Received: 17 July 2024 | Revised: 6 December 2024 | Accepted: 14 January 2025

IET Radar, Sonar & Navigation

DOI: 10.1049/rsn2.70002

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Bayesian Radar Cosplace: Directly estimating location uncertainty in radar place recognition

Suyash Agarwal | **Jianhao Yuan** | **Paul Newman** | **Daniele De Martini** | **Matthew Gadd**

Mobile Robotics Group, University of Oxford, Oxford, UK

Correspondence

Daniele De Martini.
Email: daniele@robots.ox.ac.uk

Funding information

Engineering and Physical Sciences Research Council, Grant/Award Number: EP/V000748/1

Abstract

Trust and explainability in localisation systems can be greatly helped by estimating a calibrated uncertainty. In this work, we argue for the first time that for this, it is best to express uncertainty in the location estimate *directly* rather than indirectly in the 'noisiness' or ambiguity of the data sample. Therefore, in this work, through a robust classification-based model, we not only identify the most probable place but also provide a measure of confidence or uncertainty associated with the prediction of the place itself—in contrast to existing approaches where uncertainty values are produced with the same dimension as the encoded feature. We specifically prove the utility of this new formulation on *CosPlace*, a state-of-the-art Geolocalisation system. Uncertainty is learnt by transforming *Cosplace* into an uncertainty-aware neural network. To validate the effectiveness of our approach, we conduct extensive experiments using the *Oxford Radar RobotCar Dataset*, where we find that the backbone features learnt in the uncertainty-aware setting result in better place recognition performance than vanilla *Cosplace*. Furthermore, by using it as a score to reject putative localisation results, we show that our uncertainty is well-calibrated to place recognition accuracy—more so than two existing systems in uncertainty-aware radar place recognition.

KEYWORDS

learning (artificial intelligence), measurement uncertainty, mobile robots, navigation, radar

1 | INTRODUCTION

Recognising places with radar typically entails transforming a real-time scan into a vector and comparing it with a database to identify if the vehicle is in a previously visited location. Radar is naturally resilient to changes in lighting or weather, yet place recognition using this technology can still be influenced by (a) changes in viewpoint, such as translation and rotation and (b) complex sensor artefacts, sensor anomalies, or 'noise'.

By tracking the probabilistic confidence level associated with each observation, we make the model aware of when its predictions are likely to be incorrect, thus pre-empting potential failure modes. If a particular location proves difficult to identify, measuring uncertainty gives the model the power to avoid making a decision which is likely to be incorrect [1]. We do this by rejecting queries which are uncertain.

As well as potentially improving overall model performance, uncertainty estimation is of particular interest in safety–critical situations such as autonomous driving, where it is invaluable to understand what makes models fail. This extends to any scenario where false positives have severe consequences. For example, when using place recognition to correct a global pose graph in SLAM, a single false positive localisation result can drastically reduce the quality of the estimated map and subsequent poses [2].

Trust is a fundamental design factor in the successful development and deployment of autonomous vehicles. The relationship between trust and uncertainty is significant, as expressing a degree of uncertainty in autonomous vehicle

decisions fosters trust in the underlying technology. This, in turn, encourages its adoption.

As place recognition is usually performed by a lookup across a vector database, with the vectorisation achieved by a learnt neural network, typical approaches to uncertainty-aware place recognition express confidence in this same output feature. However, this output feature is produced from each scan independently, and the neural network does not estimate the location itself (this being computed from downstream vector comparisons).

We argue, and prove in this work, that a stronger coupling of location estimate and uncertainty estimate is available when place recognition is treated as a classification problem—that is, the neural network is trained to classify a scan as belonging to one of many known places. For this, we exploit the recently developed *CosPlace* [3], which has been shown to exhibit state-of-the-art generalisation performance for this task. We adapt *CosPlace*, introducing novel uncertainty-aware training as well as query-rejection at test-time for introspective inference.

Figure 1 shows our system. Our primary contributions are as follows:

1. **Direct uncertainty-aware classification-based radar place recognition training**: Our learnt features are more robust for the task of place recognition, even when discarding the learnt uncertainty,
2. **Direct introspective, uncertainty-based query-rejection**: Our directly estimated location uncertainty is calibrated with place recognition accuracy, which we show by using it to reject localisation attempts if unlikely to succeed, and
3. **Discretisation of the Oxford Radar RobotCar Dataset**: Adapting the *Oxford Radar RobotCar Dataset* [4] for classification-based radar place recognition.

In the broader area of place recognition, our work introduces a novel approach by integrating—for the first time—uncertainty estimation into *CosPlace* [3], a classification-based place recognition framework which itself uses the large margin cosine loss (LMCL) [5]. In the specific area of radar for mobile autonomy, in contrast to prior methods relying on feature variability [2, 6], we *directly estimate uncertainty* within the classification process, offering improved integration and precision.

We evaluate our approach over a comprehensive and statistically robust cross-validation of 5 forays of the *Oxford Radar RobotCar Dataset* [4].

2 | RELATED WORK

2.1 | Radar place recognition

Place recognition can be regarded as a process of querying and retrieving, where a map is constructed using a dictionary of features which represent original radar scans. With each new query scan encoded into a query vector, the algorithm seeks to retrieve the most similar feature from the dictionary. The encoding function, often a neural network [7–10], must ensure that samples from nearby locations are proximate in the feature space and samples from far-flung locations are distant in the feature space.

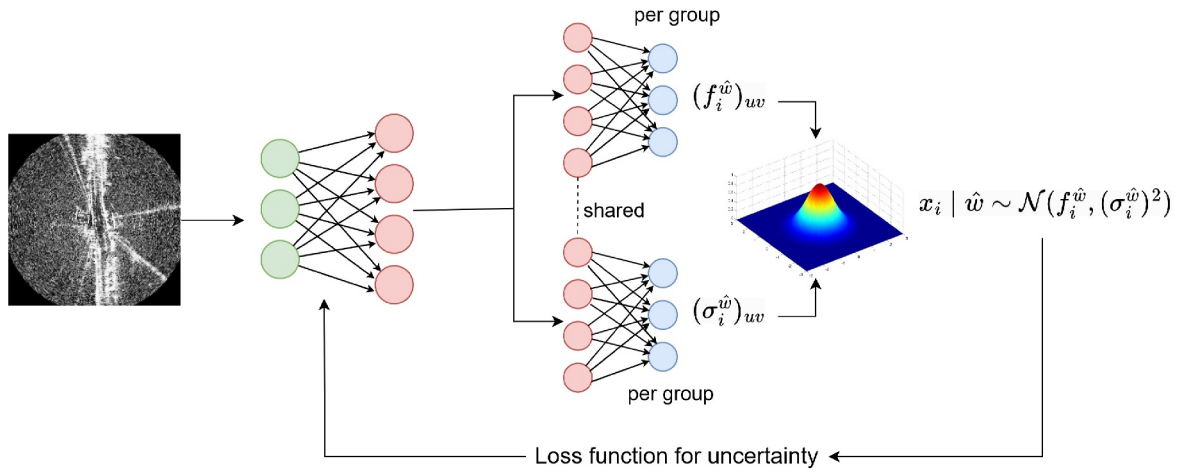


FIGURE 1 The setup for training our direct uncertainty-aware place recognition system. The model outputs two values, $f_i^{\hat{w}}$ and $\sigma_i^{\hat{w}}$, which parameterise a high-dimensional Gaussian distribution, $x_i|\hat{w} \sim \mathcal{N}\left(f_i^{\hat{w}}, \left(\sigma_i^{\hat{w}}\right)^2\right)$. This distribution has the same number of dimensions as the number of classes in the classification problem—which is equivalent to the number of locations in the discretised *Oxford Radar RobotCar Dataset* [4]. The variance, $\sigma_i^{\hat{w}}$, therefore contains the uncertainty in the classification of the location of the input scan. The index i corresponds to the i^{th} input radar scan that is fed into the network. *CosPlace* [3], the method we are extending in this work, in fact, trains multiple classification heads (rather than just a single classification head)—with each head containing a subset of the classes/locations in the world. For this reason, the two heads above are repeated for each group in the dataset (represented by the u, v indices). In our work, we therefore must also train multiple uncertainty heads. Overall, we train a model with multiple classification heads to learn to output a mean and variance for each location.

Approaches not based on deep learning, such as clustering [11], correlative scan matching [12], graph matching [13, 14], and *Scan Context* [15–18], have proven effective in utilising radar perception for place recognition and localisation.

In contrast, methods based on deep learning [19–25] have demonstrated impressive performance. These exploit rotational invariance and spatial-temporal consistency in radar scans and have shown remarkable effectiveness in comparison to more widely explored visual place recognition systems in the face of drastic changes in illumination and weather [26].

Furthermore, multi-modal methods incorporating additional modalities, such as LIDAR points, through direct joint learning [27] or point-cloud registration [28], are also explored to enhance radar-based recognition.

2.2 | Uncertainty-aware visual and LiDAR place recognition

Estimating uncertainty is vital for assessing the confidence of model predictions, particularly in safety–critical applications where the model must be able to reject predictions with low confidence.

Mason et al. [29] use consistency across an ensemble of networks to express uncertainty for LiDAR place recognition. While our approach does use multiple output heads to express uncertainty, these final layers are nevertheless quite small (do not contain excessive parameters) and, in contrast to ref. [29], use just a single backbone feature extractor (ensemble approaches requiring inference of a suite of networks).

In ref. [30], uncertainty is based on the consistency of the location of the retrieved samples—with the intuition that if the database candidates retrieved have locations that are quite scattered, the query must inherently contain a lot of uncertainty. This approach uses the map trajectories' locations to express uncertainty however. In contrast, our uncertainty is free of this ground truth requirement.

In ref. [31], a self-supervised visual place recognition method is presented which uses a calibration sequence to fine-tune the network with pose graph optimisation, also providing uncertainty estimates tailored to the new environment. Our approach, in contrast, does not require a SLAM backend.

STUN [2] learns a variance in the encoding of images, which, as we have argued above, is not strongly coupled to an estimate of location. Additionally, this method relies on a dynamic binning strategy that relies on a large batch of test samples to determine a meaningful uncertainty rejection threshold during inference. Specifically, this method only leverages the relative uncertainty within a batch, making it unsuitable for real-time place recognition deployment, where the entire test sequence is not known a priori.

Finally, Claxton et al. [32] use a multi-layer perceptron acting on query and matched features to *classify* the retrieved database candidate and their distance to the query as a genuine localisation result or not. This is an interesting approach but requires inference on retrieved map data, whereas our approach infers likely localisation success with purely the query as input.

2.3 | Uncertainty-aware radar localisation

While uncertainty estimation in radar-based place recognition tasks remains largely unexplored, efforts have been made to develop introspective radar systems for odometry.

Barnes et al. [33] interpret a softmax over correlation scores as the probability that each pose candidate is optimum. Our place recognition system does not need to perform fine-grained pose estimation of this kind, in contrast (and is incidentally less computationally expensive), and also avoids the well-known issue of softmax overconfidence [34].

Adolfsson et al. [35] introduce a radar system that introspectively verifies loop closure candidates by integrating multiple place-recognition techniques. These techniques include tightly coupled place similarity, odometry uncertainty search, creating loop descriptors from origin-shifted scans, and deferring loop selection until after verification. In contrast, our approach does not require a full SLAM implementation.

Conversely, Aldera et al. [36] concentrate solely on odometry and utilise inertial sensors to train an uncertainty classifier. In contrast to this work, our approach does not require an external sensor (an inertial measurement unit being used in ref. [36]).

In contrast to STUN [2], and specifically developed for place recognition in the radar domain, Yuan et al. [6] propose a static binning strategy that utilises VAE variance prediction. This approach compares the estimated uncertainty against a predefined prior distribution, enhancing the reliability of uncertainty estimation for real-time applications. Nevertheless, the approach of Yuan et al. [6] still expresses uncertainty in the sample rather than location, which we tackle in this work. Section 6 shows that our method outperforms Yuan et al. [6] and STUN [2] in terms of both baseline performance and introspective query rejection.

2.4 | Sparsity-based approaches

Conventional frequency-modulated continuous-wave (FMCW) radars suffer from interference due to their linear frequency sweeps over the entire bandwidth. Sparsity-based approaches such as [37–39] allow the signal to avoid continuous frequency sweeps, thereby mitigating interference. In our work, we avoided a sparsity-based approach because the class of radar we used presents dense images radar—that is, we could not modulate the frequency sweep pattern due to hardware constraints. However, sparse radar might be noisier due to reduced signal energy, lower sampling density, and increased reliance on reconstruction algorithms sensitive to noise. Therefore, sparse approaches may also benefit from an uncertainty-based system as we present in this paper.

3 | PRELIMINARIES

3.1 | Dataset discretisation

The method of choice for training the classifier in *CosPlace* [3] derives from a state-of-the-art face recognition method [5], using a loss function detailed below in Section 3.2. However, this requires the training dataset to be divided into classes. Generally the place recognition ground truth is continuous, for example, GPS coordinates, but it can be discretised by splitting the area into square geographical cells with the set side length. This basic process of GPS discretisation is illustrated in Figure 2a–2c and 3, along with the appropriate modification for the radar modality's full field-of-view and is shown in particular for the Oxford Radar RobotCar Dataset in Figure 4.

Importantly, some care must be taken when dividing the training set into classes. This is because there is an inherent problem with training a place recognition model via classification: very similar images may be assigned to different classes if they geographically lie on different sides of a class boundary despite being right next to each other. This means the classifier must learn that virtually identical images actually belong to different classes. To overcome this challenge, the dataset is split into *groups of classes*, where each group contains only non-adjacent classes. Therefore, each group is not susceptible to the class boundary problem, and we can train on each group sequentially [3]. These groups of classes are called *CosPlace* groups.

As illustrated in Figure 1, this also results in us training multiple classifier heads (and uncertainty heads, see Section 4.3 below).

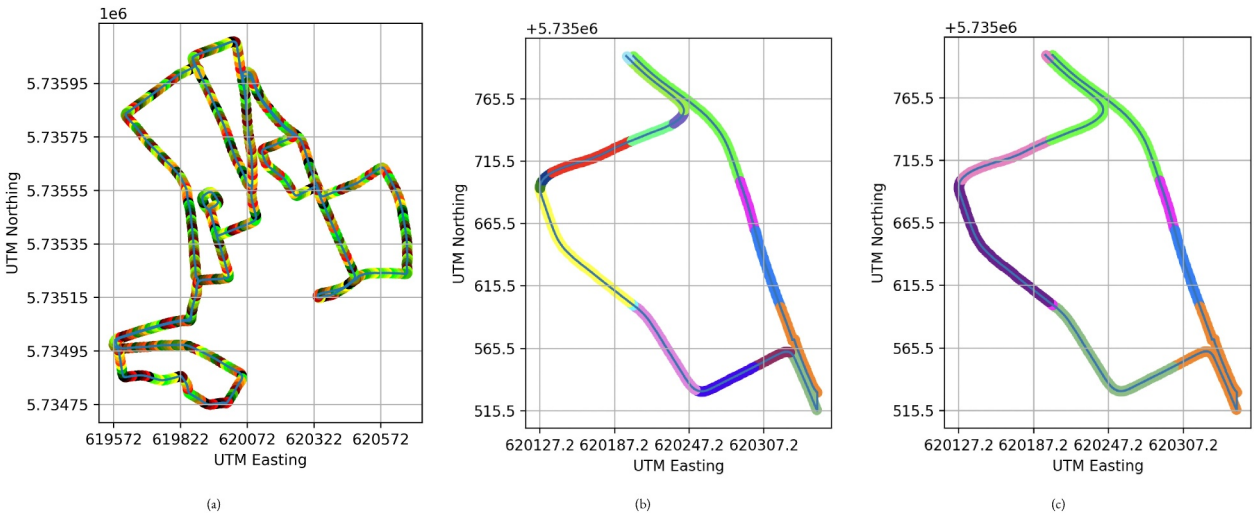


FIGURE 2 (a) The different *CosPlace* training groups are separated by colour on this example traversal, which corresponds to the satellite view in Figure 3. Each colour corresponds to a different location, or class in the classification problem. (b) Visualising the *CosPlace* training groups within a smaller geographical area, with GPS positions demarcated by both heading and UTM northing/easting—we are only looking at the area corresponding to *Keble College, Oxford*. (c) The *CosPlace* groups when we split the dataset up by GPS position, that is, UTM northing/easting only (not by heading). The *CosPlace* work [3] used cameras, so the limited field-of-view must also be considered when defining classes, as a single class can contain visually different scenes when looking in different directions. Our radar, in contrast, has 360° field-of-view (see Section 5.1 for more details). Therefore, defining location classes based on heading would lead to poor training of the neural network, particularly when randomly rotating input training radar scans from the same place (for rotational invariance). Therefore, (c) has fewer classes as compared to (b), for example, in the bottom right, the orange class holds for both forward and backward traversals of the same street, whereas in Figure 2b, the forward traversal is green, as the orientation was distinct. Please also see Figure 3.

FIGURE 3 On the left, a 90° field-of-view is assumed, so each cell must be further divided into 4. On the right, we have a 360° field-of-view, so each class corresponds to an entire cell. This is formalised in Section 4.1.

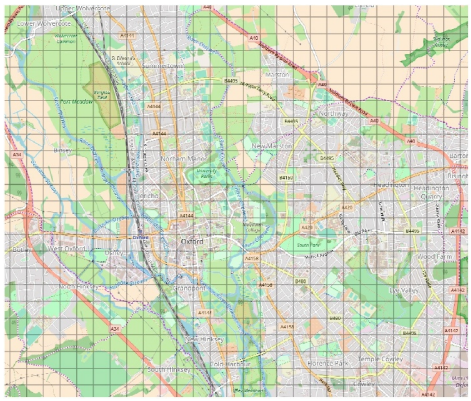


FIGURE 4 Class locations on the *Oxford Radar RobotCar Dataset* (shown as cells).

3.2 | Large margin cosine loss

The LMCL was recently proposed as a novel loss function for face recognition [5]. Central to face recognition problems such as face verification and identification is the task of discriminating facial features. Traditionally, deep convolutional neural networks have utilised the softmax loss, but it has recently been argued [40–42] that the standard softmax function lacks the discriminative power required to maximise inter-class variance and minimise intra-class variance. The softmax function is defined as follows:

$$\sigma(g_i) = \frac{e^{g_i}}{\sum_{j=1}^{C} e^{g_j}}, \tag{1}$$

where C is the number of classes and $\mathbf{g}$ would be a vector of output activations, or logits, from a neural network, each one corresponding to a *CosPlace* cell (class within a group).

The softmax function has some useful properties, one of which is that its element-wise application to a vector (as above) results in a probability distribution over the elements of the vector. The element which maximises this function gives the classification result.

It also means the softmax can be defined as the likelihood function for a classification task [1, 43], so we can define a loss function using the negative log-likelihood and average over N training samples. This leads to the softmax loss function:

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{e^{g_{y_i}}}{\sum_{j=1}^{C} e^{g_j}}, \tag{2}$$

where an input feature vector $\mathbf{x}_i$ has a corresponding label y_i. Again, $\mathbf{g}$ is a vector of activations of a fully connected layer at the end of a classification neural network.

The LMCL reformulates the softmax loss of Equation (2) [5]. Here, activation g_j can be written as produced by a weight vector $\mathbf{W}_j$ and bias B_j. If bias B_j is set to 0 for simplicity, g_j can be written as

$$g_j = \mathbf{W}_j^T\mathbf{x} = \|\mathbf{W}_j\|\ \|\mathbf{x}\|\cos(\theta_j), \tag{3}$$

where θ_j is the angle between $\mathbf{W}_j$ and $\mathbf{x}$. As the magnitude $\|\mathbf{W}\|$ should not affect the discriminative power of the features (just their scale), it can be L_2 normalised to 1. This improves the conditioning of the optimisation problem and speeds up the convergence of stochastic gradient descent [44]. Furthermore, fixing $\|\mathbf{x}\| = s$ removes radial variations so the model only learns features that are separable in the angular space. This is equivalent to requiring $\cos\theta_1 > \cos\theta_2$ to classify into class C_1 (for the case of binary classification, but the same idea extends to more classes).

Finally, LMCL introduces a cosine margin to discriminate features according to the tighter constraint $\cos\theta_1 - m > \cos\theta_2$. This encourages an extra margin in the cosine space to reinforce discrimination of features.

Formally, the LMCL is defined (subject to the normalisation of feature and weight vectors) as [5]

$$L_{\text{lmc}} = -\frac{1}{N}\sum_{i}\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{y_i,j})}}, \tag{4}$$

where $\cos(\theta_{j,i}) = \mathbf{W}_j^T\mathbf{x}_i$ is the angle for the j^{th} activation element producing by the i^{th} sample input to the network, and therefore $\theta_{y_i,i}$ indicates the angle for the activation of the ground truth class y_i for the i^{th} sample. This is therefore referred to as a cosine loss. Large margin cosine loss is used as the loss function for training *CosPlace* groups [3].

3.3 | Entropy-based uncertainty

The softmax function, given in Equation (1), appears just before the output layer. Its inputs are the activations of the final hidden layer, also referred to as logits, and its output is a categorical probability distribution over the classes. The softmax layer must therefore have the same number of nodes as the output layer. If X represents a random vector with a categorical distribution over k classes, we can express the Shannon entropy of X as

$$\mathrm{H}(X) = -\sum_{i=1}^{k} p_i \cdot \log p_i, \tag{5}$$

where p_i is the probability of class i.

In the context of place recognition, each class corresponds to a cell within a *CosPlace* group (see Section 3.1 above). Entropy is an established measure of uncertainty from information theory; it is maximised if we have a uniform distribution over the classes, as we effectively have no information about the relative likelihood of classes. By contrast, entropy is low for a peaked distribution as this is more informative; we can interpret this as the model being highly confident in its predictions. Modern neural networks are widely acknowledged to be overconfident in their predictions, with poorly calibrated uncertainty [45, 46].

The contribution of our work is a learnt predictive uncertainty which outperforms this. Our method is fundamentally different as we use a data-driven uncertainty metric that is learnt. We use the naïve entropy-based uncertainty as a baseline for comparison in our experimental results in Section 6, where we show that our proposed uncertainty metric outperforms entropy.

4 | METHOD

Our method is designed according to these two key findings from our review in Section 2:

1. Existing approaches to radar-based place recognition focus on learning features optimised for matching, which are then compared to identify places, and
2. Uncertainty estimation in such methods typically focuses on quantifying the variability in these features, rather than directly estimating the uncertainty of place predictions

In contrast, our method *reframes radar place recognition as a classification task*. By treating the problem as place classification rather than feature matching, we enable the direct estimation of uncertainty as part of the classification output. Specifically, the confidence distribution over predicted places inherently provides a measure of uncertainty. This approach simplifies the pipeline, bypasses the need for explicit variability modelling of features, and aligns uncertainty estimation directly with the place recognition task.

4.1 | *CosPlace* groups for the radar modality

Unlike the cameras used in ref. [3], our radar has a complete 360° field of view. Therefore, we define the groups without any consideration of angular separation. We now justify this formally. A naïve approach to defining location classes would be [3]

$$C_{e_i,n_j,h_k} = \left\{ (\text{east}, \text{north}, \text{heading}) : \left\lfloor \frac{\text{east}}{M} \right\rfloor = e_i, \left\lfloor \frac{\text{north}}{M} \right\rfloor = n_j, \left\lfloor \frac{\text{heading}}{\alpha} \right\rfloor = h_k \right\} \tag{6}$$

Here, the UTM coordinates of a place are {east and north}, and M and α determine the extent of each class in position (in metres) and heading (in degrees), respectively. However, this does not satisfy the requirement for non-adjacent cells within groups discussed above. For this, two more parameters are required.

Firstly, N defines the minimum number of cells between two classes of the same group. Secondly, L defines how many groups are present within a single cell (separated by heading). A *CosPlace* group G_{uvw} can now be defined by the set:

$$G_{uvw} = \left\{ C_{e_i,n_j,h_k} : (e_i \% N = u) \wedge (n_j \% N = v) \wedge (h_k \% L = w) \right\}. \tag{7}$$

Here, ignoring heading for illustration purposes, for example $N = 4$, we have $4\%4 = 0$ as well as $8\%4 = 0$ such that classes with $e_i = 4, 8, \ldots$ and $n_j = 4, 8, \ldots$ belong to the same group identified by $u, v = 0$ while we also have $3\%4 = 1$ and $7\%4 = 1$ such that classes with $e_i = 3, 7, \ldots$ and $n_j = 3, 7, \ldots$ belong to the another group identified by $u, v = 1$.

In our work, we altered the parameters by setting $\alpha = 360°$ (as this is the case for radar) and $L = 1$. The effect of the first is to totally coarsen the discretisation of heading in Equation (6) entirely with

$$\left\lfloor \frac{\text{heading}}{360°} \right\rfloor = 1 \quad \forall \quad 0° < \text{heading} < 360° \tag{8}$$

The effect of the second is $h_k \% 1 = 0 \quad \forall k$. Thus, we have $w = 0$ for all groups, which is to say that heading has no effect on the group.

In Figure 2c, we show *CosPlace* groups separated by UTM position only, not by heading, in this way. We zoom in on a smaller area of the dataset for clarity. This can be contrasted with Figure 2b, where we split groups by both UTM position and heading.

4.2 | Multi-headed predictive uncertainty over class locations

We wish to be able to pick out specific queries that have higher uncertainty than others and reject these queries for classification if there is insufficient confidence to make a decision. When using a certain rejection rate (say 25%), the network will effectively take its set of queries and decline to localise the 25% most uncertain ones, as determined by the learnt uncertainty score. So there is no set decision boundary for uncertainty as this is a percentile-based system.

In brief, and as illustrated in Figure 1, we split the network such that it has two output heads: one maps an input to its logit output $f_i^{\hat{w}}$, while the other maps to its learnt uncertainty, given by a variance, $\left(\sigma_i^{\hat{w}}\right)^2$.

Our loss function draws on the Bayesian neural network presented by Kendall and Gal [1], with adaptations to work with the LMCL loss (c.f. Section 3.2).

As per [1], a Gaussian distribution centred on $f_i^{\hat{w}}$ and spread by $\sigma_i^{\hat{w}}$ is placed over the vector of logits:

$$\hat{x}_i | \hat{w} \sim \mathcal{N}\left(f_i^{\hat{w}}, \left(\sigma_i^{\hat{w}}\right)^2\right) \tag{9}$$

with $\hat{w}$ as the parameters of the network. Note that the variance $\left(\sigma_i^{\hat{w}}\right)^2$ does not *directly* represent distance error, as this demands ground truth (e.g. GPS), and we are interested in deploying radar-only localisation systems. Instead, we trained such that it is correlated with distance error, serving as a predictive measure. This enables the system to evaluate and reject unreliable localisation results, improving accuracy by acting as a proxy for assessing confidence in estimated outputs.

Sampling from this Gaussian can be efficiently achieved by sampling from a zero-mean, unit-variance Gaussian $\mathcal{N}(0, I)$ as

$$\hat{x}_{i,t} = f_i^{\hat{w}} + \sigma_i^{\hat{w}} \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{10}$$

where t is the sample index. Through this, we vary the logits $\hat{x}_{i,t}$ and simulate sampling weights $\hat{w}$ (given that the mean and variance are produced by the neural network and in turn impacted by its parameters).

Now, considering the logits available as per the cosine form in Equation (3) and the subsequent probability output by the softmax in Equation (4) as

$$\hat{p}_{c,i} = \frac{e^{s\left(\cos\left(\theta_{c,i}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{c,i}\right)-m\right)} + \sum_{j\neq c} e^{s\,\cos\left(\theta_{c,j}\right)}} \tag{11}$$

we can express with c' representing the ground truth class for input i an expected value of the probability of the ground truth class as

$$\log \mathbb{E}_{p(\hat{x}_i\,|\,\hat{w})}\left[\hat{p}_{c',i}\right] = \log \int \hat{p}_{c',i}\; p\left(\hat{x}_i|\hat{w}\right)\; d\hat{x}_i. \tag{12}$$

If we maximise Equation (12), we therefore maximise the expected value of the probability assigned to the correct class.

In practice, Equation (10) facilitates taking the expectation over $p\left(\hat{x}_i\,|\,\hat{w}\right)$, where we approximate Equation (12) by Monte Carlo integration as

$$\begin{aligned}
&\log \mathbb{E}_{p(\hat{x}_i|\hat{w})}\left[\hat{p}_{c',i}\right] \\
&\approx \log \frac{1}{T}\sum_t \frac{e^{s\left(\cos\left(\theta_{i,c',t}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{i,c',t}\right)-m\right)} + \sum_{j\neq c'} e^{s\,\cos\left(\theta_{i,j,t}\right)}} \\
&\hat{x}_{i,t} = f_i^{\hat{w}} + \sigma_i^{\hat{w}}\,\epsilon_t,\ \epsilon_t \sim \mathcal{N}(0,I)
\end{aligned} \tag{13}$$

where the sampled $\hat{x}_{i,t}$ produces $\cos\theta_j$ as per Equation (3), that is, with $\cos\theta_{j,i,t} = \mathbf{W}_j \cdot \mathbf{x}_{i,t}$.

With the over all samples, the final loss function that we used to train our uncertainty-aware neural network is

$$L_{\text{final}} = -\sum_i \log \frac{1}{T}\sum_t \frac{e^{s\left(\cos\left(\theta_{i,c'}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{i,c'}\right)-m\right)} + \sum_{j\neq c'} e^{s\,\cos\left(\theta_{i,j}\right)}} \tag{14}$$

Here, i indexes the sample in the dataset, t indexes the Monte Carlo sample, and c indexes the class.

4.3 | Coarse-to-fine uncertainty-aware network architecture

To train our network and measure uncertainty, we need to output both a logit $f_i^{\hat{w}}$ and variance $\sigma_i^{\hat{w}}$. This is done by two separate heads after the backbone feature extractor, as illustrated in Figure 1.

The head which produces $f_i^{\hat{w}}$ is a `softargmax` similar to the softmax in Equation (4), that is, it is the argument which maximises the probabilities in the LMCL.

In order to not constrain the uncertainties as probabilities (which they are not), the uncertainty heads are implemented differently—simply as a fully connected layer with input size and the dimensionality of the output of the backbone feature extractor, and output size equal to the number of classes in the *CosPlace* group.

As a reminder, each group has its own optimised classifier. Thus, under our system, each group has both its own optimised classifier and its own optimised uncertainty head which is trained to express uncertainty in the set of location classes in that group.

As per the original *CosPlace* work [3], we discard the classifier layers and perform place recognition in the usual way—by matching output features of the backbone feature extractor. This means that the trained features can be used in other deployments/domains, which do not share the same classes/locations.

Under our proposed architecture, the uncertainty heads are not discarded. This means that uncertainty can only be expressed in the domain that you have trained the neural network in (i.e. over the set of discretised GPS classes that you have trained the network with). This is not so different to [31], where uncertainty capability is fine-tuned in the target domain.

In addition to this, we also need to select the maximum score from just one of the various uncertainty heads (corresponding to groups of location classes) to use as the final uncertainty score. As each of the uncertainty heads is dedicated to a subset of classes which corresponds to location, this means the usability of our uncertainty mechanism is loosely coupled to the location itself.

However, considering that the *Oxford Radar RobotCar Dataset* that we experiment with (c.f. Section 5.1) is over an area of approximately 1,500,000 m^2 and we use approximately 900 *CosPlace* classes, we have that each class corresponds to approximately 1600 m^2 of location classes. Considering that our place recognition system is accurate to approximately 600 m^2 (see Section 5.4 below), this means that our uncertainty mechanism requires only a coarse a priori knowledge of position.

Considering the benefits of a calibrated localisation uncertainty to introspective localisation (which we measure in Section 6.3) and considering other similar radar localisation systems such as [14, 21, 35] as well as, more broadly than radar, coarse-to-fine systems in LiDAR such as *ScanContext* [15, 18] which operate with a hierarchy or coarse-to-fine sub-localisers, this is a good trade-off.

5 | EXPERIMENTS

5.1 | Dataset

We evaluate our method across urban data collected in the *Oxford Radar RobotCar Dataset* [4]. The dataset was collected using an autonomous-capable *Nissan LEAF*. This follows the original *Oxford RobotCar Dataset* [47] route over 32 traversals in different traffic, weather and lighting conditions, totalling ≈280 km of driving. The dataset uses a *CTS350-X Navtech* FMCW scanning radar with an operating frequency range of 76–77 GHz and a scan rate of 4 Hz. The *CTS350-X Navtech* rotates about its vertical axis while transmitting and receiving frequency-modulated radio waves. Scans have 3768

range bins at a resolution of 4.38 cm, with the total range of 165 m. There are 400 azimuths (i.e. angular resolution 0.9°). Some example scans are shown in Figure 5, with green being good matches and red being bad matches—with far more good matches for our approach (bottom three rows) as compared to the non-BNN approach (top three rows).

5.2 | Cross-validation

From these 32 traversals, or experiences, of Oxford in the *Oxford Radar RobotCar Dataset*, we used five (each consisting of approximately 8000 radar scans) for our cross-validation experiments.

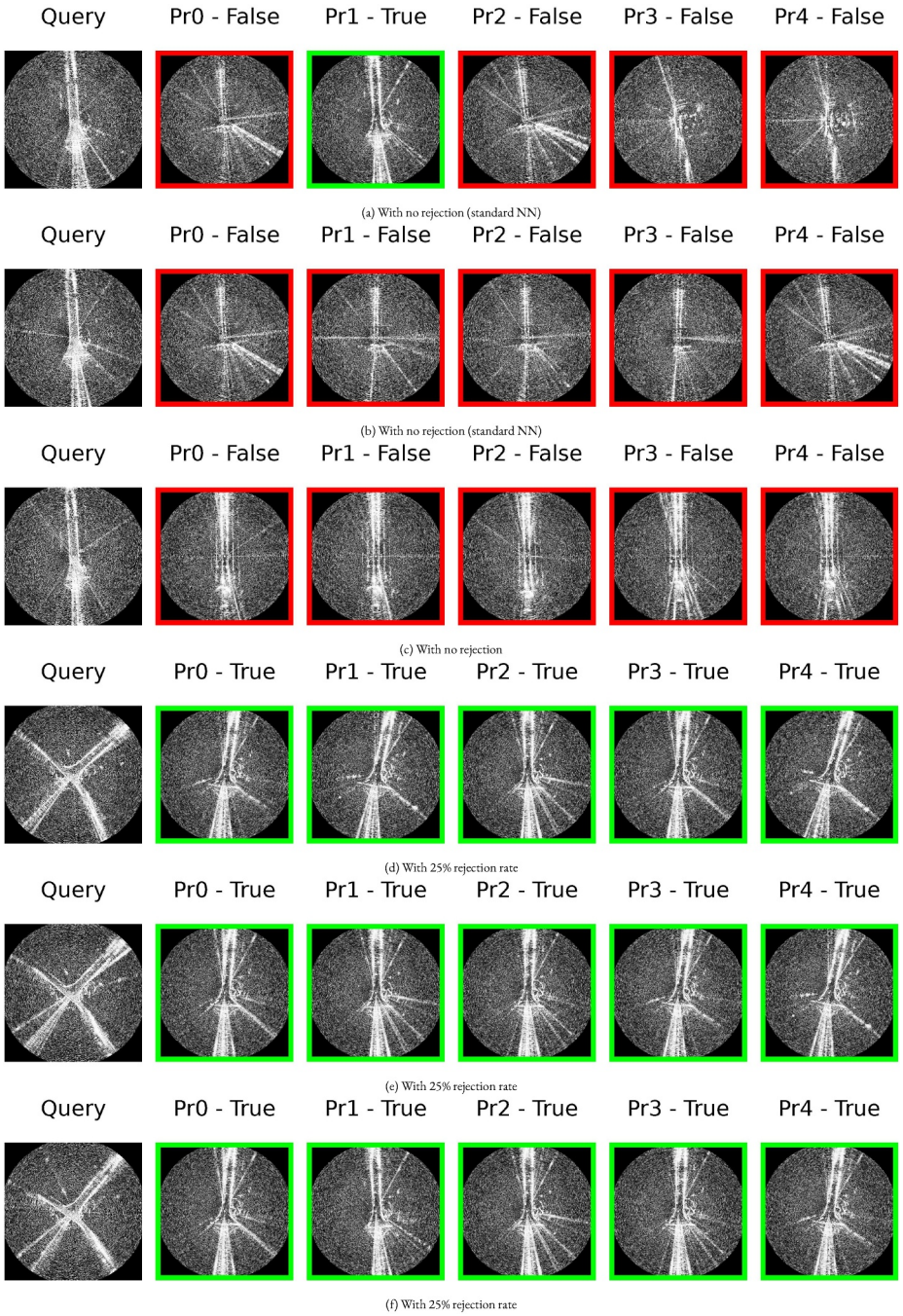


FIGURE 5 Comparison of performance with and without uncertainty-based rejection. Here, `Pr` refers to 'prediction' and corresponds to the n^{th} nearest neighbour to the `query` (0–4). Whether this returned candidate is indeed a correct localisation match is indicated by `true/false` as well as by green/red (respectively). Each of the first three rows (a)–(c) shows a set of five predictions from the network, with the corresponding query image shown on the far left of each row. If a certain prediction returned by the network is within 25 m of the query scan location, it is counted as a positive result. Conversely, radar scans located more than 25 m from the query image are counted as incorrect predictions. The bottom three rows (d)–(f) show a *different* set of three rows, as this time, the network rejects the 25% of queries it is least certain about, which can include the three rows in a to c. Qualitatively, the 25% rejection rate greatly improves the accuracy of the network on the queries that it does still localise.

Testing requires two experiences—one to act as the set of localisation queries and the other to act as the database from which the most likely matches are returned. The correspondence between database and query sets is shown in the 'difference matrices' of Figure 8, where we plot the pairwise feature-space distances between query and database descriptors.

Our cross-validation setup takes from the five experiences for each cross-validation trial: one for training, two for validation and two for testing. We then conducted 5 cross-validation runs; each one using one of the five experiences for training,[1] and the others for validation and testing. Specifically, for each run, one of the 5 datasets is used for training, while the other 4 are used for validation and testing and we then *cyclically permuted* the experiences used for training, validation and testing before presenting all results as averaged over these five permutations (cross-validation results) In this, each of the five trained models was trained on independent data, with each one being tested and validated on data that was independent from its training data.

5.3 | Performance metrics and uncertainty calibration

To assess place recognition performance, we use `Recall@N (R@N)`. This metric measures the percentage of query frames which have a nearest neighbour in the map which is actually close in a physical space. Specifically, when N candidate map descriptors/locations are retrieved as nearest neighbours, `Recall@N` considers the query successful if at least one of them is from a nearby place in the physical space. This is the standard performance metric in place recognition research [48]. 'Nearby' is taken as at most 25 m, as per [3, 11, 22, 25, 48]. This evaluation is *distance-based* because localisation accuracy is determined by comparing the physical distances between scan capture points, measured using GPS synchronisation, to validate correct or incorrect matches within a predefined threshold. Note that we cannot use this distance-based evaluation online because it relies on ground truth locations, provided by GPS synchronisation, to measure accuracy. In real-time applications, ground truth data is unavailable for such validations.

For this, we use a 'difference matrix' in Figure 8 *Right* with the embedding distances between live and map features and compare it to a similar ground truth matrix which uses the GPS/INS provided in the *Oxford Radar RobotCar Dataset* (a NovAtel SPAN-CPT ALIGN inertial and GPS navigation system).

We test uncertainty calibration by rejecting between 0% and 100% of queries based on the level of uncertainty we predict them to have (see Section 4.3). This uncertainty level is as per our method as well as some other competitors.

If the predicted uncertainty is poorly calibrated with the accuracy of the model, it will reject queries that would have ultimately led to good localisation results, and the `Recall@N` performance will decrease as more queries are rejected. If the predicted uncertainty is well-calibrated to localisation outcomes, the system will reject queries that would have been poorly localised, and we should see improved `Recall@N` as more queries are rejected.

Beyond `Recall@N`, we also present precision-recall curves. In this, we do not retrieve N neighbours but instead retrieve all neighbours within some embedding-distance threshold τ. The locus of the precision and recall over a range of τ values defines the precision-recall curve.

While our results are presented over a range of thresholds (forming e.g. precision-recall curves), the reader should choose this threshold for new deployments based on the following guide:

1. Collect a calibration dataset that represents the conditions and environments similar to the target deployment data.
2. Pass the calibration data through the feature matching and uncertainty estimation pipeline, computing the uncertainty scores for each localisation attempt.
3. Annotate the calibration data with ground truth labels (e.g. GPS) to identify which localisation attempts are correct (true matches) or incorrect (false matches).
4. Compute precision and recall across a range of uncertainty thresholds using the calibration data.
5. Find the highest threshold value where the system achieves 100% precision, ensuring that all accepted matches are correct.
6. Integrate this threshold into the deployed system to filter localisation attempt based on uncertainty.

5.4 | Baselines and ablation

We compare against other uncertainty-aware radar place recognition systems. We also perform an ablation on our system and compare it with and without uncertainty-aware features—as the core contribution of this work. The methods for comparison are

1. `LMCL`[B1] or `Non-BNN`[B1b]: This is *CosPlace* [3] but trained on radar data and with uncertainty based on logit entropy as in Section 3.3.
2. `BNN`[B2]: This is our method as described in Section 4, including trained uncertainty heads. In our results, we make it clear whether we are referring to the base performance of the features learnt in this approach or whether performance is based on some level of introspective query rejection. Regardless, all results labelled this way are at least *trained* in an uncertainty-aware fashion.
3. `VML`[B3]: As in ref. [6], this is a recent method for uncertainty-aware radar place recognition that relies on variational metric learning. This is thus an alternative probabilistic setup, but our method also differs in that we do not fine-tune the feature space by metric learning—ours being rather classification-based, as per *CosPlace*.

[1] `2019-01-10-11-46-21, 2019-01-10-12-32-52, 2019-01-10-14-02-34, 2019-01-10-14-50-05, 2019-01-10-15-19-41`

This was the core motivation for our work, where our intuition is that the uncertainties learnt by our proposed method are more interpretable than methods like VML [6] which yield a confidence which we feel is more directly interpretable as a variance in the data, rather than the task output.

4. STUN[B4]: As in ref. [2]. this first trains a teacher net using a standard metric learning pipeline to produce embedding priors. Then, supervised by the pretrained teacher net, a student network with an additional variance branch is trained to fine-tune the embedding priors and estimate the uncertainty on a sample-by-sample basis. STUN has already been shown to be outperformed by VML in ref. [6]—in the context of introspective radar place recognition—but we include it here for completeness—there not being many place recognition systems which are uncertainty-aware (in any sensor modality).

For VML and STUN we re-tested our model on the same testing data as is used in ref. [6]—5 different traversals (note: not the same as the 5 traversals discussed in Section 5.2 above[2] of the *Oxford Radar RobotCar Dataset*. In this experimental setting, each testing run requires a pair of experiences (for queries and database sets), so we have a total of 10 cross-validation runs, since $\binom{5}{2} = 10$. Also in this experimental setting only each dataset is also only a partial traversal of the full route of the *Oxford Radar RobotCar Dataset*—meaning we only consider a small subsection of the data. It is important to note that all other results in this work are presented over the full route, making this work more comprehensively tested than in competing systems [6].

5.5 | Architecture, settings and hyperparameters

The backbone of the neural network architecture was a ResNet18 model. This is a convolutional neural network with 18 layers. It consists of skip connections, which have been shown to alleviate the vanishing gradient problem in deep neural networks by forwarding activations to later network layers [49]. Throughout the experiments, we choose a fully connected output layer dimension of 512. Our model was trained with a batch size of 8 for 50 epochs, each with 10,000 iterations. We used the Adam optimiser [50] with a learning rate of $1e-5$.

The hyperparameters for the CosPlace method were set as follows. We had 25 *CosPlace* groups (defined in Section 4.1), separated only by UTM coordinates, and a total of approximately 900 classes, that is, cells of GPS coordinates around an interest location. There were at least 5 cells/classes of separation of the same group.

[2] 2019-01-18-14-46-59, 2019-01-18-15-20-12, 2019-01-18-14-14-42, 2019-01-14-14-15-12, 2019-01-16-14-15-33

6 | RESULTS

6.1 | Benefit of uncertainty-aware training to features for place recognition

Figure 6 presents a precision-recall curve which is averaged over all 5 cross-validation trials (as defined in Section 5.2).

It is important to note that precision-recall as we calculate it is a much more strict definition of localisation success than Recall@N curves (see below). Specifically, it is defined as

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TP}}, \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{15}$$

Consider in this the number of false negatives (FN), being defined as the number of database candidates from nearby physical locations which are not retrieved by the embedding-space search. Contrast this with the relatively looser definition of success for Recall@N, where—even when N database candidates are retrieved—only one being a true localisation defines the query as being *totally* successful.

In any case, we see from Figure 6 that our proposed system, BNN, outperforms non-BNN. Specifically, consider the recall at 100% precision. At this point on the curve, *none* of the retrieved candidates are incorrect. BNN enjoys a recall at 100% precision of around 10%–15%, whereas non-BNN (i.e. LMCL, the vanilla *CosPlace* approach with no uncertainty-aware training) has 0% recall at 100% precision. This means that non-BNN is not able to be tuned to the point where it gets any localisation results with all of them being correct.

In the middle region in Figure 6, the performance of the non-BNN model slightly surpasses that of our proposed BNN method. Here, our uncertainty-aware model might be overly conservative—filtering out some true positives or down-weighting certain matches due to its penalty for uncertainty.

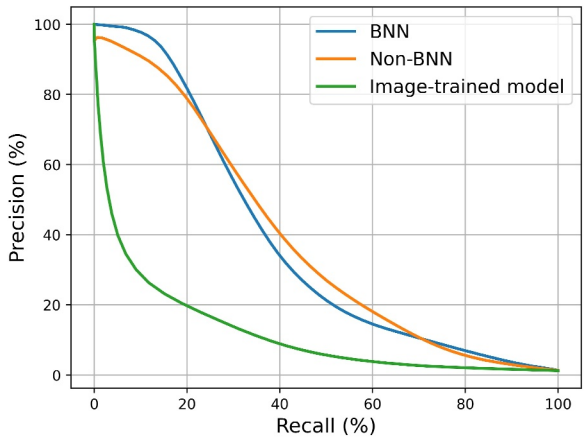


FIGURE 6 Precision-recall curve comparing our system with the *CosPlace* trained on radar scans (non-BNN) and the pre-trained standard *CosPlace* model trained on images. The pre-trained standard *CosPlace* model trained on images is included as a trivial baseline, solely to confirm the task is not easy. Our approach exhibits the best performance, especially when high precision is needed.

This cautious behaviour results in slightly lower precision compared to the standard method in these regions. The important region of operation is the low-recall scenario in the upper left of these plots. Here, the uncertainty-aware model is more cautious and informed decisions, prioritising precise matches over risky or ambiguous ones.

This experiment, not yet exploring the introspective ability of our learnt predictive uncertainty (see below), validates that our novel approach of learning a predictive uncertainty for place recognition is in fact beneficial for the features that the encoder produces—that is, it is beneficial for training *CosPlace*. This is likely due to the network being able during training to avoid overfitting to difficult, uncertain samples.

Qualitatively, consider Figure 8 where we visualise the 50 nearest neighbours in the map (columns) to each query descriptor (rows). We note that the matrix based on our proposed system BNN (*Bottom*) has a darker streak towards the top right than for LMCL (*Top*), which was a situation where the robot was revisiting a short section of the route. Being darker means that our approach picks up more true positives in this scenario.

6.2 | Benchmarking against other uncertainty-aware radar place recognition systems

Table 1 compares our approach against two other systems which have been specifically trained and tested for uncertainty-aware place recognition, VML and STUN (described in Section 5.4 above). This is similar to Section 6.1 done by simply comparing the quality of the learnt features themselves to showcase that uncertainty-aware training is beneficial to enforcing that metric space, with Section 6.3 below exploring the calibration of the learnt predictive uncertainty and its use for introspective query rejection. Note that the test set for these results is totally unseen during training or calibration of the tested models—see Figure 7.

As can be seen, our approach, BNN, outperforms the others for Recall@1, Recall@5, and Recall@10. For example, we have 91.1% for Recall@1, whereas VML and STUN have only 80.0% and 74.6%, respectively.

It should be noted that, in order to be consistent with the results presented in ref. [6], testing in this case is done with a *soft zone*, where the positive distance threshold is 25 m (as for *all* other results in our work), but in Table 1, we do not count a result as a false positive unless it is incorrect by more than 50 m. This is done in ref. [6] which itself was based on the experimental setup in ref. [19].

Consider Figure 8 again. Despite the darker streak at the top right in the short revisiting section as discussed in Section 6.1 for our method, both seem to pick up some amount of false positives (see the general 'speckle noise' across the matrices). This is where query rejection via uncertainty can be useful—those false positives should be identified early (at the point of query) by a predictive uncertainty as being unlikely to result in good localisation.

TABLE 1 Comparison of our system with other uncertainty-aware place recognition methods, VML [6] and STUN [2]. Our system performs best across Recall@1, 5 and 10.

	BNN	VML	STUN
Recall@1	91.1	80.0	74.6
Recall@5	96.9	92.3	89.5
Recall@10	98.6	94.1	97.5

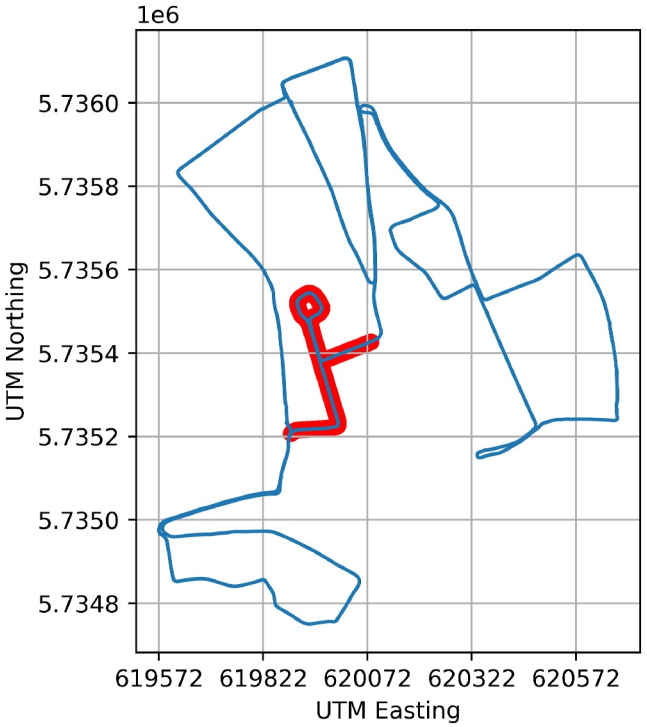


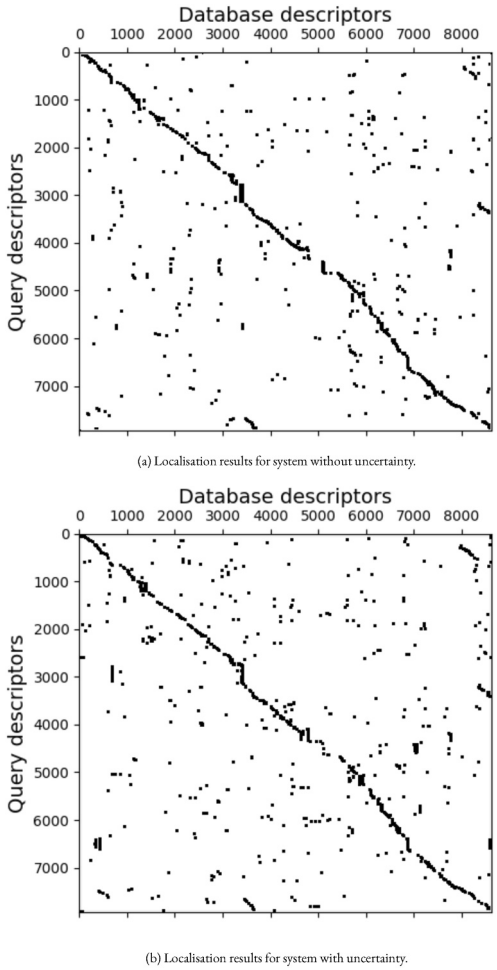
FIGURE 7 Held-out test portion (red) from the *Oxford RobotCar Dataset* which is not seen during training or calibration of uncertainty thresholds.

6.3 | Introspective query rejection with a calibrated location-specific uncertainty

Therefore, we next test the quality of our proposed uncertainty mechanism versus that which is native to a classification-based place recognition network like *CosPlace* (i.e. softmax scores, see Section 3.3 above).

Specifically, for the LMCL baseline and our proposed approach BNN, we apply a range of thresholds to the output uncertainty scores. Where the threshold is exceeded, a query localisation is 'denied'—which is to say that localisation is not attempted in that scenario. Thus, we sacrifice the localisation rate across the route. The trade-off, however, is that the localisation queries which are not denied are more generally successful (result in a localisation which is in fact correct).

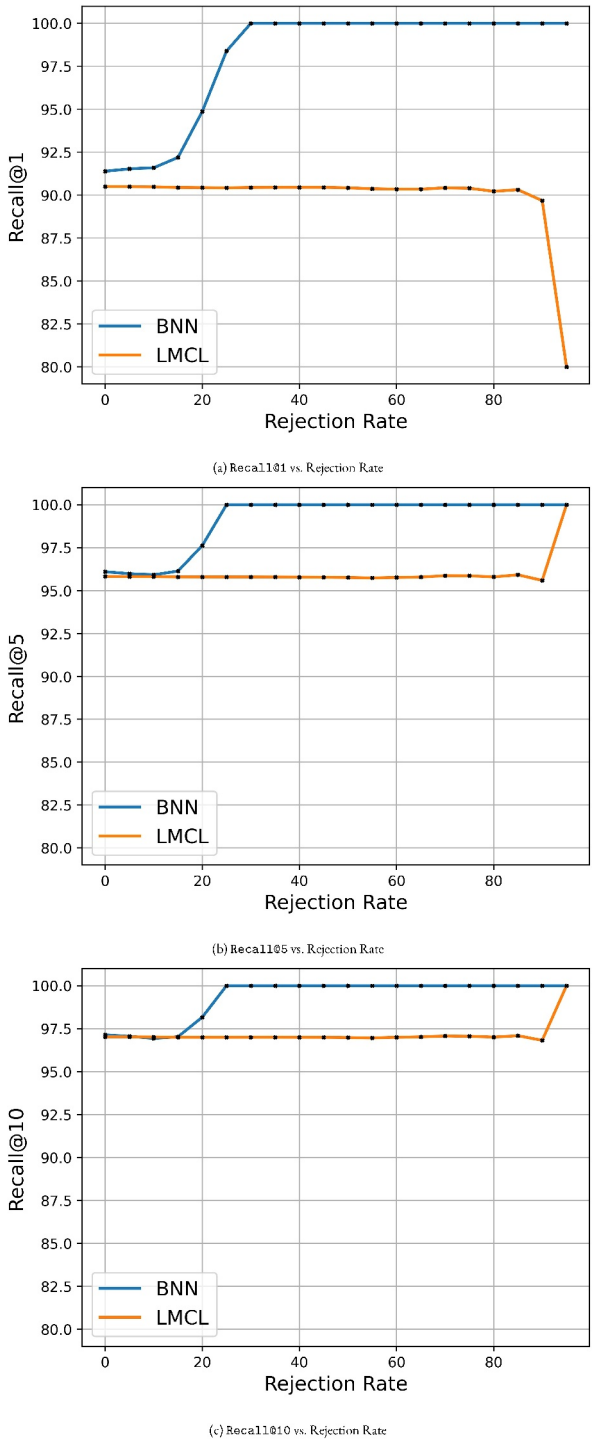
Indeed, consider Figure 9. Here, *rejection rate* corresponds to the percentage of queries that are rejected. Note that 100% of the rejection rate *would* correspond to rejecting all queries, which would lead to the vehicle never being localised. This is



(a) Localisation results for system without uncertainty.

(b) Localisation results for system with uncertainty.

FIGURE 8 Comparison of the top 50 results (as determined by the distances in the distance matrix) between the system without (top) uncertainty heads and the system with uncertainty heads (bottom). This is effectively a visualisation of `Recall@50` localisation results. Dark black marks represent localisation matches. The same general level of false matches is observed, but the dark streaks in the top right are more pronounced, indicating that the features learnt by our proposed approach are more discriminative. Furthermore, our system is better capable of rejecting these false matches—see Figure 9 and Section 6.3.



(a) `Recall@1` vs. Rejection Rate

(b) `Recall@5` vs. Rejection Rate

(c) `Recall@10` vs. Rejection Rate

FIGURE 9 Recall improves for both models as the *rejection rate* increases, but it improves faster for our approach, suggesting our learnt uncertainty is better calibrated with model performance.

therefore not done in this paper. We claim 100% precision/recall at certain rejection rates, but these rejection rates are always less than 100%. Here, our approach (blue) tends to 100% for `Recall@1/5/10` as the rejection rate increases. The baseline (orange) generally remains flat, indicating that the uncertainty is not generally correlated with the correctness of the localisation outcome. For `Recall@5/10`, the baseline does eventually approach 100%—however, only for very high rejection rates (where almost all queries are denied), whereas ours is at 100% from the ≈30 % rejection rate. For `Recall@1`, the baseline actually drops in the success rate at these high rejection rates, whereas ours is again at 100% from the ≈30 % rejection rate.

Note that all the data in Figure 9 is averaged over cross-validation trials using the 5 traversals of the *Oxford Radar RobotCar dataset* [4] as defined in Section 5.2.

Figure 10 shows the corresponding rejection curve, that is, when testing in an uncertainty-aware fashion. This test is carried out over the route, as shown in Figure 7, and therefore constitutes a new deployment as compared to training or uncertainty calibration data. For instance, consider the 35% rejection rate point in Figure 9, where our approach reaches

100% recall. A similar point in Figure 10 has our approach outperforming VML but itself outperformed by STUN. However, we outperform at lower rejection rates and also reach 100% Recall@10. This result serves to prove that calibrating this uncertainty threshold in a new domain is important and that we cannot lean too heavily on a threshold calibrated before the domain shift.

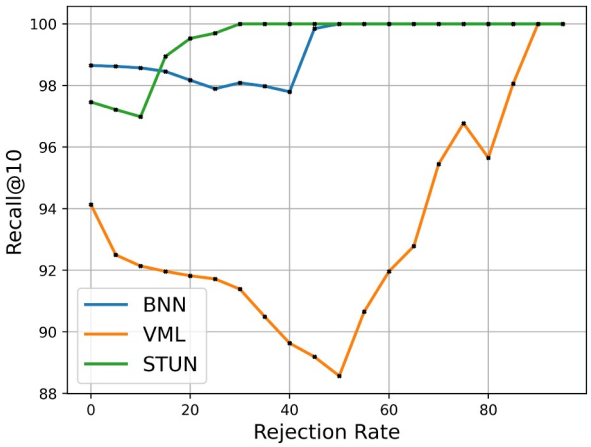


FIGURE 10 Results from a scenario different to the training/calibration datasets showing Recall@10 for our method in comparison to competing uncertainty-aware place recognition methods over a range of rejection rates.

Figure 11 evaluates the generalisation of our approach, and we conducted experiments using data collected from rugged off-road driving in the Scottish Highlands, taken from the *Oxford Offroad Radar Dataset* dataset [26]. The GPS trace shows the vehicle completing a route and then returning along the same path. The difference matrix highlights a clear main diagonal, demonstrating our method's ability to localise between two traversals of the same route, even in challenging environments.

7 | CONCLUSION

We have formulated place recognition as tightly coupled to classification of the location class for the first time. We showed that this outperforms radar place recognition systems based on the variability of data [6] and loss attenuation [51]. We also showed that this outperforms uncertainty based on the entropy of the classification layer outputs as a naïve baseline.

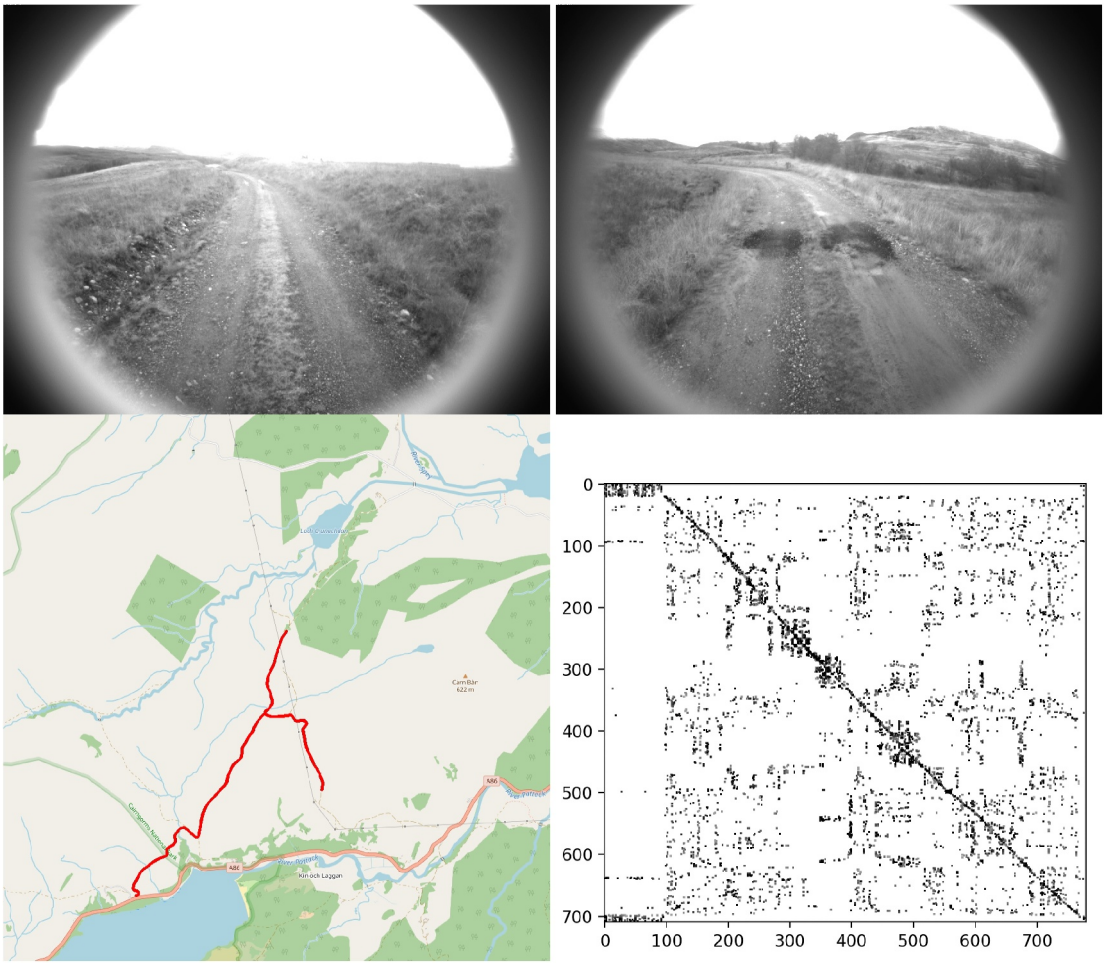


FIGURE 11 Demonstration of our approach's generalisation to rugged off-road driving data collected in the Scottish Highlands. The GPS trace illustrates the vehicle's route, which involved driving through open areas and wooden areas within a short distance of a river and ending at a reservoir. The main diagonal in the difference matrix is visible/evident, confirming successful localisation between two trips on the same route, showcasing the robustness of our method.

Uncertainty in place recognition is largely under-explored, but nevertheless important—where calibrated uncertainty can more broadly be used to fuse this source of location information in multi-localiser and multi-modal navigation and localisation systems—and this work will therefore be useful in the formalisation of place recognition uncertainty theory going forwards.

In the future, we will explore stacking the input with multiple frames from the recent history of scans recorded by the robot. We expect that uncertainty will be reduced, as it was for very confusing laser scan matching in the 'many-to-many' approach taken in ref. [52]. However, our previous work which develops sequential matching for radar place recognition [22] is not totally free from false positives; therefore, learning a predictive uncertainty over sequences should prove useful too. Although we focused on radar place recognition in this work due to the interesting complex sensor artefacts and noises for this sensor class, in the future, it will be interesting to see this new formulation applied to visual place recognition with *CosPlace*, as uncertainty due to, for example, visual ambiguity is a critical problem to tackle in that modality.

AUTHOR CONTRIBUTIONS

Suyash Agarwal: Conceptualisation; investigation; methodology; software; writing. **Jianhao Yuan**: Validation; writing. **Paul Newman**: Writing; funding acquisition; supervision. **Daniele De Martini**: Project administration; data acquisition; writing. **Matthew Gadd**: Conceptualisation; data acquisition; formal analysis; methodology; writing; supervision.

ACKNOWLEDGEMENT

EPSRC Programme Grant 'From Sensing to Collaboration' (EP/V000748/1). We are also grateful to our partners at Navtech Radar.

CONFLICT OF INTEREST STATEMENT

Daniele De Martini is a guest editor for RSN-2024-07-0188.R1. This article will not be handled by the guest editorial team in the review process and will instead be handled by one of the journal's associate editors.

DATA AVAILABILITY STATEMENT

Data used in this study are freely available at `oxford-robotics-institute.github.io/radar-robotcar-dataset`.

ORCID

Matthew Gadd https://orcid.org/0000-0001-9447-8619

REFERENCES

1. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. 30 (2017)
2. Cai, K., Lu, C.X., Huang, X.: STUN: self-teaching uncertainty estimation for place recognition. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 6614–6621. IEEE (2022)
3. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4878–4888. IEEE (2022)
4. Barnes, D., et al.: The oxford radar robotcar dataset: a radar extension to the oxford robotcar dataset. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2020)
5. Wang, H., et al.: CosFace: large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
6. Yuan, J., Newman, P., Gadd, M.: Off the radar: variational uncertainty-aware unsupervised radar place recognition for introspective querying and map maintenance. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2023)
7. Masone, C., Caputo, B.: A survey on deep visual place recognition. IEEE Access 9, 19516–19547 (2021). https://doi.org/10.1109/access.2021.3054937
8. Zhang, X., Wang, L., Su, Y.: Visual place recognition: a survey from deep learning perspective. Pattern Recogn. 113, 107760 (2021). https://doi.org/10.1016/j.patcog.2020.107760
9. Barros, T., et al.: Place Recognition Survey: An Update on Deep Learning Approaches (2021). arXiv preprint arXiv:2106.10458
10. Garg, S., Fischer, T., Milford, M.: Where is your place, visual place recognition? IJCAI 8, 4416–4425 (2021). https://doi.org/10.24963/ijcai.2021/603
11. Gadd, M., Newman, P.: Open-RadVLAD: fast and robust radar place recognition. In: IEEE Radar Conference (RadarConf). IEEE (2024)
12. Checchin, P., et al.: Radar Scan Matching Slam Using the Fourier-Mellin Transform. Springer Tracts in Advanced Robotics (2009)
13. Cen, S.H., Newman, P.: Radar-only ego-motion estimation in difficult settings via graph matching. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 298–304. IEEE (2019)
14. Hong, Z., Petillot, Y., Wang, S.: RadarSLAM: radar based large-scale slam in all weathers. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5164–5170. IEEE (2020)
15. Kim, G., Kim, A.: 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), pp. 4802–4809. IEEE (2018)
16. Kim, G., et al.: MulRan: multimodal range dataset for urban place recognition. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2020)
17. Wang, H., Wang, C., Xie, L.: Intensity scan context: coding intensity and geometry relations for loop closure detection. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2095–2101. IEEE (2020)
18. Kim, G., Choi, S., Kim, A.: Scan context++: structural place recognition robust to rotation and lateral variations in urban environments. IEEE Trans. Robot. 38(3), 1856–1874 (2021). https://doi.org/10.1109/tro.2021.3116424
19. Săftescu, Ș., et al.: Kidnapped radar: topological radar localisation using rotationally-invariant metric learning. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 4358–4364. IEEE (2020)
20. Komorowski, J., Wysoczanska, M., Trzcinski, T.: Large-scale topological radar localization using learned descriptors. In: Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II 28, pp. 451–462. Springer (2021)
21. De Martini, D., Gadd, M., Newman, P.: kRadar++: coarse-to-fine FMCW scanning radar localisation. Sensors 20(21), 6002 (2020). https://doi.org/10.3390/s20216002
22. Gadd, M., De Martini, D., Newman, P.: Look around you: sequence-based radar place recognition with learned rotational invariance. In: 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), pp. 270–276. IEEE (2020)
23. Wang, W., et al.: RadarLoc: learning to relocalize in FMCW radar. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2021)

24. Cait, K., Wang, B., Lu, C.X.: Autoplace: robust place recognition with single-chip automotive radar. In: 2022 International Conference on Robotics and Automation (ICRA), pp. 2222–2228. IEEE (2022)
25. Gadd, M., De Martini, D., Newman, P.: Contrastive learning for unsupervised radar place recognition. In: IEEE International Conference on Advanced Robotics (ICAR). IEEE (2021)
26. Gadd, M., et al.: OORD: The Oxford Offroad Radar Dataset (2024). arXiv preprint arXiv:2403.02845
27. Yin, H., et al.: Radar-to-lidar: heterogeneous place recognition via joint learning. Front. Robot. and AI 8 (2021). https://doi.org/10.3389/frobt.2021.661199
28. Park, Y.S., Kim, J., Kim, A.: Radar localization and mapping for indoor disaster environments via multi-modal registration to prior lidar map. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1307–1314. IEEE (2019). https://doi.org/10.1109/IROS40897.2019.8967633
29. Mason, K., et al.: Uncertainty-aware lidar place recognition in novel environments. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3366–3373. IEEE (2023)
30. Zaffar, M., Nan, L., Kooij, J.F.: On the estimation of image-matching uncertainty in visual place recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17743–17753. IEEE/CVF (2024)
31. Lajoie, P.-Y., Beltrame, G.: Self-supervised domain calibration and uncertainty estimation for place recognition. IEEE Rob. Autom. Lett. 8(2), 792–799 (2022). https://doi.org/10.1109/lra.2022.3232033
32. Claxton, O., et al.: Improving Visual Place Recognition Based Robot Navigation through Verification of Localization Estimates (2024). arXiv preprint arXiv:2407.08162
33. Barnes, D., Weston, R., Posner, I.: Masking by Moving: Learning Distraction-free Radar Odometry from Pose Information (2019). arXiv preprint arXiv:1909.03752
34. Pearce, T., Brintrup, A., Zhu, J.: Understanding Softmax Confidence and Uncertainty (2021). arXiv preprint arXiv:2106.04972
35. Adolfsson, D., et al.: TBV Radar SLAM – Trust but Verify Loop Candidates (2023). arXiv preprint arXiv:2301.04397
36. Aldera, R., et al.: What could go wrong? introspective radar odometry in challenging environments. In: IEEE Intelligent Transportation Systems Conference (ITSC). IEEE (2019)
37. Potter, L.C., et al.: Sparsity and compressed sensing in radar imaging. Proc. IEEE 98(6), 1006–1020 (2010). https://doi.org/10.1109/jproc.2009.2037526
38. Wang, X., et al.: Two-level block matching pursuit for polarimetric through-wall radar imaging. IEEE Trans. Geosci. Rem. Sens. 56(3), 1533–1545 (2017). https://doi.org/10.1109/tgrs.2017.2764920
39. Sun, S., Zhang, Y.D.: 4D automotive radar sensing for autonomous vehicles: a sparsity-oriented approach. IEEE J. Sel. Top. Signal Process. 15(4), 879–891 (2021). https://doi.org/10.1109/jstsp.2021.3079626
40. Wen, Y., et al.: A discriminative feature learning approach for deep face recognition. In: Computer Vision–ECCV 2016: 14th European Conference, amsterdam, the netherlands, October 11–14, 2016, Proceedings, Part VII 14, pp. 499–515. Springer (2016)
41. Liu, W., et al.: Large-margin softmax loss for convolutional neural networks. In: International Conference on Machine Learning, pp. 507–516. PMLR (2016)
42. Liu, W., et al.: SphereFACE: deep hypersphere embedding for face recognition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6738–6746. IEEE, IEEE Computer Society, Los Alamitos, CA, USA (2017). https://doi.org/10.1109/CVPR.2017.713
43. Gal, Y., et al.: Uncertainty in Deep Learning. Ph.D. thesis. University of Cambridge, Cambridge (2016)
44. Salimans, T., Kingma, D.P.: Weight normalization: a simple reparameterization to accelerate training of deep neural networks. Adv. Neural Inf. Process. Syst. 29 (2016)
45. Guo, C., et al.: On calibration of modern neural networks. In: International Conference on Machine Learning, pp. 1321–1330. PMLR (2017)
46. Rahaman, R., et al.: Uncertainty quantification and deep ensembles. Adv. Neural Inf. Process. Syst. 34, 20063–20075 (2021)
47. Maddern, R., et al.: 1 Year, 1000 km: the oxford RobotCar dataset. Int. J. Robot Res. 36(1), 3–15 (2017). https://doi.org/10.1177/0278364916679498
48. Arandjelovic, R., et al.: NetVLAD: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016)
49. He, K., et al.: Deep Residual Learning for Image Recognition (2015). CoRR abs/1512.03385
50. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR). Springer, San Diega, CA, USA (2015)
51. Cai, K., Lu, C.X., Huang, X.: STUN: self-teaching uncertainty estimation for place recognition. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6614–6621 (2022). https://doi.org/10.1109/IROS47612.2022.9981546
52. Olson, E.: M3RSM: many-to-many multi-resolution scan matching. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5815–5821. IEEE (2015)

How to cite this article: Agarwal, S., et al.: *Bayesian Radar Cosplace*: Directly estimating location uncertainty in radar place recognition. IET Radar Sonar Navig. e70002 (2025). https://doi.org/10.1049/rsn2.70002