Radar Odometry Combining Probabilistic Estimation and Unsupervised Feature Learning

Keenan Burnett*, David J. Yoon*, Angela P. Schoellig, and Timothy D. Barfoot

Institute for Aerospace Studies, University of Toronto

Abstract—This paper presents a radar odometry method that combines probabilistic trajectory estimation and deep learned features without needing groundtruth pose information. The feature network is trained unsupervised, using only the on-board radar data. With its theoretical foundation based on a data likelihood objective, our method leverages a deep network for processing rich radar data, and a non-differentiable classic estimator for probabilistic inference. We provide extensive experimental results on both the publicly available Oxford Radar RobotCar Dataset and an additional 100 km of driving collected in an urban setting. Our sliding-window implementation of radar odometry outperforms most hand-crafted methods and approaches the current state of the art without requiring a groundtruth trajectory for training. We also demonstrate the effectiveness of radar odometry under adverse weather conditions. Code for this project can be found at: https://github.com/utiasASRL/hero_radar_odometry

I. INTRODUCTION

While the reliability of autonomous vehicles continues to improve, operating in rain and snow remains a challenge. Most autonomous driving systems rely primarily on a combination of cameras and lidar for perception, with radar sensors taking a back-seat role [13]. Cen and Newman [15] presented a successful application of a scanning radar sensor to large-scale outdoor ego-motion estimation. Their work has inspired a resurgence of research into radar-based perception and navigation systems. Compared to lidar, radar is more robust to precipitation due to its longer wavelength. For this reason, radar may be a key to enabling self-driving vehicles to operate in adverse weather. The ultimate goal of this research is to approach the performance of lidar-based algorithms in ideal conditions and surpass them in adverse conditions.

Previous works in this area have made significant progress towards radar-based odometry [2–4, 9, 11, 14–16, 26, 34, 39] and place recognition [20, 22, 31, 43, 46]. However, previous approaches to radar odometry have either relied on hand-crafted feature extraction [2–4, 14–16, 26, 34], correlative scan matching [11, 39], or a (self-)supervised learning algorithm [9, 11] that relies on trajectory groundtruth. Barnes and Posner [9] previously showed that learned features have the potential to outperform hand-crafted features. On the other hand, it has not yet been shown whether correlative scan matching systems can be scaled up to large-scale mapping and localization. In order to address these limitations, we propose a method that is able to learn features directly from radar data without relying on groundtruth pose information.

In this work, we present an unsupervised radar odometry pipeline that approaches the state of the art as reported on the Oxford Radar RobotCar Dataset [10]. Our network parameters
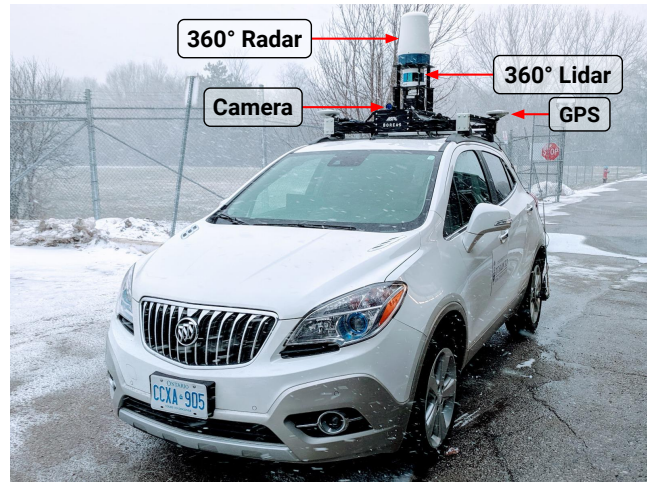

Fig. 1. Our data-taking platform, *Boreas*, which includes a Velodyne Alpha-Prime (128-beam) lidar, Navtech CIR204-H radar, FLIR Blackfly S monocular camera, and Applanix POSLV GNSS.

are trained using only the on-board radar sensor, alleviating the need for an accurate groundtruth trajectory. To our knowledge, this is the first example of a totally unsupervised radar odometry pipeline. We show additional experimental results on 100 km of radar data collected on the data-taking platform shown in Figure 1. We also provide a comparison of radar odometry performance in ideal and harsh weather conditions.

Our approach is based on the Exactly Sparse Gaussian Variational Inference (ESGVI) parameter learning framework of Barfoot et al. [8], a nonlinear batch state estimation framework that provides a family of scalable estimators from a variational objective. Model parameters can be optimized jointly with the state using a data likelihood objective. Yoon et al. [51] recently applied the framework to train a deep network, demonstrating feature learning for lidar odometry with only on-board lidar data. We extend their methodology and apply it to radar odometry. Our approach is modular, enabling the use of modern deep learning tools and classical estimators with minimal interface requirements. Importantly, our method does not require the estimator to be differentiable, a limitation of some other learning-based methods [11].

The hybridization of deep learning and probabilistic state estimation allows us to achieve the best of both worlds. Deep learning can be leveraged to process rich sensor data, while classical estimators can be used to deal with probabilities and out-of-distribution samples through outlier rejection schemes, motion priors, and other estimation tools. Furthermore, classical estimators make the incorporation of additional sensors and constraints relatively straightforward.

---

* Equal contribution.

We review related work in Section II. An overview of ESGVI parameter learning is presented in Section III, and the methodology of our radar odometry is in Section IV. Experimental results are presented in Section V. Concluding remarks and future work are in Section VI.

II. RELATED WORK

Initially, radar-based navigation was reliant on hand-tuned detectors specifically designed for radar [19, 29]. One such detector is CFAR (Constant False Alarm Rate), which uses a sliding window to find relative peaks in radar data [41]. Although CFAR works well for detecting aircraft against a blank sky, it struggles to discriminate between clutter and objects of interest on the ground where the relative differences in radar cross section are low. These early methods relied primarily on location information to match detections and were thus limited by the accuracy of their data association.

Subsequent works have sought to improve feature detection and matching. Checchin et al. [18] side-stepped this problem by applying correlative scan matching between entire radar scans via the Fourier-Mellin transform. Although correlation-based techniques have been shown to achieve excellent odometry performance, it remains challenging to apply them to large-scale mapping and localization due in part to their high data storage requirements during mapping.

Recent works have taken advantage of improvements to frequency-modulated continuous wave (FMCW) radar sensors. The radar sensor produced by Navtech [1] is popular for navigation research due to its combination of high range resolution, angular resolution, range, and $360°$ field of view.

Cen and Newman [15] demonstrated large-scale outdoor ego-motion estimation with a scanning radar. Their work inspired a resurgence of research into radar-based perception and navigation systems. Research in this area has also been accelerated by the introduction of the Oxford Radar RobotCar Dataset [10]. Other datasets that feature a scanning radar include the MulRan Dataset [31] for multimodal place recognition, and the RADIATE Dataset [44] for object detection and tracking in adverse weather.

Cen and Newman [16] later provided an update to their radar odometry pipeline with an improved gradient-based feature detector along with a new graph matching strategy. Aldera et al. [3] learn an attention policy to down-sample the number of measurements provided to the data association stage, thus speeding up the odometry pipeline. Aldera et al. [4] train a classifier on the principal eigenvector of their graph matching problem in order to predict and correct for failures in radar odometry. Park et al. [39] applied the Fourier-Mellin Transform to log-polar images computed from down-sampled Cartesian images. Barnes et al. [11] demonstrated a fully differentiable, correlation-based radar odometry pipeline. Their approach learns a binary mask to remove distractor features before using brute force search to find the pose with the minimum cross correlation. Hong et al. [26] used vision-based features and graph matching to demonstrate the first radar-based SLAM system to operate in adverse weather. Burnett et al. [14] demonstrated that motion distortion and

Doppler effects can have an adverse effect on radar-based navigation, but that these effects can be compensated using continuous-time estimation techniques.

These works represent significant steps towards radar-based autonomy. However, each of these methods relies on either hand-crafted feature detectors and descriptors or cumbersome phase correlation techniques. Barnes and Posner [9] showed that learned features can result in superior radar odometry performance. Their work currently represents the state of the art for point-based radar odometry. Despite these results, their approach requires the estimator to be differentiable and they require groundtruth poses as a supervisory signal. Our approach does not suffer from either of these drawbacks.

Other works have focused on radar-based place recognition. Săftescu et al. [43] perform place recognition by learning rotationally invariant descriptors within a metric feature space. Gadd et al. [22] improve on this work by incorporating sequence-based place recognition based on SeqSLAM [37]. De Martini et al. [20] proposed a two-stage system to integrate topological localization with metric pose estimation. A related avenue of research has been to localize radar scans using existing satellite imagery [46] [47]. Further research has tackled radar-based perception through occupancy [48], traversability [12, 49], and semantic segmentation [30].

Barfoot et al. [8] presented the ESGVI framework and showed that model parameters can be jointly optimized along with the state using an Expectation-Maximization (EM) iterative optimization scheme on a data likelihood objective. In the E-step, model parameters are held fixed, and the state is optimized. In the M-step, the state distribution is held fixed, and the model parameters are optimized. This idea originates from a line of work that applied EM for linear system identification [24, 45]. Ghahramani and Roweis [25] extended the idea to simple nonlinearities approximated with Gaussian radial basis functions. Wong et al. [50] first applied EM under the ESGVI framework to large-scale trajectory estimation by learning noise models robust to outliers. Yoon et al. [51] demonstrated the first application of ESGVI parameter learning to a deep network, where lidar features for odometry were learned without groundtruth. We apply their methodology and adapt it for radar odometry.

Our framework shares similarities with the Variational Autoencoder (VAE) [33] framework, as both start from a data likelihood objective and optimize the Evidence Lower Bound (ELBO). In contrast to the VAE, which approximates latent state inference with a network, our framework applies classic state estimation (e.g., factor graph optimization). Extensions to the VAE for problems with graphical structure exist [28], but our method makes use of existing estimation tools familiar to the robotics field. Most similar to our framework is the work of DeTone et al. [21]. They present a self-supervised visual odometry framework that has a deep network frontend trained according to a bundle adjustment backend. Compared to their framework, ours is based on a probabilistic objective and can handle uncertainty in the posterior estimates.

III. EXACTLY SPARSE GAUSSIAN VARIATIONAL INFERENCE PARAMETER LEARNING

This section summarizes parameter learning in the ESGVI framework as presented in prior work [8, 50, 51]. The loss function is the negative log-likelihood of the observed data,

$$\mathscr{L} = -\ln p(\mathbf{z}|\boldsymbol{\theta}), \qquad (1)$$

where $\mathbf{z}$ is the data (e.g., radar measurements), and $\boldsymbol{\theta}$ are the parameters (e.g., network parameters). We apply the usual EM[1] decomposition after introducing the latent trajectory, $\mathbf{x}$, which is written as

$$\mathscr{L} = \underbrace{\int_{-\infty}^{\infty} q(\mathbf{x})\ln\left(\frac{p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta})}{q(\mathbf{x})}\right)d\mathbf{x}}_{\leq 0} - \underbrace{\int_{-\infty}^{\infty} q(\mathbf{x})\ln\left(\frac{p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{x})}\right)d\mathbf{x}}_{\text{upper bound}},$$
$$(2)$$

where we define our posterior approximation as a multivariate Gaussian distribution, $q(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

We work with the upper bound term, commonly referred to as the (negative) ELBO. Applying the definition of entropy of a Gaussian and dropping constants, we can rewrite the term as the ESGVI loss functional,

$$V(q|\boldsymbol{\theta}) = \mathbb{E}_q[\phi(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})] + \frac{1}{2}\ln\left(|\boldsymbol{\Sigma}^{-1}|\right), \qquad (3)$$

where $\mathbb{E}[\cdot]$ is the expectation operator, $|\cdot|$ is the matrix determinant, and we define the joint factor $\phi(\mathbf{x},\mathbf{z}|\boldsymbol{\theta}) = -\ln p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})$.

We apply Generalized EM (GEM) [38], a variation of EM that does not run the M-step to completion (convergence), to gradually optimize the data likelihood, $\mathscr{L}$. In the E-step, we hold $\boldsymbol{\theta}$ fixed and optimize $V(q|\boldsymbol{\theta})$ for $q(\mathbf{x})$. In the M-step, we hold $q(\mathbf{x})$ fixed and optimize $V(q|\boldsymbol{\theta})$ for $\boldsymbol{\theta}$. When the expectation over the posterior, $q(\mathbf{x})$, is approximated at the mean of the Gaussian, the E-step is the familiar Maximum A Posteriori (MAP) estimator [8].

IV. UNSUPERVISED DEEP LEARNING FOR RADAR ODOMETRY

A. Problem Definition

In this subsection, we summarize the sliding-window odometry formulation presented by Yoon et al. [51]. The state we estimate at time t_k is $\mathbf{x}_k = \{\mathbf{T}_{k,0}, \boldsymbol{\varpi}_k\}$, where the pose $\mathbf{T}_{k,0} \in SE(3)$ is a transformation between frames at t_k and t_0, and $\boldsymbol{\varpi}_k \in \mathbb{R}^6$ is the body-centric velocity[2]. We optimize a sliding-window of w frames, $t_\tau, \ldots, t_{\tau+w-1}$, where each state has a corresponding radar scan. The first pose of the window, $\mathbf{T}_{\tau,0}$, is locked (not optimized) and treated as the reference frame for keypoint matching.

Our joint factor, $\phi(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})$, splits into motion prior factors and measurement factors. Figure 2 shows an example factor graph illustration. We write $\phi(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})$ as

$$\sum_{k=\tau+1}^{\tau+w-1}\left(\phi^p(\mathbf{x}_{k-1},\mathbf{x}_k) + \sum_{\ell=1}^{L_k}\phi^m(\mathbf{z}_k^\ell,\mathbf{r}_\tau^\ell|\mathbf{x}_\tau,\mathbf{x}_k,\boldsymbol{\theta})\right), \quad (4)$$

---

[1]While the acronym stays the same, we work with the negative log-likelihood and are technically applying Expectation Minimization.

[2]Despite the radar being a 2D sensor, we formulate our problem in $SE(3)$ to be compatible with other 3D sensor modalities.
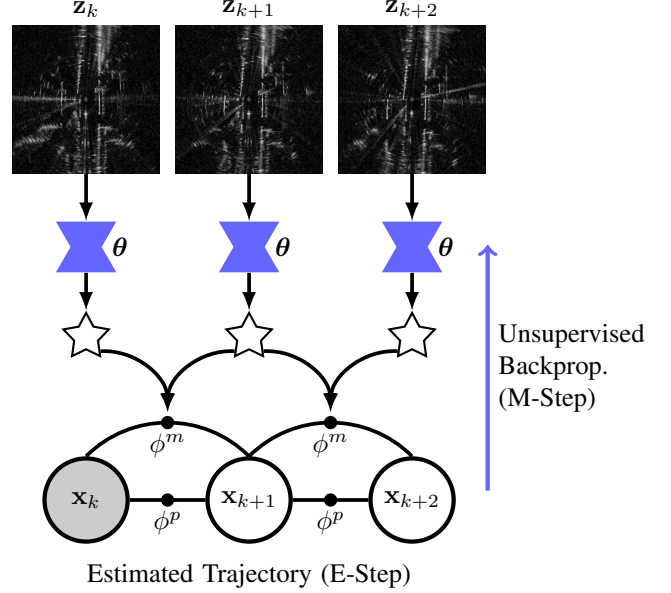

Fig. 2. This figure depicts the factor graph of our radar odometry pipeline. $\mathbf{x}_k$ and $\mathbf{z}_k$ are defined as the state of the vehicle and the radar scan at time t_k, respectively. The vehicle trajectory is estimated over a sliding window of w frames, where $w = 3$ in this figure. The deep network parameters are denoted by $\boldsymbol{\theta}$. The output of the network is a set of feature locations (x_i, y_i), their associated inverse covariance matrices, $\mathbf{W}_i$, and their learned descriptors, $\mathbf{d}_i$, which are together represented by stars in the diagram. These features are then matched between pairs of frames using a differentiable softmax matcher. The matched features are then used to form measurement factors, ϕ^m. A white-noise-on-acceleration motion prior is applied to create prior factors, ϕ^p.

where $\mathbf{z}_k^\ell$ is the ℓth keypoint measurement in frame k, which has a total of L_k keypoints, and $\mathbf{r}_\tau^\ell$ is its matched point from frame τ.

For the motion prior factors, ϕ^p, we apply a white-noise-on-acceleration prior as presented by Anderson and Barfoot [5], which is defined by the following kinematic equations:

$$\begin{aligned}\dot{\mathbf{T}}(t) &= \boldsymbol{\varpi}(t)^\wedge \mathbf{T}(t), \\ \dot{\boldsymbol{\varpi}} &= \mathbf{w}(t), \quad \mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c\delta(t - t'))\end{aligned} \qquad (5)$$

where $\mathbf{T}(t_k) = \mathbf{T}_{k,0}$, and $\mathbf{w}(t) \in \mathbb{R}^6$ is a zero-mean, white-noise Gaussian process. The operator, $\wedge$, transforms an element of $\mathbb{R}^6$ into a member of Lie algebra, $\mathfrak{se}(3)$ [7]:

$$\begin{bmatrix}\mathbf{u}\\\mathbf{v}\end{bmatrix}^\wedge = \begin{bmatrix}\mathbf{v}^\wedge & \mathbf{u}\\\mathbf{0}^T & 0\end{bmatrix}, \quad \mathbf{v}^\wedge = \begin{bmatrix}v_1\\v_2\\v_3\end{bmatrix}^\wedge = \begin{bmatrix}0 & -v_3 & v_2\\v_3 & 0 & -v_1\\-v_2 & v_1 & 0\end{bmatrix}, \quad (6)$$

where $\wedge$ is also overloaded as the skew-symmetric operator for elements of $\mathbb{R}^3$.

The measurement factors, ϕ^m, are of the form:

$$\phi^m(\mathbf{z}_k^\ell,\mathbf{r}_\tau^\ell|\mathbf{x}_\tau,\mathbf{x}_k,\boldsymbol{\theta}) = \frac{1}{2}{\mathbf{e}_k^\ell}^T\mathbf{W}_k^\ell\mathbf{e}_k^\ell - \ln\left|\mathbf{W}_k^\ell\right|, \quad (7)$$

$$\mathbf{e}_k^\ell = \mathbf{D}\left(\mathbf{z}_k^\ell - \mathbf{T}_{k,0}\mathbf{T}_{0,\tau}\mathbf{r}_\tau^{\ell_k}\right), \qquad (8)$$

where we use the log-likelihood of a Gaussian as the factor, and $\mathbf{D}$ is a 3×4 constant projection matrix that removes the homogeneous element. The homogenous keypoint, $\mathbf{z}_k^\ell$, its point match, $\mathbf{r}_\tau^{\ell_k}$, and its inverse covariance (weight) matrix, $\mathbf{W}_k^\ell$, are quantities that depend on the network parameters, $\boldsymbol{\theta}$.
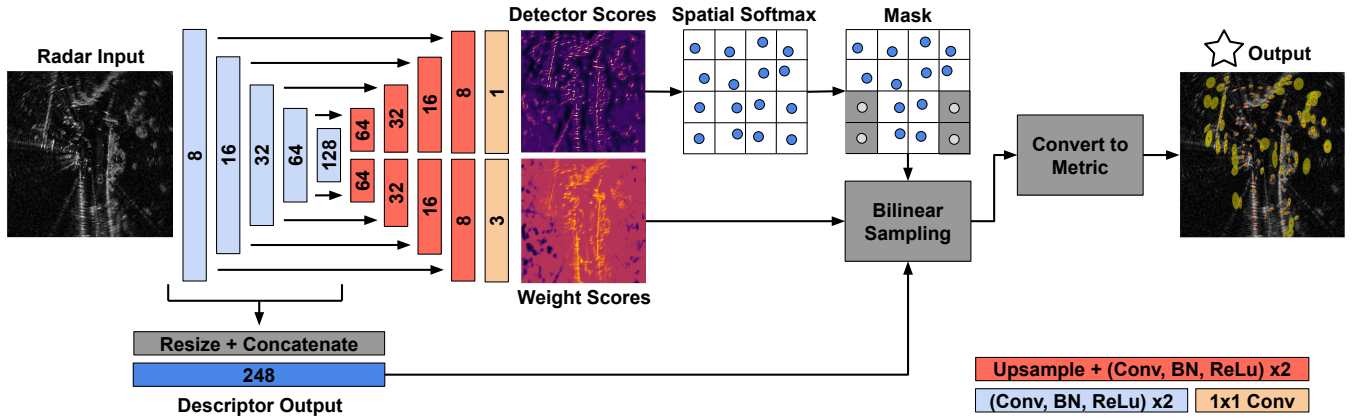
Fig. 3. We based our network on the architecture presented by Barnes and Posner [9]. The network outputs detector scores for keypoint detection, weight scores predicting keypoint uncertainty, and descriptors for matching. The weight scores are composed into 2×2 inverse covariance matrices (see (11), the corresponding image is the log-determinant). Descriptors are the concatenation of all encoder layer outputs after resizing via bilinear interpolation. The encoder and decoder layers are a double application of a 3×3 convolution, batch normalization, and ReLU nonlinearity. The layer sizes vary by a factor of 2 through max-pooling (encoder) and bilinear upsampling (decoder). Note that the output size is the same as the input, and are visually smaller in the interest of space. An output 1×1 convolution is applied for the detector and weight scores. The detector score map is partitioned into uniform cells, where a spatial softmax and weighted summation of coordinates are applied to yield a keypoint for each cell. Corresponding weights and descriptors are obtained via bilinear sampling.

B. Network

Our network is based on the architecture presented by Barnes and Posner [9], which is a U-Net [42] style convolutional encoder-multi-decoder architecture to output radar keypoints, weights, and descriptors. The input to the network is a 2D Cartesian projection of the polar radar data. Example radar scans are shown at the top of Figure 2.

An illustration of our network architecture is shown in Figure 3. A dense descriptor map is created by resizing the output of each encoder block before concatenation into a 248-channel tensor. In our approach, the weight score is a 3-channel tensor and the detector score is a 1-channel tensor. The detector score tensor is then partitioned into ($N = 400$) equally sized cells, with each producing a candidate 2D keypoint. A spatial softmax within each cell, followed by a weighted summation of pixel coordinates produces the image-space keypoint coordinates of each cell. The corresponding descriptor and weight score vectors are bilinearly sampled using the keypoint coordinates. The image coordinates are converted to metric coordinates using the known m/pixel resolution. Finally, we formulate the 3D homogenous keypoint, $\mathbf{z}_k^\ell$, by appending a 0 as a third coordinate, and a 1 as the fourth homogenous element.

Our detector produces a candidate keypoint for each square cell in a uniformly partitioned radar image. This leads to candidate keypoints in regions of the image that are void of data (i.e., black regions of a radar image). As we are training without groundtruth, we found it necessary in practice to mask (reject) these keypoints. We threshold each azimuth of the polar radar scan by a scalar multiple, $\beta = 3$, of its mean intensity, where exceeding the threshold is considered valid. Projecting the result into Cartesian space produces a binary mask of valid pixels. We then threshold on the ratio of valid pixels in each square cell for robustness to noise. Keypoints belonging to cells with less than 5% valid pixels are rejected.

A dense match between each keypoint descriptor and the reference descriptor tensor is applied with a softmax to preserve differentiability. We compute the dot product between each keypoint descriptor and all descriptors of the reference:

$$\mathbf{c}_k^{\ell T} = \mathbf{d}_k^{\ell T} \begin{bmatrix} \mathbf{d}_\tau^1 & \cdots & \mathbf{d}_\tau^N \end{bmatrix}, \tag{9}$$

where $\mathbf{d}_k^\ell$ is the descriptor vector of keypoint $\mathbf{z}_k^\ell$, and $\mathbf{d}_\tau^1, \dots, \mathbf{d}_\tau^N$ are the descriptor vectors of the reference. We apply a softmax on $\mathbf{c}_k^\ell$ and compute a weighted summation. The reference match for keypoint $\mathbf{z}_k^\ell$ is therefore

$$\mathbf{r}_\tau^{\ell_k} = \begin{bmatrix} \mathbf{p}_\tau^1 & \cdots & \mathbf{p}_\tau^N \end{bmatrix} \times \text{softmax}(T\mathbf{c}_k^\ell), \tag{10}$$

where $T = 100$ is a softmax temperature constant, $\mathbf{p}_\tau^1, \dots, \mathbf{p}_\tau^N$ are the homogeneous reference coordinates, and $\mathbf{p}_\tau^n \in \mathbb{R}^4$. The weight score vectors of each keypoint are assembled into matrices with the following decomposition [35, 51]:

$$\mathbf{W}_k^\ell = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & c \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ d_3 & 1 \end{bmatrix} \begin{bmatrix} \exp d_1 & 0 \\ 0 & \exp d_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ d_3 & 1 \end{bmatrix}^T, \tag{11}$$

where (d_1, d_2, d_3) is the weight score vector corresponding to keypoint $\mathbf{z}_k^\ell$, and $c = 10^4$ is a constant corresponding to the (inverse) variance of the third coordinate (which will always be 0). Visualizations of the learned keypoint covariances, $\mathbf{R}^{-1}$, are shown in Figure 4 as uncertainty ellipses.

C. Training and Inference

We follow the training methodology of Yoon et al. [51], which blends the GEM iterative optimization scheme required by ESGVI with Stochastic Gradient Descent (SGD) applied in conventional network training. The E-step, which optimizes for the current best posterior estimate, $q(\mathbf{x})$, is simply included as part of the forward propagation routine.
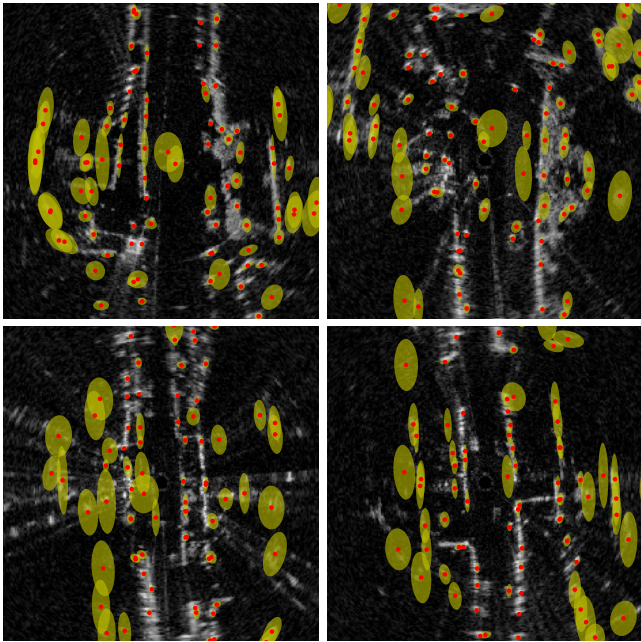
Fig. 4. A visualization of keypoints (red) on radar images with 5 standard deviation uncertainty ellipses (yellow). Many keypoints have elongated uncertainties consistent to the scene geometry, and as expected, keypoints further away from the sensor (image center) are more uncertain.

Windows of radar scans are randomly sampled as a minibatch of data. Forward propagation is summarized as:

1) Traditional forward propagation of the network to output radar features (see Section IV-B).
2) Construct the motion prior factors, ϕ^p, and measurement factors, ϕ^m, (see Section IV-A and IV-B).
3) Batch inference for the current best posterior estimate $q(\mathbf{x})$ for each window (E-step).

We approximate the E-step with the Gauss-Newton algorithm, which involves approximations to the Hessian and approximating the expectation in (3) at only the mean of the posterior.

The M-step is network backpropagation on the loss functional (3), which only applies to the measurement factors since the motion prior factors are constant with respect to the parameters, $\boldsymbol{\theta}$. Similar to the E-step, we approximate the expectation at only the mean of the posterior. Intuitively, we are using our best-guess trajectory as a supervisory signal to then carry out standard SGD to train the feature network.

D. Outlier Rejection

We apply the Geman-McClure robust cost function on the measurement factors, ϕ^m, in the E-step for outlier rejection. For the M-step, we follow Yoon et al. [51] and apply a constant threshold, α, on the squared Mahalanobis distance of the measurement factor with the current best posterior estimate,

$$\mathbf{e}_k^{\ell\, T} \mathbf{W}_k^\ell \mathbf{e}_k^\ell > \alpha, \tag{12}$$

where $\mathbf{e}_k^\ell$ is as defined in (8). We do not backpropagate keypoint matches that are greater than $\alpha = 16$.

After training, we improved odometry performance by rejecting keypoints with inverse covariances that have a log determinant, $\log|\mathbf{R}|$, less than a threshold ($\eta = 4.0$). We also use RANSAC at test time and only estimate on the inliers.

Methods	Supervision	Trans. Error (%)	Rot. Error (deg/1000m)
UnderTheRadar [9]	Supervised (L)	2.0583	6.7
Cen RO [15]	Unsupervised (HC)	3.7168	9.5
MC-RANSAC [14]	Unsupervised (HC)	3.3190	10.93
CFEAR [2]	Unsupervised (HC)	**1.76**	**5.0**
HERO (Ours)	Unsupervised (L)	1.9879	6.524

V. EXPERIMENTAL RESULTS

A. Experiment Setup

We evaluate our approach, Hybrid-Estimate Radar Odometry (HERO), on the publicly available Oxford Radar RobotCar Dataset [10] and on our own dataset, collected using the data-taking platform in Figure 1. The Oxford dataset is divided into 32 sequences, each approximately 10 km in length. We follow existing work [9] by training on 24 sequences, validating on 1 sequence, and testing on the same 7 sequences as [9].

Our implementation is a hybrid between Python and C++, where network-related code is implemented in Python using PyTorch [40], and estimation-related code is implemented using STEAM[3], an open-source C++ estimation library. When training the network, we use a fixed random seed and the Adam optimizer [32] with a learning rate of 1×10^{-5} and a mini-batch size of 1 window ($w = 4$) for up to 100k iterations. The Navtech radar sensor used in the Oxford dataset has a range resolution of 0.0438 m/bin. To minimize projection errors, we chose a Cartesian resolution of 0.2628 m/pixel to be an integer multiple of the radar resolution. The Cartesian radar images are made to be square with a width of 640 pixels. For the spatial softmax operation, each cell is 32×32 pixels, resulting in 400 total keypoints. We use a softmax temperature of 100. For the motion prior, $\mathbf{Q}_c$ is made to be diagonal. The entries of $\mathbf{Q}_c^{-1}$ can be thought of as penalty terms on body-centric linear and angular acceleration. It is possible to learn $\mathbf{Q}_c$ as is done [50]. However, in our implementation, the values are hand-tuned. Similar to Barnes and Posner [9], we augment our training data with random rotations of up to 0.26 radians. Our sliding-window implementation[4] takes on average 0.07 seconds, 0.13 seconds, and 0.18 seconds for window sizes of 2, 3, and 4, respectively. Since the radar sensor spins at 4 Hz, our current implementation is real-time capable. During evaluation, we use the timestamp of each measurement to do continuous-time estimation with STEAM to compensate for motion distortion. For more information on this approach, we refer readers to the work of Sean Anderson et al. [5] [6].

B. Oxford Radar RobotCar Dataset

Here, we compare our radar odometry results with other point-based radar odometry methods including Cen and Newman [15] (Cen RO), Barnes and Posner [9] (Under the Radar), CFEAR [2], and Burnett et al. [14] (MC-RANSAC). Following these previous works, we report our results using the KITTI odometry metrics [23], which average the relative

---

[3]https://github.com/utiasASRL/steam
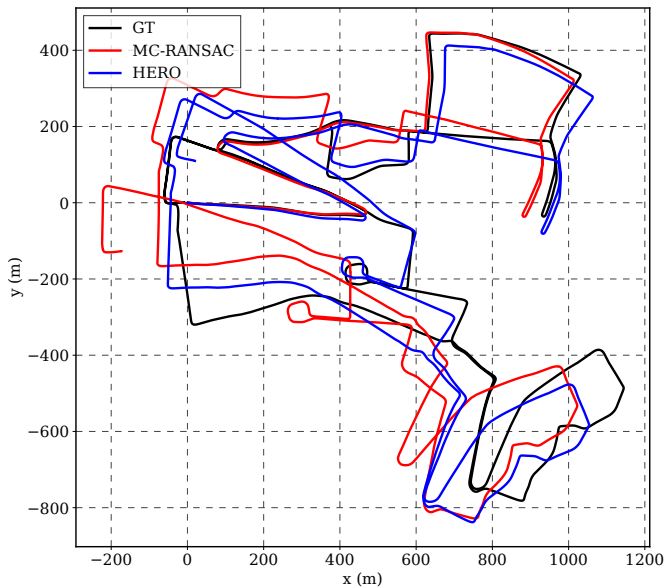[4]On an Nvidia Tesla V100 GPU and 2.2 GHz Intel Xeon CPU.

Fig. 5. This figure depicts the results of our unsupervised Hybrid-Estimate Radar Odometry (HERO) in blue alongside the results of a hand-crafted radar odometry pipeline based on motion compensated RANSAC (MC-RANSAC) [14] in red. The groundtruth trajectory is plotted in black (GT). Sequence: 2019-01-10-14-02-34-radar-oxford-10k.

TABLE II
Ablation study. Results for sequence 2019-01-10-14-02-34-radar-oxford-10k.

Configuration	Translational Error (%)	Rotational Error (deg/1000m)
Baseline	2.262	7.601
Scalar Weight	2.484	7.997
No Mah. Threshold	3.083	9.300
No Masking	3.203	10.07
No Augmentation	5.054	16.60
Window Size 2	2.488	7.901
Window Size 3	2.393	7.657
Cart Res: 0.2160	2.396	7.602
Cart Res: 0.3024	2.407	7.927
Baseline + $(\log|\mathbf{R}| < 4.0)$	1.953	6.532

position and orientation errors over every sub-sequence of length $(100\,\text{m}, 200\,\text{m}, \cdots, 800\,\text{m})$. The results in Table I show that our method, HERO, is competitive with other unsupervised radar odometry methods, surpassing the hand-crafted algorithms Cen RO and MC-RANSAC. In addition, our method exceeds the performance of Under the Radar, a learned point-based radar odometry approach, without needing any groundtruth supervision. This feature gives our method an advantage for deployment in regions where a source of high-quality groundtruth is unavailable. Sources of groundtruth, such as a GPS/INS system, are not only costly, but require a clear line of sight to the sky for GPS reception. Notably, the authors of the Oxford Radar RobotCar Dataset [9] state that the accuracy of their GPS/INS system varied significantly due to poor GPS reception [36]. They addressed this issue by including visual odometry and loop closures into a large scale optimization [9]. Figure 5 illustrates an example sequence comparing our odometry method to MC-RANSAC.

We provide an ablation study of our method in Table II. We first compare our baseline to the following variations:

- **Scalar Weight**: Instead of learning a 2×2 weight (inverse covariance) matrix (see (11)), we learn a scalar weight.
- **No Mah. Threshold**: We do not apply the outlier rejection threshold on the squared Mahalanobis distance 12.
- **No Masking**: We do not mask keypoints in void regions of the radar images, as described in Section IV-B.
- **No Augmentation**: We do not augment our training data with random rotations.

We also show the effect of varying the window size of our sliding-window optimization and the resolution of the projected radar images. Our baseline uses a window size of 4 and a resolution of 0.2628 m/pixel.

The results in Table II suggest that rotation augmentation,

masking keypoints, and thresholding on Mahalanobis distance were the most significant hyperparameters. We theorize that because a large fraction of the Oxford dataset consists of either driving straight or waiting at a red light, training without rotation augmentation makes it less likely that the network will learn to properly match features across large frame-to-frame rotations. Putting a hard threshold on the Mahalanobis distance before backpropagation allows us to only backpropagate the errors from likely inliers. The masking of keypoints from empty regions of the input radar scan was necessary to achieve good odometry results. Our interpretation is that adding this additional structure to the learning problem is needed for the unsupervised model to succeed.

C. Additional Experiments on the Boreas Dataset

In this section, we provide additional experimental results for our radar odometry using 100 km of driving obtained in an urban environment[5]. The dataset was collected using the platform shown in Figure 1, which includes a Velodyne Alpha-Prime (128-beam) lidar, Navtech CIR204-H radar, FLIR Blackfly S monocular camera, and Applanix POSLV GNSS. This Navtech sensor has a range resolution of 0.0596 m and a total range of 200 m, compared to the 0.0438 m resolution and 163 m range of the Navtech sensor used in the Oxford dataset. We do not require groundtruth position data to train our network, but obtain it for the test sequences in order to calculate KITTI odometry drift metrics. Groundtruth positioning is obtained from the post-processed GNSS results and is accurate to within 12 cm.

The 100 km of total driving data is divided into 11 individual sequences, each approximately 9 km in length. We use 7 sequences for training, 1 for validation, and 3 for testing. Of the 3 test sequences, two were taken during a snow storm, and the other was taken on a sunny day.

Figure 6 illustrates the large differences in weather conditions. Between the sunny and snowy days, there were significant visual appearance changes. Most of the lane lines are not visible during the snow storm. Even more stark is the contrast between the lidar data taken on the two days (after the ground plane is removed). During the snow storm, the lidar scan is littered with a large number of detections associated with snow flakes. Although we do not provide experimental results for this, it seems probable that the accuracy of a lidar odometry

---

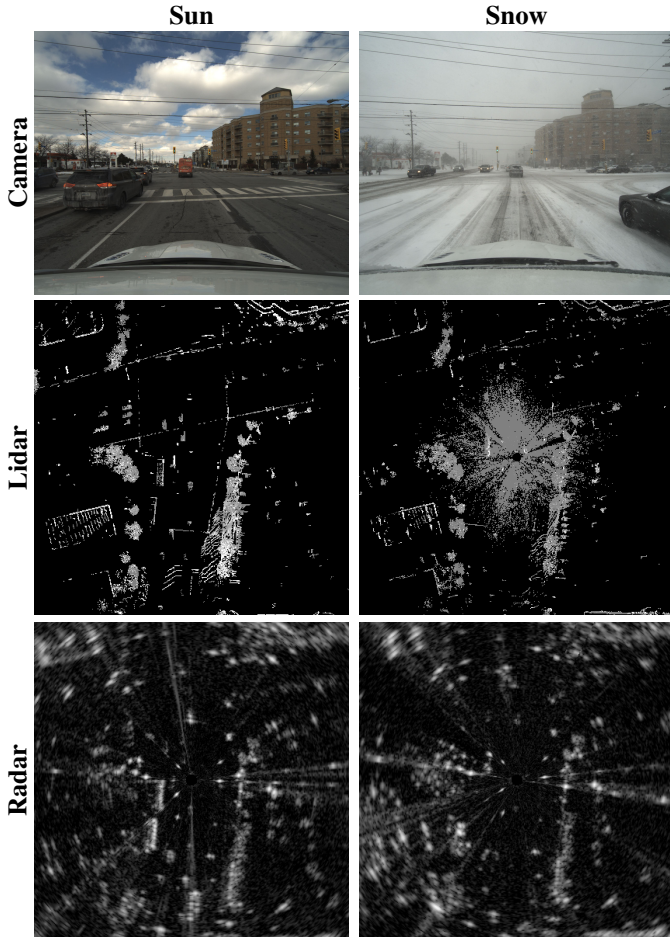[5]We plan on making our dataset publicly available in the next year.

Fig. 6. This figure illustrates the differences in camera, lidar, and radar data between a sunny day and a snow storm. During snowfall, the lidar sensor becomes littered with noisy snowflake detections but the radar data appears unperturbed.

Fig. 7. Odometry results on the snowy sequence 2021-01-26-11-22.

TABLE III
Boreas test results.

Sequence	Weather	Trained On	Trans. Error (%)	Rot. Error (deg/1000m)
01-26-10-59	Snow	Boreas	2.003	5.599
01-26-11-22	Snow	Boreas	1.980	5.288
02-09-12-55	Sun	Boreas	2.073	5.886
01-26-10-59	Snow	Oxford	2.355	8.137
01-26-11-22	Snow	Oxford	2.112	6.442
02-09-12-55	Sun	Oxford	2.546	8.911

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we applied the ESGVI parameter learning framework to learn radar features for odometry using only the on-board radar data. Our odometry performance on the Oxford Radar RobotCar Dataset approaches the current state of the art, which is a hand-crafted method [2]. We provided additional experimental results on 100 km of driving taken in an urban setting. Within this dataset, we demonstrated the effectiveness of radar odometry during heavy snowfall.

In recent years, robotics research has become increasingly reliant on high-quality datasets for training deep learning algorithms. The size and scope of these datasets is a limiting factor in the performance of many robotic systems. However, collecting and annotating these datasets is an expensive and labour-intensive process. For this reason, it is important that research in robotics trend towards solutions that are more data efficient. By foregoing the need for groundtruth pose information, our unsupervised architecture enables large quantities of training data to be collected with relative ease.

Our framework is a hybrid of modern deep learning and classic probabilistic state estimation. Deep learning can be used to process rich sensor data while probabilistic estimation can be used to handle out-of-distribution samples through outlier rejection schemes. Furthermore, our modular framework enables many future extensions and avenues of research. For example, we plan on incorporating IMU factors into our estimator, which should be straightforward with our probabilistic estimator.

system would suffer under these conditions. However, Charron et al. [17] have shown that it is possible for these effects to be compensated. Unsurprisingly, it is difficult to discern the difference between the radar scans, other than the movement of large vehicles.

The robustness of radar to inclement weather is also exemplified in the odometry drift results reported in Table III. In fact, the drift rates on the snowy days were lower than on the sunny day. We postulate that the error increases on the sunny day due to the increase in the vehicle's speed. It is interesting to note that our rotation drift metrics here are lower than on the Oxford dataset. While it is difficult to point to a single factor as the reason for this difference in performance, we note that our trajectory involves less turns and we use a different source of groundtruth for testing. In Figure 7, we provide a plot of our odometry results during a snowy sequence. Here, we have plotted the results of training on the Oxford dataset and testing on this sequence, as well as training on the Boreas dataset and testing on this sequence. There is a noticeable increase in the drift rate when transferring from the Oxford dataset to this Boreas dataset. We hypothesize that this difference is mainly due to the different range resolutions of the sensors.
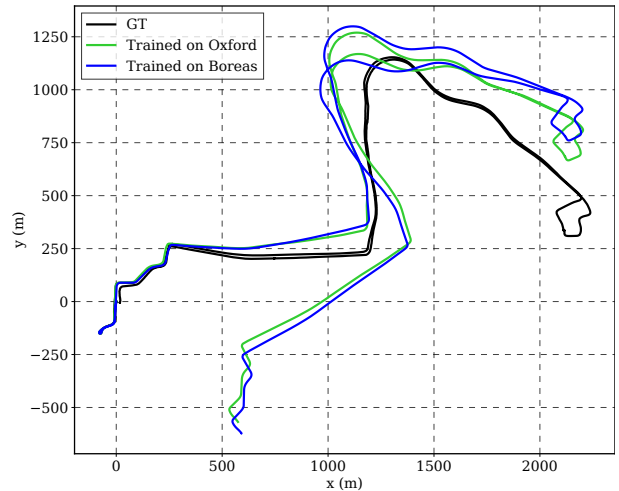
REFERENCES

[1] Safety is everything. - Navtech Radar. https://navtechradar.com/. Accessed: 2020-11-13.

[2] Daniel Adolfsson, Martin Magnusson, Anas Alhashimi, Achim J Lilienthal, and Henrik Andreasson. CFEAR Radar-odometry – Conservative Filtering for Efficient and Accurate Radar Odometry. *arXiv preprint arXiv:2105.01457*, 2021.

[3] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. Fast radar motion estimation with a learnt focus of attention using weak supervision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1190–1196. IEEE, 2019.

[4] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. What could go wrong? introspective radar odometry in challenging environments. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2835–2842. IEEE, 2019.

[5] Sean Anderson and Timothy D Barfoot. Full STEAM Ahead: Exactly Sparse Gaussian Process Regression for Batch Continuous-Time Trajectory Estimation on SE(3). In *IROS*, 2015.

[6] Sean Anderson, Timothy D Barfoot, Chi Hay Tong, and Simo Särkkä. Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression. *Autonomous Robots*, 39(3):221–238, 2015.

[7] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[8] Timothy D Barfoot, James R Forbes, and David J Yoon. Exactly Sparse Gaussian Variational Inference with Application to Derivative-Free Batch Nonlinear State Estimation. *IJRR*, 2020.

[9] Dan Barnes and Ingmar Posner. Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9484–9490, 2020.

[10] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438, 2020.

[11] Dan Barnes, Rob Weston, and Ingmar Posner. Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information. In *Conference on Robot Learning*, pages 303–316, 2020.

[12] Michael Broome, Matthew Gadd, Daniele De Martini, and Paul Newman. On the Road: Route Proposal from Radar Self-Supervised by Fuzzy LiDAR Traversability. *AI*, 1(4):558–585, 2020.

[13] Keenan Burnett, Jingxing Qian, Xintong Du, Linqiao Liu, David J. Yoon, Tianchang Shen, Susan Sun, Sepehr Samavi, Michael J. Sorocky, Mollie Bianchi, Kaicheng Zhang, Arkady Arkhangorodsky, Quinlan Sykora, Shichen Lu, Yizhou Huang, Angela P. Schoellig, and Timothy D. Barfoot. Zeus: A system description of the two-time winner of the collegiate SAE autodrive competition. *Journal of Field Robotics*, 38(1):139–166, 2021.

[14] Keenan Burnett, Angela P Schoellig, and Timothy D Barfoot. Do We Need to Compensate for Motion Distortion and Doppler Effects in Spinning Radar Navigation? *IEEE Robotics and Automation Letters*, 6(2):771–778, 2021.

[15] Sarah H Cen and Paul Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[16] Sarah H Cen and Paul Newman. Radar-only ego-motion estimation in difficult settings via graph matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 298–304. IEEE, 2019.

[17] N. Charron, S. Phillips, and S. L. Waslander. De-noising of Lidar Point Clouds Corrupted by Snowfall. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 254–261, 2018.

[18] Paul Checchin, Franck Gérossier, Christophe Blanc, Roland Chapuis, and Laurent Trassoudaine. Radar scan matching slam using the fourier-mellin transform. In *Field and Service Robotics*, pages 151–161. Springer, 2010.

[19] Steve Clark and Hugh Durrant-Whyte. Autonomous land vehicle navigation using millimeter wave radar. In *1998 IEEE Conference on Robotics and Automation (ICRA)*, volume 4, pages 3697–3702. IEEE, 1998.

[20] Daniele De Martini, Matthew Gadd, and Paul Newman. kRadar++: Coarse-to-Fine FMCW Scanning Radar Localisation. *Sensors*, 20(21):6002, 2020.

[21] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-improving visual odometry. *arXiv preprint arXiv:1812.03245*, 2018.

[22] Matthew Gadd, Daniele De Martini, and Paul Newman. Look Around You: Sequence-based Radar Place Recognition with Learned Rotational Invariance. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 270–276, 2020.

[23] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[24] Zoubin Ghahramani and Geoffrey E Hinton. Parameter Estimation for Linear Dynamical Systems. Technical Report CRG-TR-96-2, U. of Toronto, 1996.

[25] Zoubin Ghahramani and Sam T Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, 1999.

[26] Ziyang Hong, Yvan Petillot, and Sen Wang. RadarSLAM: Radar based Large-Scale SLAM in All Weathers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[27] Ziyang Hong, Yvan Petillot, Andrew Wallace, and Sen Wang. Radar SLAM: A Robust SLAM System for All Weather Conditions. *arXiv preprint arXiv:2104.05347*, 2021.

[28] Matthew James Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Composing graphical models with neural networks for structured representations and fast inference. *Advances in Neural Information Processing Systems*, 2016.

[29] Ebi Jose and Martin David Adams. Relative RADAR cross section based feature identification with millimeter wave RADAR for outdoor SLAM. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 425–430. IEEE, 2004.

[30] Prannay Kaul, Daniele De Martini, Matthew Gadd, and Paul Newman. RSS-Net: Weakly-Supervised Multi-Class Semantic Segmentation with FMCW Radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[31] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[32] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2014.

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.

[34] Pou-Chun Kung, Chieh-Chih Wang, and Wen-Chieh Lin. A Normal Distribution Transform-Based Radar Odometry Designed For Scanning and Automotive Radars. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[35] Katherine Liu, Kyel Ok, William Vega-Brown, and Nicholas Roy. Deep inference for covariance estimation: Learning Gaussian noise models for state estimation. In *ICRA*, 2018.

[36] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 2017.

[37] Michael J Milford and Gordon F Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE, 2012.

[38] Radford M Neal and Geoffrey E Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer, 1998.

[39] Yeong Sang Park, Young-Sik Shin, and Ayoung Kim. PhaRaO: Direct Radar Odometry using Phase Correlation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2617–2623, 2020.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

[41] Hermann Rohling. Ordered statistic CFAR technique-an overview. In *2011 12th International Radar Symposium (IRS)*, pages 631–638. IEEE, 2011.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Comp.-Assisted Intervention*, 2015.

[43] 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.

[44] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: A Radar Dataset for Automotive Perception. *arXiv preprint arXiv:2010.09076*, 2020.

[45] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[46] Tim Y Tang, Daniele De Martini, Dan Barnes, and Paul Newman. RSL-Net: Localising in Satellite Images From a Radar on the Ground. *IEEE Robotics and Automation Letters*, 5(2): 1087–1094, 2020.

[47] Tim Y Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-Supervised Localisation between Range Sensors and Overhead Imagery. In *2020 Robotics: Science and Systems*, 2020.

[48] Rob Weston, Sarah Cen, Paul Newman, and Ingmar Posner. Probably unknown: Deep inverse sensor modelling radar. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5446–5452. IEEE, 2019.

[49] David Williams, Daniele De Martini, Matthew Gadd, Letizia Marchegiani, and Paul Newman. Keep off the Grass: Permissible Driving Routes from Radar with Weak Audio Supervision. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[50] Jeremy N Wong, David J Yoon, Angela P Schoellig, and Timothy D Barfoot. Variational Inference with Parameter Learning Applied to Vehicle Trajectory Estimation. *RAL*, 2020.

[51] David J Yoon, Haowei Zhang, Mona Gridseth, Hugues Thomas, and Timothy D Barfoot. Unsupervised Learning of Lidar Features for Use in a Probabilistic Trajectory Estimator. *RAL*, 2021.

APPENDIX

TABLE IV

Evaluation on 7 sequences from the Oxford Dataset. Performance is reported as translational drift (%) / rotational drift (deg/1000m) using the common KITTI odometry metric. This table is restricted to algorithms that have been tested on similar sequences. Cen RO, Kung RO, and MC-RANSAC are hand-crafted algorithms. Masking by Moving, Under the Radar, and HERO are all learning-based algorithms. HERO is currently the only unsupervised learning-based radar odometry approach. *Tested on all 32 sequences. **Uses dense correlation between scans.

Method	Evaluation	Sequences							Mean
		10-14-02	11-12-26	11-14-02	14-12-05	15-13-06	16-11-53	17-11-46	
Cen RO [15]	[9]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3.72/9.5
Kung RO* [34]	[34]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.96/6.0
Masking by Moving** [11]	[11]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.16/3.0
Under the Radar [9]	[9]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	2.05/6.7
MC-RANSAC [14]	Ours	3.43/11.2	3.72/12.9	3.27/10.9	3.41/10.7	3.07/9.8	3.12/9.9	3.21/11.0	3.31/10.9
HERO (Ours)	Ours	1.86/5.9	1.85/6.2	1.93/6.5	1.96/6.8	1.80/6.1	2.43/7.1	2.08/7.1	1.99/6.5

(a) 11-12-26   (b) 11-14-02   (c) 14-12-05

(d) 15-13-06   (e) 16-11-53   (f) 17-11-46
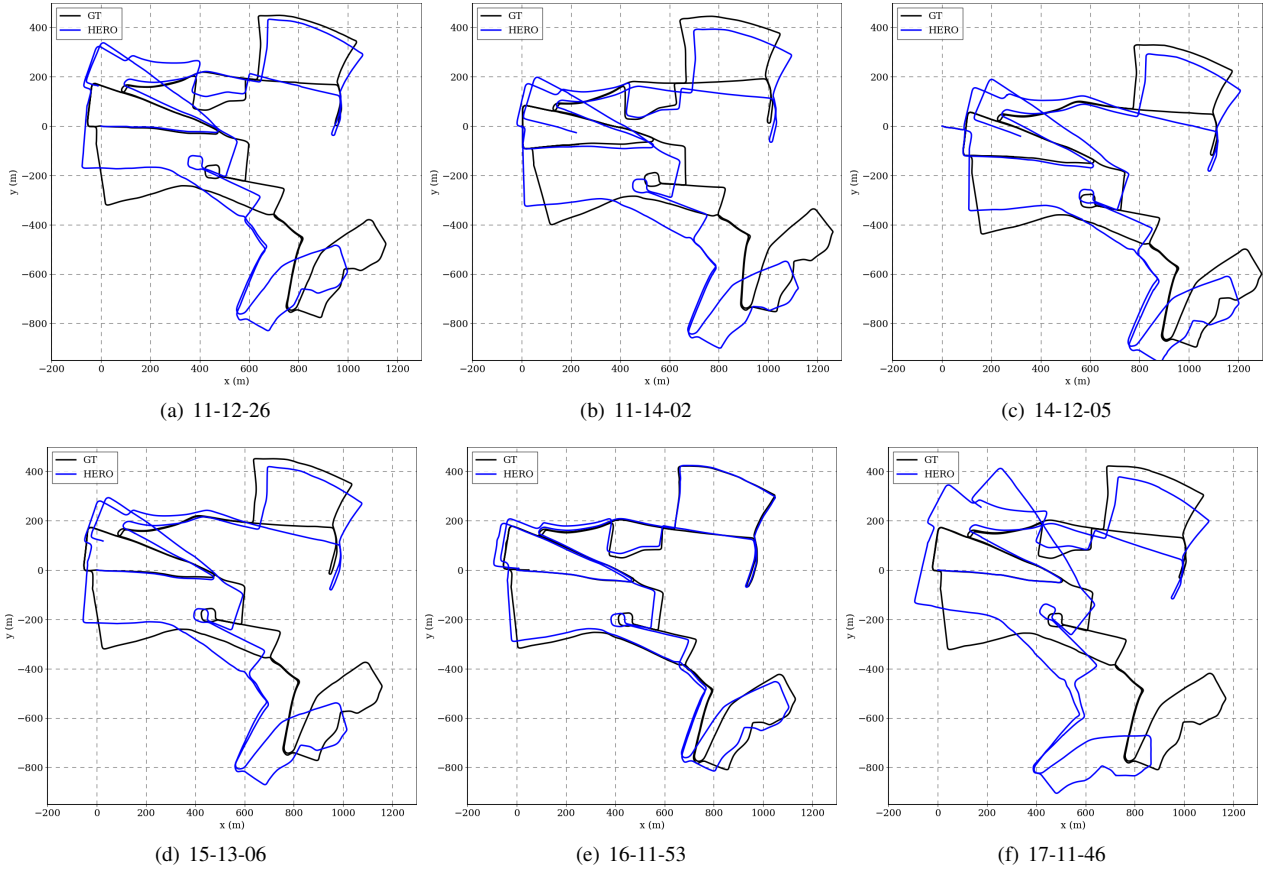
Fig. 8.   These results illustrate the performance of HERO during each of the test sequences. See Figure 5 for HERO's performance on 10-14-02.

TABLE V

An evaluation on 8 sequences from the Oxford Dataset. This experiment required a new train/validation/test split. 16-13-42 was used for validation. Performance is reported as translational drift (%) / rotational drift (deg/100m) using the common KITTI odometry metric. *Uses loop closures.

Method	Evaluation	Sequences								Mean
		10-11-46	10-12-32	16-11-53	16-13-09	17-13-26	18-14-14	18-14-46	18-15-20	
Hong RO [27]	[27]	2.16/0.6	2.32/0.7	2.49/0.7	2.62/0.7	2.27/0.6	2.29/0.7	2.12/0.6	2.25/0.7	2.32/0.7
Hong SLAM* [27]	[27]	1.96/0.7	1.98/0.6	1.81/0.6	1.48/0.5	1.71/0.5	2.22/0.7	1.68/0.5	1.77/0.6	1.83/0.6
CFEAR [2]	[2]	1.65/0.48	1.64/0.48	1.99/0.53	1.86/0.52	1.66/0.48	1.71/0.49	1.79/0.5	1.75/0.51	1.76/0.50
HERO (Ours)	Ours	2.14/0.71	1.77/0.62	2.01/0.61	1.75/0.59	2.04/0.73	1.83/0.61	1.97/0.65	2.20/0.77	1.96/0.66