc4D backbone only extracts local patterns of voxelized points but lacks the ability to mine their long-range dependency. This hinders better local feature representation. Therefore, we modify a linear Transformer [34] as a complementary to the MinkLoc4D. It enhances local features by aggregating their global context.

The architecture of the Transformer module is shown in Fig. 3. It first upsamples the sparse voxel features F_b using an interpolation layer [6], generating the same number of features $F \in \mathbb{R}^{N\times 256}$ at coordinate $\tilde{P} \in \mathbb{R}^{N\times 3}$ as the original input voxels of the point cloud. Then the interpolated features are forwarded to the multi-head self-attention layer. Specifically, through Eq. 6, the input features $F \in \mathbb{R}^{N\times 256}$ are first projected to H heads of query, key, and value vectors $\{(Q_h \in \mathbb{R}^{N\times S}, K_h \in \mathbb{R}^{N\times S}, V_h \in \mathbb{R}^{N\times L})\}_{h=1}^{H}$ by the corresponding projection matrix $W_{Q_h} \in \mathbb{R}^{S\times 256}$,

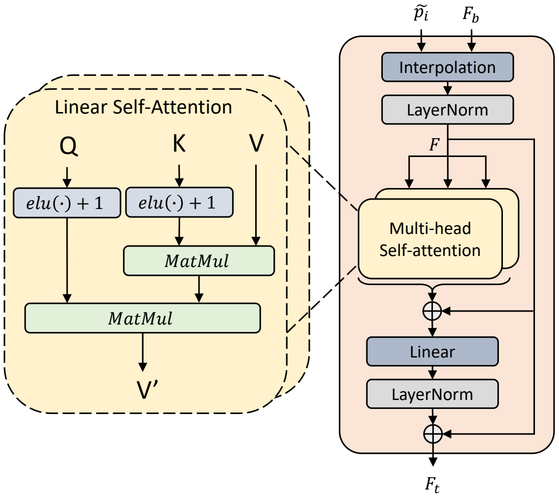

Figure 3. Schematic of the Transformer for local feature enhancement, where interpolation and linear self-attention are employed.

$W_{K_h} \in \mathbb{R}^{S\times 256}$, and $W_{V_h} \in \mathbb{R}^{L\times 256}$.

$$Q_h = FW_{Q_h}^T, \quad K_h = FW_{K_h}^T, \quad V_h = FW_{V_h}^T \tag{6}$$

Then, through Eq. 7, the output vector $V'_{h,i}$ of the self-attention layer is generated by the weighted sum of the value vectors $\{V_{h,j}\}$. The weights are determined by the similarity scores between the query $Q_{h,i}$ and the keys $\{K_{h,j}\}$.

$$V'_{h,i} = \frac{\sum_j sim(Q_{h,i}, K_{h,j})V_{h,j}}{\sum_j sim(Q_{h,i}, K_{h,j})} \tag{7}$$

Since a large number of local features will result in a big similarity matrix that is computationally inefficient, we follow Linear Transformer [16, 34] to substitute the original exponential kernel with an alternative $sim(Q, K)$ $=\phi(Q)\phi(K)^T$, where $\phi() = elu() + 1$. According to the associativity property of matrix products, $\phi(K)^T V$ can be carried out first as in Eq. 8, so that the largest dimension N can be eliminated to reduce computational complexity.

$$V'_h = \frac{\left(\phi(Q_h)\phi(K_h)^T\right) V_h}{\phi(Q_h)\phi(K_h)^T} = \frac{\phi(Q_h)\left(\phi(K_h)^T V_h\right)}{\phi(Q_h)\phi(K_h)^T} \tag{8}$$

$$V'_{h,i} = \frac{\phi(Q_{h,i})^T \sum_{j=1}^N \phi(K_{h,j})V_{h,j}^T}{\phi(Q_{h,i})^T \sum_{j=1}^N \phi(K_{h,j})} \tag{9}$$

$$F_t = F + \mathrm{LN}(\mathrm{Linear}(\mathrm{Concat}(\{V'_h\}_{h=1}^H) - F)) \tag{10}$$

As in Eq. 10, firstly, the residuals between the concatenated H heads of vectors $\{V'_h\}$ and the original local features F are mapped and normalized. Linear and LN denote $Linear$ and $LayerNorm$ layers. Finally, the enhanced local features F_t are formulated by merging the original local features F and the attention residuals. Overall, the Transformer module produces the enhanced local features F_t whose shape is the same as the original local features F. As a complementary to MinkLoc4D that extracts local patterns, the Transformer exploits the long-range correlation of points to aggregate their global context, regardless of the total number of points and their coordinates.

3.4. Point Cloud Descriptor and Network Training

Generalized Mean (GeM) pooling [32] has shown to be superior when coupled with classification learning [3] or large batch ranking [20]. Therefore, to generate the final 4D point cloud descriptor F_g of the proposed TransLoc4D, we employ GeM pooling to aggregate the enhanced local features F_t, as Eq. 11.

$$F_g = \left[\left(\frac{1}{|F_t|}\sum_{f\in F_t} f^p\right)^{\frac{1}{p}}\right]_{c=1\ldots 256} \tag{11}$$

c is the channel index, $|F_t|$ represents the total number of the local features, and p is a trainable parameter. It can be noted that the feature enhancement and aggregation in Sec. 3.3 and Sec. 3.4 are both independent of the positions of points. The dimensionality of the final descriptor F_g is also independent of the total number of points $|F_t|$. These make the TransLoc4D descriptor robust to point cloud disorder and scalable to point clouds of different scales.

To train our TransLoc4D network, we adopt the Truncated Smooth-AP (TSAP) loss [20]. It prompts the network to maximize the average precision of the top-k positive candidates through data-driven learning. Formally, the smooth average precision is defined in Eq. 12.

$$AP_q = \frac{1}{|P_q|}\sum_{i\in P_q}\frac{1+\sum_{j\in P_q, j\neq i}\mathcal{G}(d(q,i)-d(q,j))}{1+\sum_{j\in \Omega, j\neq i}\mathcal{G}(d(q,i)-d(q,j))} \tag{12}$$

$$\mathcal{L}_{TSAP}(\Omega) = \frac{1}{m}\sum_{q=1}^{m}(1-AP_q) \tag{13}$$

Among a batch of samples Ω with m queries, P_q is a set of k positives with the minimum descriptor similarity to the q^{th} query. $\mathcal{G}(x) = (1+\exp(-x/\tau))^{-1}$ denotes the differentiable approximation of the indicator function, in which the temperature constant τ controls the sharpness of the approximation. $d(q, i)$ is the Euclidean distance between the q^{th} query point cloud and the i^{th} reference point cloud in the descriptor space. Then the TSAP loss $\mathcal{L}_{TSAP}$ of a batch is formulated as Eq. 13. To enable large-batch training for optimal performance, multistaged backpropagation [33] is applied to minimize $\mathcal{L}_{TSAP}$ during training.

4. Experiments

4.1. Benchmark Datasets

So far, there has been no open-sourced dataset specifically for the 4DRPR task. Therefore, we create two

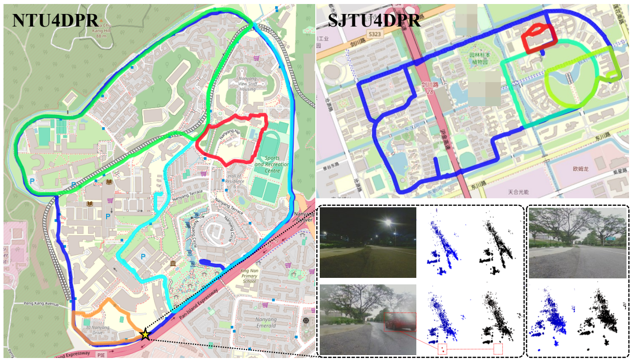

Figure 4. Overview of NTU4DPR (left) and SJTU4DPR (top right) datasets. Examples are given for nighttime and rainy days.

benchmark datasets for 4DRPR based on the open-source NTU4DRadLM [49] and SJTU4D [24] datasets, as well as supplementary data that we newly collected at NTU.

NTU4DRadLM [49] is a multi-modal dataset encompassing approximately $17.6km$ data from six sensors: 4D radar, thermal camera, IMU, 3D LiDAR, visual camera, and RTK GPS. Proposed for SLAM, it provides accurate ground truth odometry and intentionally crafted loop closures.

SJTU4D [24] comprisess data from calibrated and synchronized LiDAR and 4D radar. It covers varied environments, such as an industrial zone and a university campus.

Both NTU4DRadLM and SJTU4D datasets were collected by the mounted sensors on a mobile vehicle. The captured data was recorded in real time. Each 4D radar reading (including three-dimensional position, radial relative velocity, and intensity of reflection) is aligned with a GPS tag based on timestamps and stored as a frame. Since NTU4DRadLM only contains three loops with repetitive trajectories utilizable for 4DRPR, we refine them as NTU4DPR with train_query, train_database, test_query, and test_database splits. NTU4DPR-NYL and NTU4DPR-SRC (orange and red loops in the left of Fig. 4) are two special subsets newly collected by a robot operating on sidewalks instead of main roads. We sample them as test sets, including query splits at nighttime and on rainy day. Similarly, we restructure SJTU4D into SJTU4DPR, including two subsets for evaluation: SJTU4DPR-TestA and SJTU4DPR-TestB. Tab. 1 shows the details of the two generated datasets.

4.2. Evaluation Methodology

Evaluation metric. We follow the standard place recognition evaluation protocol [2, 3, 13] to evaluate our method. A query 4D radar point cloud is considered to be correctly localized if at least one of the top N retrieved candidates is within 25 meters of the query geolocation. The performance of the model is measured by $Recall@N$, which is the percentage of correctly identified queries (recall) when given a specific number $N \in \{1, 5, 10\}$ of candidates.

Table 1. Description of Benchmark Datasets

Dataset	Attributes	Splits	Split Size
NTU4DPR	car, main road 3 trajectories 36,026 frames	train_query	7,620
		train_database	10,000
		test_query	7,002
		test_database	10,839
	robot, sidewalk 5 trajectories 55,699 frames	nyl_night_q	7,283
		nyl_rain_q	6,085
		nyl_cloudy_db	7,410
		src_night_q	9,069
		src_daytime_db	8,061
SJTU4DPR	car, main road 4 trajectories 39,135 frames	test_a_query	7,634
		test_a_database	7,500
		test_b_query	6,501
		test_b_database	2,500

Implementation details. In this work, all experiments are performed using PyTorch [28] deep learning framework on an Nvidia 2080Ti GPU. During training, various transformations of data augmentation are randomly applied to avoid overfitting, including jitter, rotation, translation, flipping, and removal of partial points or blocks. The models are trained from scratch through the pipeline in Sec. 3.4. An AdamW [25] optimizer is used to minimize the TSAP loss in Eq. 13 for 150 epochs with a learning rate of 0.001.

4.3. Ablation Study

As elaborated in Sec. 3, we propose the TransLoc4D as the first solution to 4DRPR with four innovations: point cloud **R**efinement based on ego-velocity regression and RANSAC filtering; converting radial relative **V**elocity into a new attribute representing velocity azimuth angle; Incorporating geometric, velocity and **I**ntensity attribute into 4D radar point representation and feature embedding; Integrating a **T**ransformer module for feature enhancement.

To analyze the individual contribution of each component in our method, we compare the TransLoc4D variants that progressively apply different components. We set the plain TransLoc4D with all components disabled as the basic model. It only consists of MinkLoc4D backbone and GeM pooling. On the basis of the plain TransLoc4D, we use additional abbreviations to denote the application of point cloud refinement (**-R**), velocity azimuth angle attribute (**-V**), intensity attribute (**-I**), and Transformer enhancement (**-T**).

As shown in Tab. 2, the plain TransLoc4D establishes a decent baseline on the evaluation sets, indicating voxelized representation is also feasible for the 4DRPR task. The significantly better performance of TransLoc4D-R than TransLoc4D demonstrates the effectiveness of our point cloud refinement (R). It prevents dynamic and noise points from interfering with scene description. Moreover, replacing the voxel representation with a numerical feature representation (V) brings stable improvements to TransLoc4D-R

Table 2. Ablation studies to evaluate the individual contributions of point cloud **R**efinement, 4D point cloud representation with numerical features of **V**elocity and **I**ntensity attributes, and the **T**ransformer module for feature enhancement.

Method	Components				Trained on NTU4DPR-Train											
					NYL-Night			NYL-Rain			SRC-Night			SJTU4DPR-TestA		
	R	V	I	T	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10
TransLoc4D	×	×	×	×	93.2	95.4	96.3	75.8	86.0	89.3	86.2	92.6	96.3	84.6	92.8	**94.3**
TransLoc4D-R	√	×	×	×	96.6	98.2	98.7	81.0	88.4	91.5	89.0	94.5	96.4	88.6	93.2	94.0
TransLoc4D-R-V	√	√	×	×	95.7	97.8	98.6	83.3	89.5	93.4	93.6	**97.3**	**98.3**	89.5	**93.2**	94.1
TransLoc4D-R-VI	√	√	√	×	96.8	98.3	98.7	82.5	89.7	92.1	94.4	96.9	97.9	89.0	92.4	93.3
TransLoc4D-R-VI-T	√	√	√	√	**97.1**	**98.4**	**98.7**	**86.8**	**91.8**	**94.0**	**94.5**	97.0	98.0	**90.8**	92.9	93.4

Table 3. Comparison with SOTA methods for LiDAR/radar-based place recognition on 4D radar datasets. Benchmark models are either pre-trained (*) on their default training set or fine-tuned on NTU4DPR.

Method	Training Set	Evaluation Set														
		NYL-Night			NYL-Rain			SRC-Night			SJTU4DPR-TestA			SJTU4DPR-TestB		
		r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10	r@1	r@5	r@10
In. Scan Context [38]	N/A	87.2	92.8	94.7	69.1	80.3	84.3	68.1	82.2	86.9	67.9	79.8	83.7	78.0	86.5	90.2
PTC-Net-L* [6]	Oxford	93.8	96.8	98.1	67.3	78.8	83.5	83.3	92.1	94.6	67.8	83.6	87.5	79.6	87.3	89.2
MinkLoc3Dv2* [20]	Oxford	93.6	97.0	98.0	74.4	85.8	89.6	86.9	94.5	96.5	86.9	**93.7**	**94.9**	85.4	88.2	89.2
TransLoc3D [42]	NTU4DPR	91.6	95.4	96.9	76.2	85.9	89.2	86.4	93.7	96.0	75.6	89.2	92.2	79.3	86.0	87.5
AutoPlace [5]	NTU4DPR	92.9	96.4	97.5	80.9	89.7	92.2	86.9	94.7	97.1	80.5	88.8	91.2	80.2	84.9	86.5
MinkLoc3Dv2 [20]	NTU4DPR	96.6	98.2	98.7	81.0	88.4	91.5	89.0	94.5	96.4	88.6	93.2	94.0	85.8	87.9	88.9
PTC-Net-L [6]	NTU4DPR	96.6	**98.6**	**99.1**	**87.6**	**92.7**	**94.3**	94.5	**98.1**	96.9	79.8	89.6	91.4	80.3	85.3	86.8
TransLoc4D (ours)	NTU4DPR	**97.1**	98.4	98.7	86.8	91.8	94.0	**94.5**	97.0	**98.0**	**90.8**	92.9	93.4	**85.9**	**88.7**	**90.5**

on most datasets, which validates the introduced new attribute of velocity azimuth angle. Additionally incorporating the intensity into feature embedding, TransLoc4D-R-VI surpasses TransLoc4D-R-V on nighttime subsets, but lags behind on the rainy and SJTU subsets. This reflects the susceptibility of the intensity to rainfall and the cross-domain environments. Nonetheless, comparing TransLoc4D-R-VI with TransLoc4D-R, stable improvements can be observed on all datasets, especially on SRC-Night with an increase of 5.4%. This demonstrates the rationality of characterizing velocity and intensity attributes as numerical features.

Further enabling the Transformer (-T) leads to another large performance increase on all datasets. The advantage on the rainy subset is particularly prominent. It indicates that the global context also contains valid information that is crucial to the task and is more robust to data domain shift. Overall, ablation studies in Tab. 2 prove the effectiveness of each module and that their advantages can be accumulated. Compared to the baseline TransLoc4D, our best model (TransLoc4D-R-VI-T) shows an overall performance advantage of about 10% on benchmark datasets.

4.4. More Results and Discussion

Comparisons with adapted SOTAs. Since there is currently no method specifically proposed for 4D radar place recognition, to further validate our TransLoc4D, we adapt the latest SOTA methods for 3D LiDAR or 3D radar-based place recognition to the 4DRPR task. The comparative models include Intensity Scan Context [38], AutoPlace [5], MinLoc3Dv2 [20], TransLoc3D [42], and PTC-Net [6].

Intensity Scan Context [38] constructs regional features based on bird's-eye view partitioning of polar coordinate images. **AutoPlace** [5] converts radar scans to 2D binary images and employs a network to encode spatial-temporal features. **MinkLoc3Dv2** [17] can be considered as a special case of TransLoc4D, with only MinkLoc4D backbone and GeM pooling. **TransLoc3D** [42] contains a different backbone architecture from MinkLoc3Dv2 and ours, and adopts NetVLAD pooling instead of GeM. **PTC-Net** [6] introduces a novel point-wise Transformer to compensate for the information loss caused by voxelization, achieving SOTA performance on 3D LiDAR place recognition task.

For fair comparisons, all differentiable models take our refined 4D point clouds with invalid points removed as input, and are trained on NTU4DPR using the same pipeline described in Sec. 3.4. To allow models pre-trained on 3D LiDAR datasets to be directly evaluated on 4D radar data, we create a new copy of 4D radar datasets that maintains the same coordinate format as 3D LiDAR datasets (Oxford), where point coordinates are normalized and quantized with a step size of 0.01.

In Tab. 3, Intensity Scan Context performs mediocre on 4D radar datasets, which reflects the limitations of handcrafted descriptors. PTC-Net and MinkLoc3Dv2 pre-trained on the 3D LiDAR dataset generalize well on the 4D radar datasets, but the performance degradation caused by domain shift is noticeable. It can be attributed to the sparse, noisy, non-panoramic characteristics of 4D radar

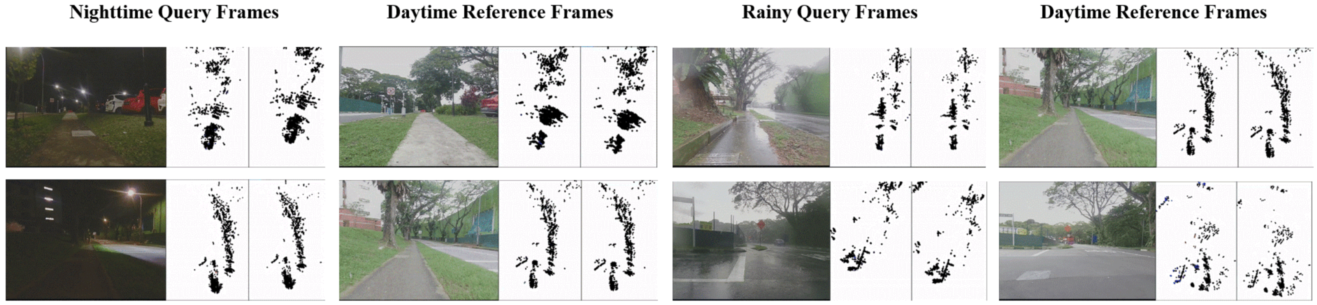

Figure 5. Challenging query frames and the reference frames retrieved by TransLoc4D. Consecutive frames are to better visualize how TransLoc4D robustly suppress noise and dynamic points (non-black) from the original point clouds (2^{nd} column in each subfigure). When images (1^{st} column) exhibit drastic appearance differences, refined 4D point clouds (3^{rd} column) demonstrate stable similarities.

data. Nevertheless, the pre-trained MincLoc3Dv2 far exceeds the hand-crafted baseline, illustrating the power of data-driven learning. The pre-trained PTC-Net-L is inferior to MinkLoc3Dv2 in some cases, proving that some specific patterns learned by a SOTA model in 3D point clouds may not be applicable to 4D radar scans. Therefore, fine-tuning on 4D radar datasets is necessary. As expected, all evaluated differentiable models, including AutoPlace, TransLoc3D, MinkLoc3Dv2, and PTC-Net-L, adapt and perform better on 4D radar datasets after training. Due to different architectures, they exhibit large performance deviations in cross-domain evaluations. While MinkLoc3Dv2 leads AutoPlace and TransLoc3D on all datasets, TransLoc4D consistently outperforms MinkLoc3Dv2, especially on the challenging NYL-Rain and SRC-Night by more than 6% and 5%. Both interpolating sparse features to the number of input voxels, PTC-Net-L achieves slightly better results than TransLoc4D on NTU splits but falls behind significantly on SJTU splits. The better generalization performance on the sparser dataset SJTU4DPR indicates that our TransLoc4D is able to capture more general features and patterns that lead to better cross-domain robustness.

Qualitative results. 4 challenging query frames from NTU4DPR-NYL and their top 1 retrieved reference frames by our TransLoc4D are shown in Fig. 5. Five consecutive frames before and after the query and the retrieved reference are made into GIF format to better visualize the robustness of our method to dynamic objects and environmental changes. When images (1^{st} column) exhibit large appearance differences due to lighting and weather, refined 4D point clouds (3^{rd} column) demonstrate stable similarities. Noise and dynamic points (non-black) in the original point clouds (2^{nd} column) are all suppressed in the refined point cloud and subsequent feature embedding, bringing robustness to our TransLoc4D descriptor. Two examples in Fig. 6 illustrate the filtering of dynamic and noise points based on radial relative velocity through decomposition steps. [1]

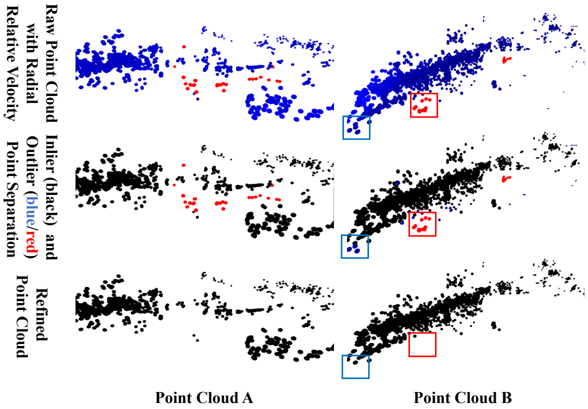

Figure 6. Examples of preprocessing 4D point clouds. Red, black, and blue represent positive, zero, and negative speed. Based on relative velocity (first row), absolute motion (second row) can be regressed to filter outlier points from dynamic objects (third row).

5. Conclusions

In this work, we propose the first end-to-end encoding architecture, TransLoc4D, for 4D radar place recognition. First, point cloud preprocessing and a novel 4D representation are presented. On this basis, the MinkLoc4D backbone is proposed to extract features from multi-modal characteristics of 4D radar scans. Then, a linear Transformer is introduced to capture the global context to enhance feature representation, followed by a GeM pooling to generate the final 4D point cloud descriptor. To validate our proposed method, we construct two datasets and set up benchmarks for 4D radar place recognition. Extensive experiments demonstrate the feasibility of TransLoc4D and its robustness against dynamic and adverse environments.

Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund-Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

[1] Due to the page limit, we provide more experiment results and visualizations in the supplementary material.

References

[1] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964, 2019. 1

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016. 2, 6

[3] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4868–4878, 2022. 1, 2, 5, 6

[4] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11046–11056, 2023. 2

[5] Kaiwen Cai, Bing Wang, and Chris Xiaoxuan Lu. Autoplace: Robust place recognition with single-chip automotive radar. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2222–2228, 2022. 3, 7

[6] Lineng Chen, Huan Wang, Hui Kong, Wankou Yang, and Mingwu Ren. Ptc-net: Point-wise transformer with sparse convolution network for place recognition. *IEEE Robotics and Automation Letters*, 8(6):3414–3421, 2023. 2, 4, 7

[7] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. *arXiv preprint arXiv:2312.09866*, 2023. 3

[8] Tianchen Deng, Hongle Xie, Jingchuan Wang, and Weidong Chen. Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction. *IEEE Robotics & Automation Magazine*, 30(1):36–49, 2023. 1

[9] Christopher Doer and Gert F. Trommer. An ekf based approach to radar inertial odometry. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 152–159, 2020. 3, 4

[10] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 551–560, Jun. 2022. 2

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. 4

[12] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV 2020*, pages 369–386, 2020. 2

[13] Peiyu Guan, Zhiqiang Cao, Junzhi Yu, Min Tan, and Shuo Wang. Visual place recognition via a multi-task learning method with attentive feature aggregation. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1, 2022. 2, 6

[14] Jiadong Guo, Paulo Vinicius Koerich Borges, Chanoh Park, and Abel Gawel. Local descriptor for robust place recognition using lidar intensity. *IEEE Robotics and Automation Letters*, 4:1470–1477, 2018. 2

[15] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6078–6087, 2021. 2

[16] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 5

[17] Giseop Kim and Ayoung Kim. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809, 2018. 3, 7

[18] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260, 2017. 2

[19] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1789–1798, 2021. 2

[20] Jacek Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3699–3705, 2022. 2, 4, 5, 7

[21] Jacek Komorowski, Monika Wysoczańska, and Tomasz Trzcinski. Minkloc++: Lidar and monocular image fusion for place recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 1

[22] Haowen Lai, Peng Yin, and Sebastian Scherer. Adafusion: Visual-lidar fusion with adaptive weights for place recognition. *IEEE Robotics and Automation Letters*, 7(4):12038–12045, 2022. 1

[23] Heshan Li, Guohao Peng, Jun Zhang, Sriram Vaikundam, and Danwei Wang. Adaptseqvpr: An adaptive sequence-based visual place recognition pipeline. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3284–3289, 2023. 2

[24] Xingyi Li, Han Zhang, and Weidong Chen. 4d radar-based pose graph slam with ego-velocity pre-integration factor. *IEEE Robotics and Automation Letters*, 8(8):5124–5131, 2023. 1, 2, 3, 6

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[26] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition.

In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[27] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006. 4

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019. 6

[29] Guohao Peng, Yufeng Yue, Jun Zhang, Zhenyu Wu, Xiaoyu Tang, and Danwei W. Wang. Semantic reinforced attention learning for visual place recognition. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13415–13422, 2021. 2

[30] Guohao Peng, Jun Zhang, Heshan Li, and Danwei Wang. Attentional pyramid pooling of salient visual residuals for place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 885–894, October 2021. 2

[31] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, 2010. 2

[32] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 2, 5

[33] Jerome Revaud, Jon Almazan, Rafael Rezende, and Cesar De Souza. Learning with average precision: Training image retrieval with a listwise loss. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5106–5115, 2019. 5

[34] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8918–8927, 2021. 4, 5

[35] Ştefan Săftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4358–4364, 2020. 3

[36] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018. 1, 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, 2017. 2

[38] Han Wang, Chen Wang, and Lihua Xie. Intensity scan context: Coding intensity and geometry relations for loop closure detection. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2095–2101, 2020. 2, 3, 7

[39] Qilong Wang, Banggu Wu, Peng Fei Zhu, P. Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11531–11539, 2019. 4

[40] Hongle Xie, Tianchen Deng, Jingchuan Wang, and Weidong Chen. Robust incremental long-term visual topological localization in changing environments. *IEEE Transactions on Instrumentation and Measurement*, 72:1–14, 2023. 1

[41] Hongle Xie, Tianchen Deng, Jingchuan Wang, and Weidong Chen. Angular tracking consistency guided fast feature association for visual-inertial slam. *IEEE Transactions on Instrumentation and Measurement*, 73:1–14, 2024. 1

[42] Tianhan Xu, Yuanchen Guo, Yu-Kun Lai, and Song-Hai Zhang. Transloc3d : Point cloud based large-scale place recognition using adaptive receptive fields. *Communications in Information and Systems*, 23:57–83, 2021. 2, 4, 7

[43] Artem Babenko Yandex and Victor Lempitsky. Aggregating local deep features for image retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015. 2

[44] Peng Yin, Yuqing He, Na Liu, and Jianda Han. Condition directed multi-domain adversarial learning for loop closure detection. *ArXiv*, abs/1711.07657, 2017. 1

[45] Peng Yin, Lingyun Xu, Xueqian Li, Chen Yin, Yingli Li, Rangaprasad Arun Srivatsan, Lu Li, Jianmin Ji, and Yuqing He. A multi-domain feature learning method for visual place recognition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 319–324, 2019. 1

[46] Peng Yin, Lingyun Xu, Ji Zhang, Howie Choset, and Sebastian Scherer. i3dLoc: Image-to-range Cross-domain Localization Robust to Inconsistent Environmental Conditions. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. 1

[47] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE transactions on neural networks and learning systems*, 31(2):661–674, 2019. 2

[48] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2019. 2

[49] Jun Zhang, Huayang Zhuge, Yiyao Liu, Guohao Peng, Zhenyu Wu, Haoyuan Zhang, Qiyang Lyu, Heshan Li, Chunyang Zhao, Dogan Kircali, Sanat Mharolkar, Xun Yang, Su Yi, Yuanzhe Wang, and Danwei Wang. Ntu4dradlm: 4d radar-centric multi-modal dataset for localization and mapping. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4291–4296, 2023. 2, 3, 6

[50] Jun Zhang, Huayang Zhuge, Zhenyu Wu, Guohao Peng, Mingxing Wen, Yiyao Liu, and Danwei Wang. 4dradarslam: A 4d imaging radar slam system for large-scale environ-

ments based on pose graph optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8333–8340, 2023. 1, 3

[51] Lianqing Zheng, Zhixiong Ma, Xichan Zhu, Bin Tan, Sen Li, Kai Long, Weiqi Sun, Sihan Chen, Lu Zhang, Mengyue Wan, Libo Huang, and Jie Bai. Tj4dradset: A 4d radar dataset for autonomous driving. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 493–498, 2022. 1

[52] Kamil Zywanowski, Adam Banaszczyk, Michał R. Nowicki, and Jacek Komorowski. Minkloc3d-si: 3d lidar place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters*, PP:1–1, 2021. 2, 4