Deep Probabilistic Methods For Improved Radar Sensor Modelling and Pose Estimation

Rob Weston

Keble College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2022

Abstract

Radar's ability to sense under adverse conditions and at far-range makes it a valuable alternative to vision and lidar for mobile robotic applications. However, its complex, scene-dependent sensing process and significant noise artefacts makes working with radar challenging. Moving past classical rule-based approaches, which have dominated the literature to date, this thesis investigates deep and data-driven solutions across a range of tasks in robotics.

Firstly, a deep approach is developed for mapping raw sensor measurements to a grid-map of occupancy probabilities, outperforming classical filtering approaches by a significant margin. A distribution over the occupancy state is captured, additionally allowing uncertainty in predictions to be identified and managed. The approach is trained entirely using partial labels generated automatically from lidar, without requiring manual labelling.

Next, a deep model is proposed for generating *stochastic* radar measurements from simulated elevation maps. The model is trained by learning the forward and backward processes side-by-side, using a combination of adversarial and cyclical consistency constraints in combination with a partial alignment loss, using labels generated in lidar. By faithfully replicating the radar sensing process, new models can be trained for down-stream tasks, using labels that are readily available in simulation. In this case, segmentation models trained on simulated radar measurements, when deployed in the real world, are shown to approach the performance of a model trained entirely on real-world measurements.

Finally, the potential of deep approaches applied to the radar odometry task are explored. A learnt feature space is combined with a classical correlative scan matching procedure and optimised for pose prediction, allowing the proposed method to outperform the previous state-of-the-art by a significant margin. Through a probabilistic consideration the uncertainty in the pose is also successfully characterised. Building upon this success, properties of the Fourier Transform are then utilised to separate the search for translation and angle. It is shown that this decoupled search results in a significant boost to run-time performance, allowing the approach to run in real-time on CPUs and embedded devices, whilst remaining competitive with other radar odometry methods proposed in the literature.

Deep Probabilistic Methods For Improved Radar Sensor Modelling and Pose Estimation



Rob Weston

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2022

Acknowledgements

For his passion and guidance I would like to thank my supervisor, Professor Ingmar Posner. Likewise, my co-authors – Sarah, Dan, Oiwi, Matt, Daniele and Paul – for their help along the way and to the A2I group for many lively discussions. A special thanks to Matt for his extensive proofreading and a wider thanks to everyone at the ORI for making it the wonderful place it is today.

To Andrew, Helen, and Beth for the making of lifelong friends. I would also like to specially thank my parents and brothers for the support they have shown me over the years.

Finally, to Zoe, my financier, lover, and biggest fan.[1]

[1] And for berating my inconsistent use of the oxford comma

Abstract

Radar's ability to sense under adverse conditions and at far-range makes it a valuable alternative to vision and lidar for mobile robotic applications. However, its complex, scene-dependent sensing process and significant noise artefacts makes working with radar challenging. Moving past classical rule-based approaches, which have dominated the literature to date, this thesis investigates deep and data-driven solutions across a range of tasks in robotics.

Firstly, a deep approach is developed for mapping raw sensor measurements to a grid-map of occupancy probabilities, outperforming classical filtering approaches by a significant margin. A distribution over the occupancy state is captured, additionally allowing uncertainty in predictions to be identified and managed. The approach is trained entirely using partial labels generated automatically from lidar, without requiring manual labelling.

Next, a deep model is proposed for generating *stochastic* radar measurements from simulated elevation maps. The model is trained by learning the forward and backward processes side-by-side, using a combination of adversarial and cyclical consistency constraints in combination with a partial alignment loss, using labels generated in lidar. By faithfully replicating the radar sensing process, new models can be trained for down-stream tasks, using labels that are readily available in simulation. In this case, segmentation models trained on simulated radar measurements, when deployed in the real world, are shown to approach the performance of a model trained entirely on real-world measurements.

Finally, the potential of deep approaches applied to the radar odometry task are explored. A learnt feature space is combined with a classical correlative scan matching procedure and optimised for pose prediction, allowing the proposed method to outperform the previous state-of-the-art by a significant margin. Through a probabilistic consideration the uncertainty in the pose is also successfully characterised. Building upon this success, properties of the Fourier Transform are then utilised to separate the search for translation and angle. It is shown that this decoupled search results in a significant boost to run-time performance, allowing the approach to run in real-time on CPUs and embedded devices, whilst remaining competitive with other radar odometry methods proposed in the literature.

Contents

List of Figures ix

Acronyms xi

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Contributions . . . 7
- 1.3 Publications . . . 9
- 1.4 Outline . . . 10

2 Background 11
- 2.1 Radar Fundamentals . . . 11
  - 2.1.1 Transmission To Reception . . . 12
  - 2.1.2 The Sensing Process . . . 14
    - 2.1.2.1 FMCW Radar . . . 15
  - 2.1.3 Sources of Uncertainty . . . 16
  - 2.1.4 Challenges . . . 20
- 2.2 Technical Background . . . 24
  - 2.2.1 Assumed Density Models . . . 25
  - 2.2.2 Heteroscedastic Models . . . 27
    - 2.2.2.1 The Semi-Supervised Case . . . 28
  - 2.2.3 Deep Implicit Models . . . 30
    - 2.2.3.1 Generative Adversarial Networks . . . 31
    - 2.2.3.2 CycleGAN . . . 35
  - 2.2.4 Architecture Design . . . 36
- 2.3 Conclusion . . . 40

3 Probably Unknown: Deep Inverse Sensor Modelling In Radar 41

4 There And Back Again: Learning to Simulate Radar Data for Real World Applications 51

5 Masking By Moving: Learning Distraction-Free Radar Odometry from Pose Information **63**
5.1 Probabilistic Correlative Scan 79

6 Fast-MbyM: Leveraging Translational Invariance of the Fourier Transform for Efficient And Accurate Radar Odometry **83**

7 Discussion **93**
7.1 Contributions 93
7.2 Future Research 96
7.3 Concluding Remarks 98

8 Appendix **99**
8.1 Probability Theory 99
8.1.1 Random Variables 99
8.1.2 The Rules of Probability 100
8.1.3 Expectations 101
8.1.3.1 The Law Of Large Numbers 101
8.1.3.2 The Reparameterisation Trick 101
8.1.4 The Kullback Leibler Divergence 102
8.2 Inference 104
8.2.1 Frequentist Inference 104
8.2.1.1 Maximum Likelihood 104
8.2.1.2 Empirical Risk 105
8.2.2 Bayesian Inference 106
8.2.2.1 Variational Bayes 107
8.2.2.2 Sampling Approaches 108
8.3 Optimisation 110
8.4 Derivations 115
8.4.1 Loss Functions For Assumed Density Models 115

References **117**

List of Figures

1.1 **Vision, lidar and radar under snowy conditions** 2
1.2 **An example of the complex reasoning required to interpret Radio Detection And Ranging (radar) data** 4

2.1 **A typical FMCW radar sensing pipeline** 14
2.2 **Sources of uncertainty in radar** 17
2.3 **Amplifier saturation and the Spectral components of a sinusoidal signal** 20

4.1 **Network Training Setup** 61

Acronyms

BCE Binary Cross Entropy. 34

CE Cross Entropy. 26

CFAR Constant False Alarm Rate. 21

CNN Convolutional Neural Network. 36

CPU Central Processing Unit. 8, 95

EM Electro-Magnetic. 11

FMCW Frequency-Modulated Continuous-Wave. 12, 14, 15, 17, 20

GAN Generative Adversarial Network. 24, 31, 34, 97

GPU Graphical Processing Unit. 8

JSD Jensen-Shannon Divergence. 33

KLD Kullback-Leibler Divergence. 28

lidar Light Detection And Ranging. 1–5, 7

MAE Mean Absolute Error. 26

MSE Mean Squared Error. 26, 30

radar Radio Detection And Ranging. 3–11

RCS Radar Cross Section. 13, 16, 18, 19, 30

TIS Training In Simulation. 7, 96

ToF Time of Flight. 14

VCO Voltage Controlled Oscillator. 15

WW2 Second World War. 11

1 Introduction

1.1 Motivation

In recent years Light Detection And Ranging (lidar) and cameras have become the sensors of choice across a wide range of robotic and autonomous vehicle applications. Cameras, similar to the human eye, are particularly suited to detecting visual cues used – and in many instances created – by humans. Their widespread adoption and availability makes them an affordable deployment solution. Lidar, on the other hand, is an active sensor, allowing the world to be observed even when natural illumination fails. In contrast to vision, it provides direct measurements of world geometry, measuring the line-of-sight distance to objects with high accuracy. Partially as a result of the often complementary properties of lidar and vision, a vast amount of effort has been dedicated to developing lidar and vision systems in order to solve a range of problems across robotics [1–4]. To this end, large datasets and problem-specific challenges have been released by academic and industrial stakeholders alike – to test, develop and promote lidar and vision solutions [5–7].

Concurrently, the adoption of deep methods applied to both the lidar and vision modalities has reaped dividends. Moving away from classical rule-based methods has allowed deep models to surpass the performance of classical methods in a wide range of problem scenarios [1–4]. This is achieved by learning feature representations

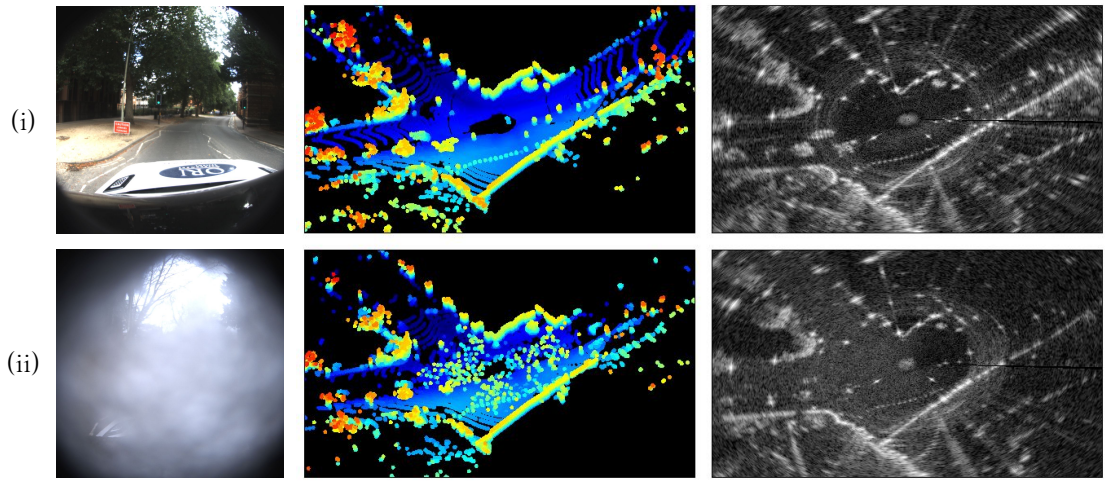


Figure 1.1 : Vision, lidar and radar under snowy conditions Vision (a), lidar (b) and radar (c) observed under normal (i) and snowy (ii) conditions. In (b-ii) lidar struggles to operate under snowy conditions, whilst in (a-ii) a snowflake has fallen on the camera lens obscuring the view entirely. The scene observed in radar remains relatively unchanged.

best suited to the task at hand from data, instead of relying on subjective and often difficult to optimise hand-crafted approaches.

Despite this, significant challenges still persist when deploying lidar and vision in the real world. Cameras require significant post-processing steps to answer relatively simple questions about world geometry, such as determining the distance to an object. When explicit three-dimensional geometry is needed, highly accurate motion estimates (structure from motion) or calibration (multiple camera setup) are required [8]. Relying on a passive sensor, vision solutions are also prone to fail when natural illumination is limited, such as at night, indoors, or when working with shadows.

Lidar comes with its own challenges. Naturally increasing with distance from the sensor, sparsity – due to the number of lasers it is physically and economically feasible to fit in a single device – in combination with sensor noise, limit the range of conventional lidars. This is a particular hindrance to the deployment of lidars in situations with large scene dynamics (such as autonomous driving) where the detection of distant but fast moving objects is crucial for safe operation. Lidars are also often thwarted by adverse environmental conditions, such as heavy

rain, snow and dust. An example of the limitations of lidar operating in snowy conditions is shown in Fig. 1.1

In contrast to vision and lidar, radar offers several significant advantages. Similar to lidar and in contrast to vision, it too is an active sensor, readily deployed when natural illumination is limited. It also provides direct measurements of world geometry, where the distance to objects is *known* explicitly. Unlike lidar, however, it is able to detect objects at greater range and under a variety of adverse conditions, including rain, snow, and dust (see Fig. 1.1 for an example). Radar holds significant promise, not only as an alternative sensor to vision and lidar but also as a complementary member of the sensing suite. This is particularly important for guaranteeing sensor redundancy, which has a key role to play in ensuring robust operation in safety critical scenarios, such as autonomous driving.

Open Challenges Despite this, in comparison to vision and lidar, radar has received relatively little attention from the research community. As discussed in more detail in Sec. 2.1.3, radar sensor observations are characterised by significant *aleatoric* noise artifacts. This makes interpreting radar sensor measurements challenging, hindering its widespread adoption. Even to a radar expert, interpreting radar measurements often relies on complex reasoning, combining scene specific context with past knowledge and intuition. As a motivating example, see Fig. 1.2.

Designing classical methods to overcome radar's complex and scene dependent acquisition process is challenging. As a result, converting and filtering the raw power measurements to a simpler-to-work-with representation is a prerequisite step in typical classical approaches [9–12]. Automotive radars, for example, produce a sparse object list as their output, involving the filtering and clustering of raw measurements in combination with tracking approaches [12]. Such representations, whilst easier to work with, result in a significant loss of information, which may be useful for the down-stream task.

In contrast to classical approaches, where manually defining the complex rules required to interpret radar measurements is challenging, deep methods have the

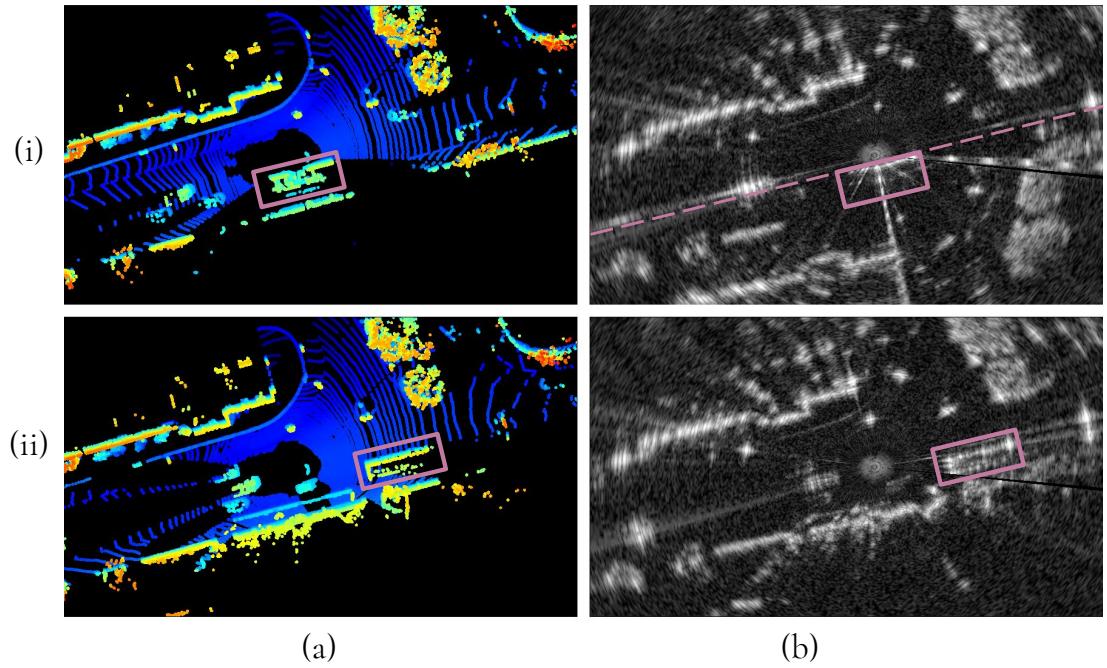


Figure 1.2 : **An example of the complex reasoning required to interpret radar data** For comparison and context (a) shows a lidar measurement of the scene. In (b-i) a bus (▭) occludes the radar's view and results in a complex ghost reflection, imprinted on the lower side of the line of reflection (- - -). Once the bus has moved on the actual radar image is revealed (b-ii). Reasoning which radar power measurements correspond to actual objects in the scene is challenging in this case, relying on scene context (the location of the bus) and prior knowledge of the sensing process (multi-path reflections).

capacity to learn such rules from raw data alone. Recent innovations in network architectures and training paradigms demonstrate the feasibility of using deep data-driven approaches to learn ever more complex mappings across a range of tasks in robotics in both vision and lidar [1–4]. Concurrently, the development of graphical computing hardware allows these mappings to be executed quickly – a crucial requirement for real-world operation.

Therefore, it seems like deep data-driven approaches are a natural fit for harnessing the full potential of raw radar sensor measurements. However, their application to radar has some unique challenges. Firstly, in contrast to vision and lidar, the availability of large scale and high quality radar datasets remains limited. As well as being expensive and tedious to generate, labelling radar datasets also requires significant specialist knowledge; the difficulty in interpreting radar scans leads to a greater chance of mislabelling and increases subjectivity. Self-

supervised solutions are therefore particularly attractive in the case of radar. Here, instead, labels can be generated automatically using information from other sensing modalities, often in combination with accurate offline processing algorithms (e.g. mapping and localisation systems).

Secondly, as already discussed, a prominent challenge when working with radar is the high uncertainty prevalent in its sensor measurements. Deep data-driven models may go some way in overcoming this challenge, providing a framework that allows for complex reasoning about the world in light of scene context and implicit understanding of the radar sensing process. However, even in the case of a perfect model, there will be times when inherent uncertainty in the predictions remains as a result of the aleatoric uncertainty arising through the radar sensing process. This is an important aspect of radar data and an observation that should not be ignored when working with radar measurements. As a solution, probabilistic approaches seek to quantify the uncertainty alongside the prediction, capturing a distribution of possible outcomes. This allows the highly uncertain cases to be identified and mitigated.

Conclusion In summary, radar holds significant promise as a viable alternative and complementary sensor to vision and lidar. It is a geometric sensor able to detect objects at far range, and under a variety of adverse conditions. However, significant noise artifacts and a complex image formation process result in radar sensor measurements which are challenging to work with. Interpreting radar scans involves complex reasoning, combining scene specific context with prior knowledge of the radar sensing process. This is difficult to distill into classical rule-based approaches, which typically resort to filtering the raw radar power returns in order to generate simpler-to-work-with but less informative representations. Instead, drawing on their successes in the vision and lidar domain, across a range of tasks in robotics, deep data-driven approaches offer a promising alternative. They allow complex reasoning steps to be learnt from raw datasets, without requiring difficult-to-define hand-crafted algorithms.

However, whilst an attractive solution and a natural fit when considering radar sensor measurements, applying deep data-driven approaches to the radar domain comes with some unique challenges. Firstly, there is a significant lack of large, diverse and high quality radar datasets available to the community. Secondly, even in the case of a perfect model, uncertainty in the predictions of the model still remains as a result of the intrinsic uncertainty in the sensor measurements. This problem is particularly pertinent to radar in comparison to other modalities, given the high uncertainty attached to its sensor measurements. In countering the first of these problems, adopting self-supervised approaches allows deep methods to be trained by generating training labels automatically. With regard to the second, by adopting probabilistic models the uncertainty in predictions can also be quantified, capturing a range of likely outcomes, and allowing highly uncertain predictions to be identified and mitigated.

This thesis therefore investigates deep data-driven approaches applied to radar. Specifically, solutions will be developed across the tasks of inverse sensor modelling, simulation, and odometry. In addition, where possible, a particular focus is given to developing self-supervised and probabilistic approaches, allowing training labels to be generated automatically, and the uncertainty in predictions to also be quantified.

1.2 Contributions

The specific contributions made by this thesis are as follows.

In chapter 2 the fundamentals of the radar sensing process and the challenges it presents, alongside key topics from deep learning and probabilistic modelling, are summarised to provide a self-contained resource of relevant background material.

In chapter 3, a deep data-driven approach to inverse sensor modelling is developed. Specifically, the problem of converting radar sensor measurements to a grid map of occupancy probabilities is considered. Through our approach we side-step the need for manual labels, relying only on partial labels generated automatically from lidar to train our model. By using a deep neural network, the model is able to successfully reason about the occupancy state considering a wider scene context, successfully outperforming classical filtering approaches. All the while, the approach remains probabilistic, quantifying the uncertainty in the occupancy state.

In chapter 4, the problem of replicating the radar sensing process in simulation is considered. Contrary to typical approaches, we model the radar sensing process with a deep model, allowing the complex interaction between world context and the sensing process to be learnt from raw-data. An inherently stochastic approach is developed allowing radar noise artifacts to be faithfully reproduced and a distribution over possible sensor measurements to be implicitly captured. Once again the developed approach remains self-supervised; our model is trained using a combination of simulated datasets and real-world labels generated automatically from lidar. Using our learnt simulator, we are able to train segmentation models in simulation, before deploying them in the real world. To the best of our knowledge this is the first time that the Training In Simulation (TIS) paradigm – already exploited widely by the vision and lidar communities – has been investigated for radar. As an added benefit, the backward model – inferring the elevation state of the world – is also learnt, and after training can be used to cast radar observations into a 2.5D representation with reasonable accuracy.

In the final chapters deep and data-driven approaches to radar odometry are developed. In chapter 5 a correlative scan matching radar odometry method is proposed, utilising a deep neural network to filter raw sensor measurements to boost performance. The entire process remains differentiable, allowing the radar feature representation output by the network to be learnt for pose prediction, without requiring hand-crafting or classical signal processing. As a joint project, the particular contribution of this thesis lies in the inherently probabilistic formulation of the approach, allowing the uncertainty in the pose estimate to be quantified and calibrated to real-world errors.

Whilst the approach developed in chapter 5 is able to run efficiently on a Graphical Processing Unit (GPU), the dense search across all possible combinations of angle and translation hinders real-time performance when high-end compute is not available. In chapter 6, therefore, the algorithm developed in chapter 5 is adapted, splitting the search for angle and translation into two stages, using properties of the Fourier Transform. This significantly increases the efficiency of the approach, allowing the model to run in real-time on both Central Processing Units (CPUs) and embedded devices, as well as requiring less time and memory resources to train.

1.3 Publications

The following is a list of publications which are included in this thesis:

- **R. Weston**, S. Cen, P. Newman, I. Posner. "Probably Unknown: Deep Inverse Sensor Modelling in radar". In: *International Conference On Robotics and Automation 2019 (ICRA Montreal 2019)*
- D. Barnes, **R. Weston**, I. Posner. "Masking by Moving: Learning Distraction-Free radar Odometry From Pose Information". In: *Conference On Robotic Learning 2019 (CoRL Osaka 2019)*
- **R. Weston**, O. Parker Jones, I. Posner. "There and Back Again: Learning to Simulate radar Data for Real-World Application". In: *International Conference On Robotics and Automation 2021 (ICRA Xi'an 2021)*
- **R. Weston**, M. Gadd, D. De Martini, P. Newman, I. Posner. "Fast-MbyM: Leveraging Translational Invariance of the Fourier Transform For Efficient And Accurate radar Odometry". In: *International Conference On Robotics and Automation 2022 (ICRA Philadelphia 2022)*

1.4 Outline

In chapter 2 the necessary background material needed for a full understanding of the subsequent chapters in this thesis is introduced. Note an introduction to the relevant material specific to each of the chapters can be found in the introduction and literature review of each of the publications. As a result, chapter 2 specifically focuses on the key and over-arching background material needed to fully understand this thesis as a whole. The fundamentals of the radar sensing process will be discussed and several pertinent challenges of working with radar highlighted. An overview of several key topics from deep learning is then given.

Chapters 3 to 6 are dedicated to the technical contributions of this thesis, presenting the publications listed in Sec. 1.3 in manuscript format. Finally, in chapter 7 limitations of the current work is discussed, and future research directions are identified.

2 Background

2.1 Radar Fundamentals

The foundations of radar (Radio Detection And Ranging) can be traced back as far as 1886 when Heinrich Hertz showed that radio waves are reflected by solid objects [13]. In 1904 Christian Hülsmeyer filed perhaps the first patent for a device using radio waves as a means of detecting physical objects. His device, the *tele-mobiloscope*, was specifically designed to detect ships through dense fog [14]. Just before and during the Second World War (WW2) agencies around the world raced to develop military technology using radio waves, such as early warning systems capable of detecting approaching aircraft.

Unlike other Electro-Magnetic (EM) wavelengths, such as visible, infrared and ultraviolet wavelengths, radio waves are particularly suited for these applications as they are only weakly absorbed by weather phenomena such as fog, rain, and snow. Radar also interacts strongly with conductive materials such as metals. Since then, radar has been exploited in a wide range of applications such as astronomy, air traffic control, meteorology and more recently robotics and autonomous vehicle applications.

Whilst, radars have been developed operating over a wide range of wavelengths (from 100 m all the way down to 1 mm), the size of object that it is possible to

(easily) detect is typically limited to a few multiples of the wavelength. In robotics applications short-wave radars operating with mm wavelenths are deployed. This allows objects down to a few cm in size to be detected at the expense of higher absorption rates, limiting the range of the radar to a few km. Smaller wavelengths also mean smaller antennas which are more practical to deploy on mobile robots. In line with this, the radar used in this thesis is a Frequency-Modulated Continuous-Wave (FMCW) radar operating at 76 GHz to 77 GHz corresponding to a ∼4 mm wavelength and with a range resolution of ∼4 cm.

The long range and the ability of radar to sense in a variety of adverse conditions where other sensors typically fail make it an attractive solution for robotic tasks. However, a complex image formation process and significant noise artifacts result in radar sensor measurements that are challenging to work with. In preparation for a more in depth discussion of the challenges presented by radar, the transmission and the FMCW sensing process are now discussed in more detail.

2.1.1 Transmission To Reception

The transmission process takes place as follows [15]:

1. A *transmitter* emits a radio wave. As the wave travels its power P_0 will be distributed over a sphere of radius $\rho_0 = c\tau_0$ (where c is the speed of light $[\mathrm{m\,s^{-1}}]$ and τ_0 is the travel time [s]). The power falling on an object at distance ρ_0 is therefore given as

$$S = \frac{G_0}{4\pi\rho_0^2} P_0 \qquad [\mathrm{W\,m^{-2}}] \tag{2.1}$$

   where the antenna gain $G_0 > 1$ is used to account for the fact that the antenna is designed to be directional. In the case of the radar used in this thesis, this allows the power returns from precise locations in the world to be determined, executing a full "scan" of the scene by rotating the transmitter.

2. On encountering an object a change in material occurs and the wave is scattered. The amount of power reflected by the object is determined by the

Radar Cross Section (RCS) σ [m^2] which is a function of the object shape and material properties. The object therefore radiates a power

$$P_\sigma = \sigma S \qquad [\text{W}] \tag{2.2}$$

back to the receiver.

3. After travelling a distance $\rho_1 = c\tau_1$ the amount of power at the receiver is given as

$$P_1 = \frac{A_1}{4\pi\rho_1^2} P_\sigma \qquad [\text{W}] \tag{2.3}$$

where A_1 [m^2] is the area of the aperture of the receiver which governs how much power is intercepted. It can be shown [15] that the antenna aperture is related to the gain G_1 as $A_1 = G_1\lambda^2/4\pi$ where λ [m] is the wavelength of the wave respectively.

The Radar Equation Combining Eqs. (2.1) to (2.3), grouping like terms and substituting $A_1 = G_1\lambda^2/4\pi$ gives

$$P_1 = k\frac{G_0 G_1 \lambda^2 \sigma}{(4\pi)^3 \rho_0^2 \rho_1^2} P_0 \tag{2.4}$$

a relationship commonly referred to as the *radar equation*. The additional constant $k \in [0, 1]$ is used to account for path loss effects dependent on the scanned environment. Note that, for the sake of clarity the discussion so far has considered the general case where the transmitter and receiver are separate from one another. However, in many instances, (as is the case for the radar used in this thesis) the antenna is shared between the receiver and transmitter and so $G_0 G_1 = G^2$ and $\rho_0^2\rho_1^2 = \rho^4$.

Time Of Flight The total time between transmission and reception is given as

$$\tau = \tau_0 + \tau_1 = \frac{\rho_0}{c} + \frac{\rho_1}{c} \tag{2.5}$$

where ρ_0 is the distance from the transmitter to the object and ρ_1 is the distance from the object back to the receiver. Considering a shared transmitter and receiver gives

$$\tau = \frac{2\rho}{c} \,. \tag{2.6}$$

where $\rho_0 + \rho_1 = 2\rho$.

2.1.2 The Sensing Process

Whilst, the transmission process is common to all types of radar, radars typically differ in how they *measure* the received power. When considering robotic applications the Time of Flight (ToF) and FMCW radars are most common. In a ToF radar set-up a short pulse of energy is emitted and after a time delay of τ is returned to the receiver. The distance to an object is determined as $\rho = c\tau/2$ in accordance with Eq. (2.6). However, to ensure a sufficiently high range resolution, a large amount of energy must be emitted over a short time period requiring a high peak power which may be challenging to implement in practice (see [16] Pg. 52 for more details).

In contrast, FMCW radars allow a finer resolution to be achieved for a reduced peak power, a particular advantage when considering robotic applications such as autonomous driving where detecting smaller objects is important and power is limited.

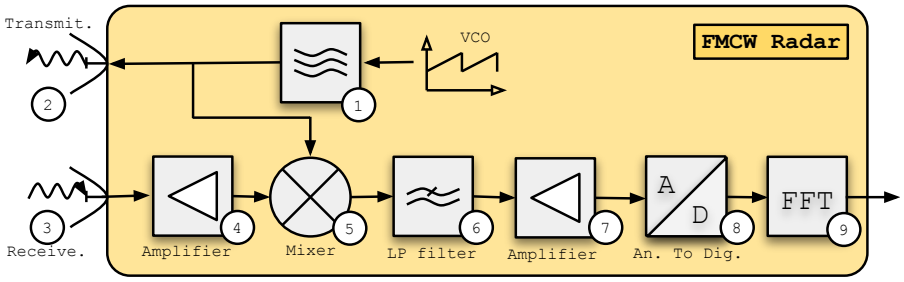


Figure 2.1 : **A typical FMCW radar sensing pipeline** A Voltage Controlled Oscillator (VCO) is used to generate the signal $v_0(t)$ in step ①. This signal is then propagated into the world using a transmitting antenna and converted back to an electronic signal $v_1(t)$ using the receiving antenna, in steps ② and ③. To extract the beat frequency component, the signal is amplified and mixed to give $m(t) = v_0(t)v_1(t)$ before a low pass filter is applied to extract only the low frequency component $v_-(t)$ (steps ④, ⑤ and ⑥). After more amplification, the power spectrum of $v_-(t)$ is calculated in the Fourier domain using an analogue to digital converter and the Fast Fourier Transform (steps ⑦, ⑧ and ⑨). Adapted from [17].

2.1.2.1 FMCW Radar

FMCW radars transmit a *frequency modulated* signal

$$v_0(t) \triangleq A_0 \cos\psi_0(t) \quad \text{where} \quad \psi_0(t) \triangleq \int_0^t f(t')dt' \tag{2.7}$$

to determine the power received from a target as shown in Fig. 2.1. In perhaps the most common set up the instantaneous frequency is linearly increased from a *carrier* frequency ω_c to a maximum frequency of $\omega_c + \alpha T$ over a time period T using a Voltage Controlled Oscillator (VCO).[1] In this case the instantaneous frequency at a time t is given as $f(t) \triangleq \omega_c + \alpha t$. For this choice of $f(t)$:

- The *transmitted* waveform becomes

$$v_0(t) \triangleq A_0 \cos\psi_0(t) = A_0 \cos(\omega_c t + \frac{1}{2}\alpha t^2) \tag{2.8}$$

  (after substituting $f(t)$ into Eq. (2.7)).

- The *received* signal (Fig. 2.1 at ③) given as $v_1(t) \propto v_0(t-\tau)$ is scaled (see Eq. (2.4)) and offset (see Eq. (2.6)) to give:

$$v_1(t) \triangleq A_1 \cos\psi_1(t) = A_1 \cos(\omega_c (t-\tau) + \frac{1}{2}\alpha (t-\tau)^2) \tag{2.9}$$

- The *mixed* signal $m(t) \triangleq v_0(t)v_1(t)$ (Fig. 2.1 at ⑤) can be expressed as

$$m(t) = A\cos\psi_-(t) + A\cos\psi_+(t) \tag{2.10}$$
$$\psi_-(t) = \alpha\tau t + \phi \tag{2.11}$$
$$\psi_+(t) = 2\omega_c t - \alpha\tau t + \alpha t^2 - \phi \tag{2.12}$$

  using $\psi_1(t) = \psi_0(t-\tau)$ where $A \triangleq A_1A_0/2$ and $\phi \triangleq \omega_c\tau - \frac{1}{2}\alpha\tau^2$. (This result derives from the trigonometric identity $2c_ac_b = c_{a+b} + c_{a-b}$ with $c_a \triangleq \cos(a)$ and defining $\psi_\pm(t) = \psi_0 \pm \psi_1$.)

[1] other waveforms can also be used

Assuming that the carrier frequency ω_c and ramp gradient α are chosen such that $\omega_c \gg \alpha t$ $(\forall t \in [0, T])$ the signal $m(t)$ in Eq. (2.10) is composed of a *low* frequency component $v_- \triangleq A\cos\psi_-(t)$ and a *high* frequency component $v_+(t) \triangleq A\cos\psi_+(t)$. Therefore, passing $m(t)$ through a low pass filter (Fig. 2.1 at ⑥), leaves only the low-frequency component

$$v_-(t) \triangleq A\cos\psi_-(t) = A\cos(\alpha\tau t + \phi) \triangleq A\cos(\omega t + \phi) \tag{2.13}$$

where the *beat frequency* $\omega \triangleq \alpha\tau$ is directly proportional to the range to the target $\omega \triangleq \alpha\tau = \alpha(2\rho/c)$.

The power $P_1(\rho, \theta)$ received from a target at range ρ and observation angle θ can therefore be measured as $P_1(\rho, \theta) \propto S_{v_-}(\omega) = S_{v_-}(2\alpha\rho/c)$ where S_{v_-} is the *power spectral density*[2] of the signal $v_-(t)$. Repeating, this procedure at ranges $\rho_1 \ldots \rho_R$ and observation angles $\theta_1 \ldots \theta_\Theta$ (by electronically steering the sensor through a sequence of azimuths) allows a dense set of power measurements $\mathbf{P} \in \mathbb{R}^{R\times\Theta}$ to be determined.

2.1.3 Sources of Uncertainty

Armed with knowledge of radar's transmission and measurement processes, several important sources of uncertainty arising through the radar sensing process may now be discussed. Some of these are common to all radars (e.g. interference, speckle noise, RCS variability, and multi-path phenomena) and are a natural consequence of how radar transmits and receives energy from the world. Others derive from hardware limitations in the sensing pipeline (e.g. phase noise, saturation and non-ideal beam artifacts). Several of these sources of uncertainty are exemplified in Fig. 2.2.

Interference Like any EM device, sources of radiation not originating from the sensor have the potential to interfere with the perceived power measurements and may arise from a combination of both natural (eg. the sun) and man-made sources (e.g. switching artifacts). External radiation sources may operate over the entire operational range or may correspond to only very specific wavelengths. When

[2] $S_{v_-} = \lim_{T\to\infty} \frac{1}{T}\hat{v}_-(\omega)\hat{v}_-^*(\omega)$ where $\hat{v}_-(\omega)$ denotes the Fourier transform of $v_-(t)$ and $*$ denotes complex conjugation.

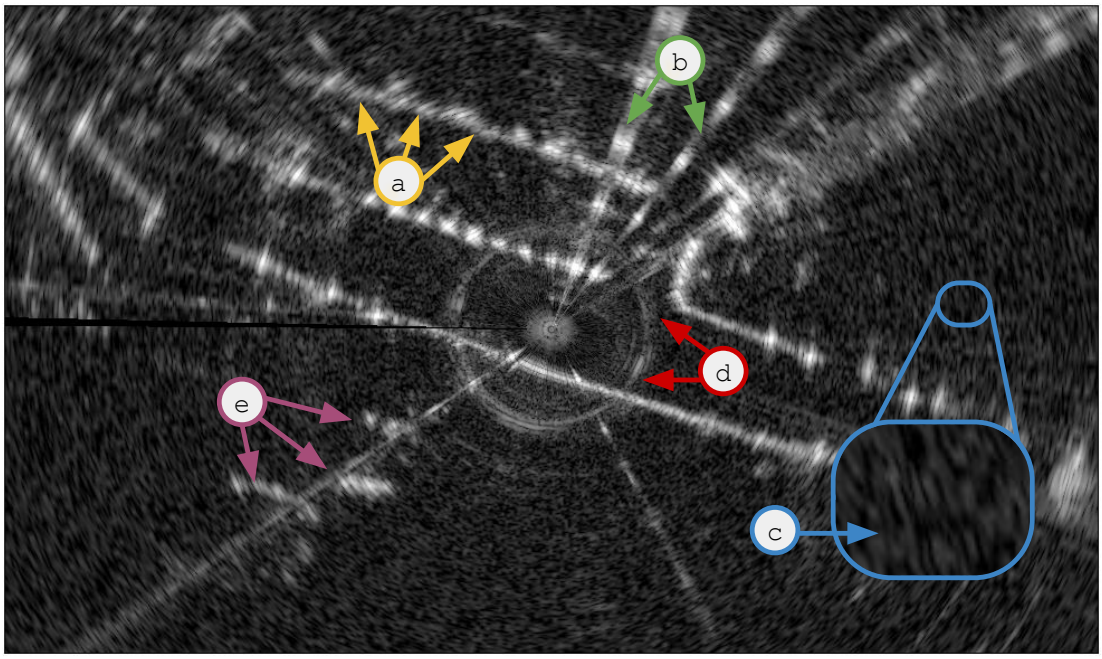


Figure 2.2 : **Sources of uncertainty in radar** At ⓐ a ghost reflection of a wall can be seen arising through multi-path reflections in the opposite wall. At ⓑ a strong response from an object has caused the amplifier to saturate leading to bright radial streaks in the radar power field. Constructive and destructive interference as a result of integrating the radar power measurements over a finite range results in speckle noise as can be seen in ⓒ. A halo artifact is observed at ⓓ. The power returns at ⓔ correspond to actual objects in the scene and are observed as the absolute height of the spreading beam grows with range. The height of such objects is difficult to infer depending on the size of objects observed downstream.

considering FMCW the latter manifests as a high perceived return at a fixed range irrespective of beam angle, and results in halo artifacts such as can be seen in Fig. 2.2 at ⓓ. A closely related problem to interference is *clutter*. In this case the returns originate from radiation emitted by the radar but correspond to unhelpful returns from distractor objects in the scene.

Speckle The finite resolution of the radar in range, $\Delta\rho$, means that a power measurement $\bar{P}_r(\rho_i)$ at range ρ_i is generated from a combination of the responses $P_1(\rho)$ for $|\rho - \rho_i| < \Delta\rho$. Over this range the change in beat frequency is negligible and so the received waveforms are coherent when they reach the receiver. In this case the responses $P_1(r)$ will sum to give $\bar{P}_r(\rho_i)e^{j\phi_i} = \int_{\rho_i-\Delta\rho}^{\rho_i+\Delta\rho} P_1(\rho)e^{j\phi(\rho)}d\rho$, interacting either constructively or destructively depending on the phase variation,

$\phi(\rho)$. Indeed, in practice $\phi(\rho)$ can change drastically over $|\rho - \rho_i| < \Delta\rho$ and is highly sensitive to properties, such as the antenna directivity [18].[3] This results in random fluctuations in power returns – from dark black to bright white – referred to as *speckle noise*. In the limit of an infinite number of waveforms interfering in this manner the observed power $\bar{P}_r(\rho_i)$ will follow an exponential distribution $p(\bar{P}_r|\lambda) = \lambda e^{-\lambda \bar{P}_r}$ where $1/\lambda$ is the average power return.

Radar Cross-Section Variability The RCS σ of a target (discussed in Eq. (2.4)) is a complex function of wave incidence angle, material properties, and geometry. As the radar or target move relative to one another the RCS can vary significantly. Targets are also rarely composed of a single scattering element and, similar to the discussion of speckle noise, as a result of the coherent nature of radar over small length scales, multiple scatterers may interact either constructively or destructively depending on their highly varying phase. A varying RCS results in a varying power response, even when considering the same target in accordance with Eq. (2.4).

Multi-Path Reflections In practice, radiation from the transmitter may take one of a number of paths when travelling back to the receiver, reflecting off multiple targets along the way. The same target may therefore be perceived at multiple ranges resulting in *ghost objects* – false images of the target. This phenomenon is particularly prevalent when considering surfaces which are smooth relative to the radar's wavelength. Planar surfaces act as mirrors to visible light in this instance exemplified in Fig. 2.2 at ⓐ.

Beam Width Unlike the sparse measurements of lidar, which senses the world using a finite number of lasers with narrow beams, radar observes the world using a wider lobe with power distributed over a range of azimuth and elevation. This allows radar to capture a denser measurement, observing more of the world at

[3] An insight in this effect can be gained by considering adding together N unit vectors all with different angles. The magnitude of the vector in this case will be random, ranging from 0 where the contributions from each vector cancel, right through to N when the vectors are aligned and super-impose on top of one another.

any time instance, but comes at the expense of greater ambiguity in the sensor measurements, as power measurements from a range of elevations are projected into a single power reading.

In addition to the main beam emitted by the transmitting antenna side lobes are also present. Power detected at a particular range and angle may also contain power from other angles covered by side lobes.

Phase Noise The analysis in Sec. 2.1.2.1 was conducted assuming it is feasible to generate a perfect ramp signal $f(t) = \omega_c + \alpha t$. In reality, this process is implemented using a VCO (step ① in Fig. 2.1). In this case the ramp signal is typically given as $f(t) = \omega_c + \alpha t + \eta(t)$ where the *phase noise* $\eta(t)$ is a consequence of the finite bandwidth of the VCO [16, 19]. The appearance of phase noise manifests as a blurring of the power measurement in the radial direction as responses from targets at different ranges are mixed together.

Saturation Considering Eq. (2.4) the power received from a target can vary greatly varying as $P_1 \propto 1/\rho^4$ with range and $P_1 \propto \sigma$ with RCS, which itself can also vary over several magnitudes, depending on the material and geometry of the target. As the returned power is such that $P_1 \propto 1/\rho^4$, it is relatively small and must be amplified (step ④ in Fig. 2.1) to be converted to a usable signal. Practically, implementing amplifiers over such a large magnitude range is challenging, and very large power returns cause the amplifier to saturate. Amplifier saturation causes a clipping of the signal and results in a splitting of its frequency into multiple harmonics. The effect of varying degrees of amplifier saturation on the frequency components of the signal is shown in Fig. 2.3 and leads to the bright azimuthal streaks seen at ⓑ in Fig. 2.2.

Doppler Shift When considering a target moving away from a radar with velocity v the beat frequency is actually given as $\omega = \alpha\tau + \omega_d$ where $\omega_d = 2v/c\omega_c$ is the Doppler shift. As a result, when the Doppler shift is not accounted for, an error $e_\rho \triangleq \tilde{\rho} - \rho$ between the measured range $\tilde{\rho} \triangleq \omega c/2\alpha$ and the true range

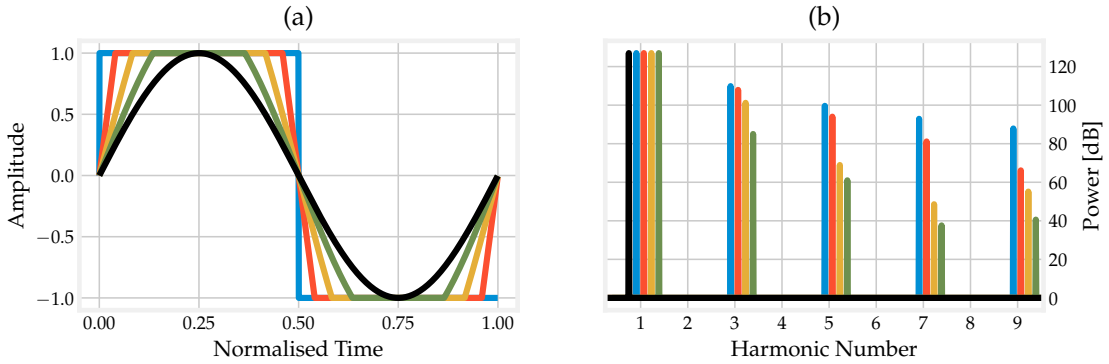


Figure 2.3 : **Amplifier saturation and the Spectral components of a sinusoidal signal** Large power signals at the receiver result in waveforms with a large amplitude which fall beyond the operating range of the amplifier ($[-1, 1]$ in (a)). This causes a clipping of the waveform at all magnitudes beyond the operating range, as can be seen in (a). Clipping causes the power in the signal to be split over multiple harmonics as can be seen in (b). When considering FMCW radar each harmonic will be perceived as a different target. Superimposing such effects, as the instantaneous frequency ramps over a cycle, results in the bright azimuthal streaks seen in Fig. 2.2 at (b)

ρ is introduced, given as $e_\rho = v/\omega_c\alpha$. It is therefore impossible to distinguish between power from a stationary target at range $\rho + e_\rho$ and power from a target moving with velocity v at range ρ.

2.1.4 Challenges

Despite the significant benefits that working with radar measurements brings to robotic deployments, it is evident that significant challenges exist, which must be overcome to realise the full potential of radar. The superposition of each of the noise artifacts and sources of uncertainty described in the previous section makes interpreting and working with raw radar sensor measurements challenging.

In many instances, classical approaches attempt to overcome these challenges by first filtering the raw power measurements into a simpler but easier to work with representation [11]. This is a common procedure in automotive radars, for example, which instead of returning raw power measurements typically provide the user with point representations of objects in the scene, generated through a combination of filtering and clustering approaches [12]. Perhaps the simplest approach to filtering raw measurements is to apply a static threshold to the radar power returns. However,

the highly scene-dependent and heterogeneous nature of the radar sensing process makes finding a suitable threshold challenging, either resulting in a high number of false or otherwise missed detections. Constant False Alarm Rate (CFAR) [20] approaches attempt to overcome this limitation by setting the power threshold dynamically using statistics of the radar power field to achieve a desired probability of false alarm. These statistics are typically estimated empirically based on *local* observations of the power field from neighboring grid cells. Several CFAR variants have been developed, depending on the assumed distribution of power returns in the presence of or absence of a target [21–23]. Other variants propose alternative methods for determining local statistics other than the average (which is known to be highly sensitive to outliers), such as Ordered statistic (OS-CFAR) [24] and Greatest Of (GO-CFAR) [25] approaches.

In practice, these classical filtering approaches have several shortfalls. The underlying distribution of radar power returns is typically unknown and complex due to the noise sources just described. When, the assumed distribution of power returns is incorrect, there are no guarantees that the constant rate of false alarm may be achieved by CFAR approaches in practice. Methods relying on local information struggle to correctly identify targets in more challenging cases. For example, ghost objects appearing in the radar sensor measurement as a result of multi-path reflection, appear locally identical, whilst they do not correspond to a valid target. In this and other examples, the validity of detections may only be made by accounting for the wider context of the power returns in the scene. Finally, designing perfect filtering algorithms is challenging, and any suboptimal algorithm will naturally involve a loss of potentially useful information. When pre-processing radar sensor measurements, this information is irretrievably lost even before the deployment task is considered.

In addition to presenting significant challenges to the deployment of radar on real-world robots, its complex scan formation process is also challenging to replicate in simulation. In turn, this poses additional challenges for real-world deployment, making the safety case difficult to prove, particularly when considering scenarios

which are too expensive or dangerous to test in the real world. As discussed in more detail in chapter 4, hand-crafted phenomenological models typically fail to capture radar's complex image formation process in practice. On the other hand, ray-tracing approaches are computationally expensive and rely on having specialized models of the world, alongside an accurate characterisation of radar hardware – such as antenna geometry and the downstream sensing process [26].

Conclusion

In the face of the challenges introduced in this section, deep models offer a valuable alternative to the classical approaches just described. This thesis develops methods that operate on raw power measurements – directly mapping from sensor measurement to system outputs in an end-to-end fashion. In this way, models are able to reason about radar power measurements in the presence of a wider scene context. The capacity of deep neural networks to capture highly non-linear mappings offers a valuable solution for overcoming radar's complex and scene-dependent image formation process, alongside its stochastic and heterogeneous noise artifacts. In this instance, the feature representation is explicitly optimised for the task at hand using any relevant information in the *raw* measurement, which might otherwise have been lost through a pre-filtering step.

In addition to providing motivation for the deep modelling approaches developed in later chapters, the notable sources of uncertainty in the radar sensing process provide significant motivation for a probabilistic approach. Even when considering a perfect model, uncertainty in predictions is likely to still persist. This is in part a result of impartial knowledge such as the difficult to measure and highly variable radar cross-section of a target, or unknown target geometry which, in combination with a beam covering a range of heights and widths, makes determining occlusion challenging. In other cases, this uncertainty derives from an inherent ambiguity between high power returns and the presence of actual targets in the scene, resulting from a combination of radar's significant capacity for multipath reflections, amplifier saturation, phase noise, and speckle artifacts. Characterising

and managing the uncertainty in radar sensor measurements through probabilistic approaches is therefore important.

2.2 Technical Background

The relevant background material for chapters 3 to 6 is now presented. Letting $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ denote a dataset of observations generated from a real process

$$(\mathbf{x}, \mathbf{y}) \sim p_\star(\mathbf{y}|\mathbf{x})p_\star(\mathbf{x}) \tag{2.14}$$

in this section we consider how we might learn a model $p_\phi(\mathbf{y}|\mathbf{x})$, using a neural network with parameters ϕ (which approximates the true process $p_\star(\mathbf{y}|\mathbf{x})$ as closely as possible), a problem which emerges throughout the remainder of this thesis. Several approaches are of particular importance for what is is to follow and are now detailed.

The discussion begins with *assumed density models*, where a particular distributional form for the model $p_\theta(\mathbf{y}|\mathbf{x})$ is fixed (eg. Gaussian, Bernoulli etc.) *a priori*. As described in Sec. 2.2.1, adopting a a maximum likelihood approach, several important loss functions naturally emerge in this case, depending on the choice of distribution $p_\theta(\mathbf{y}|\mathbf{x})$. This section provides the foundation for the approaches developed in chapters 5 and 6 and sets the scene for the discussion in Sec. 2.2.2 and Sec. 2.2.3.

Building upon the homoscedastic case introduced in Sec. 2.2.1, in which the uncertainty in each prediction is assumed fixed, next the heteroscedastic case is considered. The uncertainty in the prediction is now captured as a *function* of the input. This is detailed in Sec. 2.2.2 in preparation for the approach developed in chapter 3.

Finally, in preperation for chapter 4, *implicit modelling approaches* are discussed. Here, instead of explicitly assuming a particular distributional form for the model, a distribution is *implicitly* captured; samples $\mathbf{x} \sim p_\theta(\mathbf{y}|\mathbf{x})$ are generated from a simpler base distribution $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ using a neural network $\mathbf{y} \triangleq f_\theta(\mathbf{x}, \mathbf{z})$. In this case it is no longer possible to evaluate the density $p_\theta(\mathbf{y}|\mathbf{x})$. Whilst deep implicit models offer a powerful approach for capturing real world processes (such as the radar sensing process in chapter 4), a likelihood-free training approach must now be adopted. To this end, the the framework of Generative Adversarial Networks (GANs) is presented, a likelihood free training approach which will be utilised

Title	Chapter	Relevance
Assumed Density Models	Sec. 2.2.1	chapters 5 and 6
Heteroscedastic Models	Sec. 2.2.2	chapter 3
Deep Implicit Models	Sec. 2.2.3	chapter 4
Architecture Design	Sec. 2.2.4	chapters 3 to 6

Table 2.1 : A Summary of the technical background material and its relevance to later chapters

extensively in chapter 4. Particular, attention is given to the cycle GAN paradigm which allows models to be learnt from pairs of examples which are unaligned in space and time, by modelling both the forward and backward mappings and imposing cyclical consistency between the two.

Having detailed specific background material in Secs. 2.2.1 to 2.2.3 to support the presentation of the technical chapters chapters 3 to 6 the final part of this chapter is dedicated to a more general discussion of network architectures applied to the radar datatype and relevant to all of the approaches developed in this thesis. This can be found in Sec. 2.2.4. A summary of the background material and its relevance to the approaches developed later on is given in Tab. 2.1.

2.2.1 Assumed Density Models

Assumed density models are constructed by choosing a distributional form for $p_\phi(\mathbf{y}|\mathbf{x})$ [27]. For example a Gaussian density might be chosen $p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Nor}(\mathbf{y}|f_\phi(\mathbf{x}), \sigma^2\mathbf{I})$ using a neural network f_ϕ to capture the mean of the distribution (where ϕ is used to denote the parameters of the network). By construction, assumed density models have well-defined likelihoods and are easily trained, exploiting the simplicity and theoretical guarantees of maximum likelihood approaches

$$\hat{\phi} \triangleq \arg\min_{\phi} -\log p(\mathcal{D}|\phi) = \arg\min_{\phi} \frac{1}{N}\sum_{n=1}^{N} \ell(\phi; \mathbf{x}_n, \mathbf{y}_n) \tag{2.15}$$

where $\ell(\phi) \triangleq -\log p_\phi(\mathbf{y}|\mathbf{x})$. As shown in Sec. 8.2.1.1 maximum likelihood estimation guarantees that if $p_\phi(\mathbf{y}|\mathbf{x})$ has the capacity to perfectly capture the true process $p_\star(\mathbf{y}|\mathbf{x})$ using parameter setting $\phi_\star$ (such that $p_{\phi_\star}(\mathbf{y}|\mathbf{x}) = p_\star(\mathbf{y}|\mathbf{x})$), then in the limit as $N \to \infty$ then $\hat{\phi} \to \phi_\star$.

Different likelihoods $p_\phi(\mathbf{y}|\mathbf{x})$ give rise to different *loss functions* $\ell(\phi) \triangleq -\log p_\phi(\mathbf{y}|\mathbf{x})$ and many common loss functions typically used to train deep models can be derived in this way. For example considering

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Nor}(\mathbf{y}|f_\phi(\mathbf{x}), \sigma^2\mathbf{I}) \implies \ell(\phi; \mathbf{y}, \mathbf{x}) = \|\mathbf{y} - \mu_{\phi_\mu}(\mathbf{x})\|_2^2 \tag{2.16}$$

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Lap}(\mathbf{y}|f_\phi(\mathbf{x}), \sigma^2) \implies \ell(\phi; \mathbf{y}, \mathbf{x}) = \|\mathbf{y} - \mu_{\phi_\mu}(\mathbf{x})\|_1 \tag{2.17}$$

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Cat}(\mathbf{y}|\pi_\phi(\mathbf{x})) \implies \ell(\phi; \mathbf{y}, \mathbf{x}) = \sum_i \mathbf{y}_i \log \pi_\phi(\mathbf{x})_i \tag{2.18}$$

and substituting $\ell(\phi; \mathbf{y}, \mathbf{x})$ for each case, the Mean Squared Error (MSE), Mean Absolute Error (MAE) and Cross Entropy (CE) training criteria naturally emerge (see Sec. 8.4.1 for proofs). Here, $f_\phi : \mathbb{R}^{n_{in}} \to \mathbb{R}^{n_{out}}$ is a neural network with parameters ϕ and $\pi_\phi(\mathbf{x}) \triangleq \texttt{softmax}(f_\phi(\mathbf{x}))$ where $\texttt{softmax}(\boldsymbol{a})_i = e^{\boldsymbol{a}_i} / \sum_k e^{\boldsymbol{a}_k}$ is the softmax activation function.

Whilst any distribution $p_\phi(\mathbf{y}|\mathbf{x})$ may be chosen, different choices of $p_\phi(\mathbf{y}|\mathbf{x})$ may lead to networks f_ϕ that capture different properties of the true data generating process $p_\star(\mathbf{y}|\mathbf{x})$. For example, when considering a Gaussian likelihood it can be shown that the optimum function $f_\phi(\mathbf{x})$ found by minimising [27] corresponds to the mean of $p_\star(\mathbf{y}|\mathbf{x})$ such that $f_{\hat{\phi}}(\mathbf{x}) = \mathbb{E}_{p_\star(\mathbf{y}|\mathbf{x})}\{\mathbf{y}\}$. In contrast, when considering a Laplace likelihood the optimum function $f_\phi(\mathbf{x})$ found by minimising Eq. (2.15) corresponds to the median of $p_\star(\mathbf{y}|\mathbf{x})$.[4] If we know that $p_\star(\mathbf{y}|\mathbf{x})$ is approximately Gaussian then the mean and median should be aligned; both MAE and MSE should lead to similar results. If this is not the case then MAE may be a better choice, as the median is less sensitive to outliers than the mean. This observation in part motivates the shift from MSE to a MAE training criterion between chapter 5 and chapter 6 when training radar odometry systems. In this case, whilst the dataset consists of generally small poses, on rare occasions much larger poses are also observed, resulting in significant error outliers.

[4] Note that multiple definitions exist for the median in the multivariate case. The marginal median is equivalent to taking the scalar median along each dimension.

2.2.2 Heteroscedastic Models

Whilst, so far the discussion has focused on *homoscedastic* models in which the uncertainty in the prediction is fixed *a priori*, it is also possible to define models where the uncertainty in the prediction is allowed to vary as function of the input. Such models are referred to as *heteroscedastic* and are utilised extensively in chapter 3.

An example of a heteroscedastic model for regression is given as

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \mathtt{Nor}(\mathbf{y}|f_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})^2\mathbf{I}) \tag{2.19}$$

where the uncertainty $\sigma_\phi(\mathbf{x})$ is now allowed to vary as a function of the input $\mathbf{x}$. (Here $\sigma_\phi : \mathbb{R}^{n_{in}} \to [0, \infty)^{n_{out}}$ is modelled using a neural network and the output constraint is typically enforced using a softplus activation function $h_L(a) = \log(1 + \exp(a))$.)

Heteroscedastic models for classification Of particular importance to chapter 3 are heteroscedastic models for *classification*, which are now discussed for the slightly more general case $\mathbf{y} \in \{0, 1\}^K$ (treated here as a 1 of K coded vector). One approach to modelling the heteroscedastic uncertainty [28] in this case is to consider a model of the form

$$p_\phi(\mathbf{z}|\mathbf{x}) \triangleq \mathtt{Nor}(\mathbf{z}|f_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})^2\mathbf{I}) \qquad \text{model} \tag{2.20}$$

$$\pi_\phi(\mathbf{z}) \triangleq \mathtt{softmax}(\mathbf{z}) \tag{2.21}$$

$$p(\mathbf{y}|\mathbf{z}) \triangleq \mathtt{Cat}(\mathbf{y}|\pi_\phi(\mathbf{z})) \qquad \text{likelihood} \tag{2.22}$$

$$p_\phi(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{z}) p_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} \qquad \text{model evidence} \tag{2.23}$$

treating $\mathbf{z} \in \mathbb{R}^K$ (input to the softmax) as a Gaussian distributed random variable. The model is trained by maximising the model evidence

$$\ell(\phi) \triangleq -\log p_\phi(\mathbf{y}|\mathbf{x}) \tag{2.24}$$

marginalising out the uncertainty attached to the latent variable $\mathbf{z}$ (see Eq. (2.23)). However, no closed form solution to the integral in Eq. (2.23) is known to exist. As an alternative, the model evidence is approximated as

$$p_\phi(\mathbf{y}|\mathbf{x}) \approx \mathtt{Cat}(\mathbf{y}|\bar{\boldsymbol{\pi}}) \quad \text{with} \quad \bar{\boldsymbol{\pi}} = \mathbb{E}_{p_\phi(\mathbf{z}|\mathbf{x})}[\pi_\phi(\mathbf{z})] \approx \frac{1}{K}\sum_k \pi_\phi(\mathbf{z}_k) \tag{2.25}$$

where $\bar{\boldsymbol{\pi}}$ is formed using Monte-Carlo integration (with $\mathbf{z}_k \sim p_\phi(\mathbf{z}|\mathbf{x})$). Substituting $p_\phi(\mathbf{y}|\mathbf{x}) \approx \texttt{Cat}(\mathbf{y}|\bar{\boldsymbol{\pi}})$, Eq. (2.24) is now easily estimated as

$$\ell(\phi) \approx -\sum_c \mathbf{y}_c \log \bar{\boldsymbol{\pi}}_c = -\sum_c \mathbf{y}_c \left(\frac{1}{K} \exp(\mathbf{z}_{c,k} - \log \sum_{c'} \exp \mathbf{z}_{c',k}) \right) \tag{2.26}$$

using the log-sum-exp trick for numerical stability and the reparameterisation trick (see Sec. 8.1.3.2) to generate samples $\mathbf{z}_k \triangleq f_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x})^2 \circ \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \texttt{Nor}(\mathbf{0}, \mathbf{I})$ to ensure $\ell(\phi)$ remains differentiable in the parameters ϕ.

2.2.2.1 The Semi-Supervised Case

In chapter 3, a closely related approach is developed for the binary classification task $y \in \{0, 1\}$ where the likelihood becomes

$$p(y|z) = \texttt{Ber}(y|z) \quad \text{with} \quad \pi(z) \triangleq \texttt{sigmoid}(z) \tag{2.27}$$

where $z \in \mathbb{R}$. In this instance, however, to facilitate a self-supervised formulation, labels are only *sometimes* available (see chapter 3 for more details).

In response, a modified training objective is proposed

$$\ell(\phi) \triangleq -\gamma \mathbb{E}_{p_\phi(z|\mathbf{x})}\{\log p(y|z)\} + \omega(1-\gamma)\mathbb{KL}[p_\phi(z|y)\|p(z)] \tag{2.28}$$

where $\mathbb{KL}$ denotes the Kullback-Leibler Divergence (KLD) (see Sec. 8.1.4) and $\gamma \in \{0, 1\}$ is used to denote whether a label is available. Here, as a stand-in[5] for Eq. (2.23), the first term now corresponds to the (binary) cross-entropy loss

$$\mathbb{E}_{p_\phi(z|\mathbf{x})}\{\log p(y|z)\} \approx \frac{1}{K} \sum_k y \log\{\pi(z_k)\} + (1-y) \log\{1 - \pi(z_k)\} \tag{2.29}$$

averaged over different logit values $z_k = f_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x})\epsilon_k$ with $\epsilon_k \sim \texttt{Nor}(0, 1)$ (using the reparameterisation trick).

On the other hand, when labels are *unavailable* ($\gamma = 0$) the KL term forces $p_\phi(z|x)$ to resort back to a prior $p(z) \triangleq \texttt{Nor}(z|0, \sigma_0^2)$ which for the case of two Gaussian models can be evaluated analytically [29]. The constant $\omega \geq 0$ is used to trade off the contribution between the two terms as $\ell(\phi)$ is summed over multiple examples.

[5] Applying Jensen's inequality we have $\log p_\phi(y|\mathbf{x}) = \log \mathbb{E}_{p_\phi(z|\mathbf{x})}\{p(y|z)\} \geq \mathbb{E}_{p_\phi(z|\mathbf{x})}\{\log p(y|z)\}$

Note, whilst this approach is designed to facilitate semi-supervised training, the form of the loss given in Eq. (2.28) can alternatively be derived as a variational lower bound (see Sec. 8.2.2.1)

$$\ell(\phi) \triangleq -\mathbb{E}_{p_\phi(z|y)} \left\{ \log p(y, z) - \log p_\phi(z|y) \right\} \tag{2.30}$$

induced through a joint distribution $p(y, z) \triangleq p(y|z)p(z)$ and treating $p_\phi(z|y)$ as a variational posterior. Using the identity presented in Eq. (8.36) we find that Eq. (2.30) is exactly equivalent to Eq. (2.28). For the unfamiliar reader, further discussion about the variational lower bound and variational inference approaches can be found in Sec. 8.2.2.1.

2.2.3 Deep Implicit Models

The final modelling paradigm discussed in this chapter is presented in preparation for the approach developed in chapter 4, in which a model is sort with the aim of replicating the radar measurement process. Before this it is worth briefly considering the limitations of assumed density models when applied to this problem.

Limitations of Assumed Density Models Consider an assumed density model of the form

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Nor}(\mathbf{y}|y_\phi(\mathbf{x}), \sigma^2\mathbf{I}) \tag{2.31}$$

tasked with generating radar observations $\mathbf{y} \in \mathbb{R}^{R\times\Theta}$ given simulated world states $\mathbf{x}$ (for example an elevation map as in chapter 4). As just discussed, the maximum likelihood objective is equivalent to MSE in this case, and, the optimum parameter setting $\phi^\star$ will result in $y_{\phi^\star}(\mathbf{x})$ capturing the mean of the true data generating process $y_{\phi^\star}(\mathbf{x}) = \mathbb{E}_{p_\star(\mathbf{y}|\mathbf{x})}[\mathbf{y}]$.

Assuming that the optimum parameters $\phi^\star$ can be found, new radar scans are sampled from the model as

$$p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Nor}(\mathbf{y}|y_\phi(\mathbf{x}), \sigma^2\mathbf{I}) \quad \Leftrightarrow \quad \mathbf{y} = \bar{\mathbf{y}} + \epsilon \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \tag{2.32}$$

where $\bar{\mathbf{y}} = y_\phi(\mathbf{x})$. This corresponds to perturbing individual power values about the mean. Even in the ideal case, where it is possible to determine $\phi^\star$ exactly it is clear $\mathbf{y} \sim p_{\phi^\star}(\mathbf{y}|\mathbf{x})$ is unlikely to simulate realistic radar scans, unless real radar scans $\mathbf{y} \sim p_\star(\mathbf{y}|\mathbf{x})$ are also generated as in Eq. (2.32). In practice, many of the noise artifacts (described in Sec. 2.1.3) are poorly captured by considering independent perturbations to pixel intensities. RCS variability, for example, results in varying power returns from a target across multiple power measurements simultaneously. Variability in saturation and halo artifacts, appearing as bright streaks in either range or azimuth, are also poorly captured in this case. Whilst the model is able to learn how to regress to average radar measurements, the distributional form of the model – fixed a priori – in this case fails to align well with the real world.

Deep Implicit Models Deep implicit models offer a powerful alternative. Instead of assuming a density $p_\phi(\mathbf{y}|\mathbf{x})$, a density is captured *implicitly*, allowing samples to be generated as

$$\mathbf{y} \sim p_\phi(\mathbf{y}|\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{y} = g_\phi(\mathbf{x}, \boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \texttt{Nor}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I}) \tag{2.33}$$

using a deep neural network g_ϕ. A potentially vast set of distributions may be captured using this approach, which vastly increases the chances of capturing a density which aligns well with the real world.

This comes at the expense however of maximum likelihood objectives; as the density $p_\phi(\mathbf{y}|\mathbf{x})$ is only implicit, it can no-longer be evaluated, making maximum likelihood approaches infeasible. Instead, implicit modelling approaches rely on optimising ϕ using *likelihood-free* approaches.

2.2.3.1 Generative Adversarial Networks

A successful and powerful approach to likelihood-free inference is afforded by the GAN framework [30, 31] which will be used extensively in chapter 4. The original GAN formulation is now discussed in more detail, in preparation. The Least Squares GAN formulation is then introduced, which is used in chapter 5 in an attempt to overcome some of the challenges presented by the original formulation.

The Original GAN Formulation In the original GAN formulation proposed in [30] the model $p_\phi(\mathbf{y}|\mathbf{x})$ is found as the solution of a two-player min-max game

$$\phi_\star, \beta_\star = \max_\beta \min_\phi \mathbb{E}_{p_\star(\mathbf{x})} \left\{ \mathbb{E}_{p_\star(\mathbf{y}|\mathbf{x})} \left\{ \log \pi_\beta(\mathbf{y}, \mathbf{x}) \right\} + \mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})} \left\{ \log \left[1 - \pi_\beta(\mathbf{y}, \mathbf{x}) \right] \right\} \right\} . \tag{2.34}$$

Here, another neural network is introduced – referred to as the discriminator $\pi_\beta : \mathcal{Y} \times \mathcal{X} \to [0, 1]$. The discriminator is tasked with distinguishing between examples generated by the model $\mathbf{y} \sim p_\phi(\mathbf{y}|\mathbf{x})$ and the real process $\mathbf{y} \sim p_\star(\mathbf{y}|\mathbf{x})$. It is trained as in Eq. (2.34) using a maximum likelihood criterion. The model (typically referred to as the *generator*) on the other hand is trained to fool the discriminator.

Optimisation Considering Eq. (2.34) separately for both ϕ and β gives objectives

$$\mathcal{C}(\beta) \triangleq \mathbb{E}_{p_\star(\mathbf{x})}\left\{\mathbb{E}_{p(\mathbf{y}|\mathbf{x})}\left\{\log \pi_\beta(\mathbf{y},\mathbf{x})\right\} + \mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{\log\left[1-\pi_\beta(\mathbf{y},\mathbf{x})\right]\right\}\right\} \tag{2.35}$$

$$\mathcal{G}(\phi) \triangleq \mathbb{E}_{p_\star(\mathbf{x})}\left\{\mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{\log\left[1-\pi_\beta(\mathbf{y},\mathbf{x})\right]\right\}\right\} \tag{2.36}$$

and the optimisation is solved iteratively maximising $\mathcal{C}(\beta)$ and minimising $\mathcal{G}(\phi)$. In practice, the training objectives Eq. (2.35) and Eq. (2.36) are estimated using Monte Carlo (see Sec. 8.1.3.1) and using the reparameterisation trick (see Sec. 8.1.3.2), re-writing

$$\mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{\log\left[1-\pi_\beta(\mathbf{y},\mathbf{x})\right]\right\} = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left\{\log\left[1-\pi_\beta\left(g_\phi\left(\boldsymbol{\epsilon},\mathbf{x}\right),\mathbf{x}\right)\right]\right\} \tag{2.37}$$

to maintain differentiability. Given a dataset of observations $\mathcal{D} = \{(\mathbf{x}_n,\mathbf{y}_n)\}_{n=1}^N$ the training objectives become

$$\mathcal{C}(\beta;\mathcal{D}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \log\{\pi_\beta(\mathbf{y}_n,\mathbf{x}_n)\} + \log\left\{1-\pi_\beta(\tilde{\mathbf{y}}_n,\mathbf{x}_n)\right\} \tag{2.38}$$

$$\mathcal{G}(\phi;\mathcal{D}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \log\left\{1-\pi_\beta(\tilde{\mathbf{y}}_n,\mathbf{x}_n))\right\} \tag{2.39}$$

where $\tilde{\mathbf{y}}_n \triangleq g_\phi(\boldsymbol{\epsilon}_n,\mathbf{x}_n)$ with $\boldsymbol{\epsilon}_n \sim p(\boldsymbol{\epsilon})$.

Crucially, this approach allows the model to be trained without ever requiring the likelihood of a particular example $p_\phi(\mathbf{y}|\mathbf{x})$ to be evaluated. The objectives $\mathcal{C}(\beta;\mathcal{D})$ and $\mathcal{G}(\phi;\mathcal{D})$ remain differentiable and so can be optimised using the conventional first-order approaches described in Sec. 8.3.

Optimality Guarantees Provided $\pi_\beta(\mathbf{y},\mathbf{x})$ and $p_\phi(\mathbf{y}|\mathbf{x})$ have sufficient capacity, solving Eq. (2.34) guarantees that the generator $p_\phi(\mathbf{y}|\mathbf{x})$ will converge to the true process $p_\star(\mathbf{y}|\mathbf{x})$ such that $p_\phi = p_\star$.

This result is derived in two stages. In the first, Eq. (2.34) is solved with respect to the discriminator parameters β. Provided that π_β has sufficient capacity, it can be shown that the *optimum* discriminator $\pi_{\beta'}$ in this case is given as

$$\pi_{\beta'}(\mathbf{y},\mathbf{x}) = \frac{p_\star(\mathbf{y}|\mathbf{x})}{p_\star(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})} \tag{2.40}$$

where $\beta' = \max_\beta \mathcal{C}(\beta)$ [30].

In the second stage, the optimum discriminator given by Eq. (2.40) is substituted into Eq. (2.36) to give

$$\tilde{\mathcal{G}}(\phi) = \log 4 - \mathbb{KL}[p_\star \| \bar{p}] + \mathbb{KL}[p_\phi \| \bar{p}] \tag{2.41}$$

$$= \log 4 - 2 \cdot \mathbb{JSD}[p_\star, p_\phi] \tag{2.42}$$

where $\bar{p}(\mathbf{y}|\mathbf{x}) = \frac{1}{2}\left[p_\star(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})\right]$ [30]. Noting that the Jensen-Shannon Divergence (JSD) $\mathbb{JSD}[p, q] \geq 0$ and is 0 if and only if $p = q$, provided that $p_\phi(\mathbf{y}|\mathbf{x})$ has sufficient capacity[6] to capture $p_\star(\mathbf{y}|\mathbf{x})$, minimising $\tilde{\mathcal{G}}(\phi)$ with respect to ϕ solves the global optimisation problem given in Eq. (2.34) and guarantees that $p_{\phi_\star} = p_\star$.

Challenges Several challenges still present themselves in this problem setting in practice [32]. Early on in training when the model $p_\phi(\mathbf{y}|\mathbf{x})$ is suboptimal it is relatively easy for the discriminator to distinguish between real and generated examples. In this case $\pi_\beta(\mathbf{y}, \mathbf{x}) \approx 0$ for all generated $\mathbf{y} \sim p_\phi(\mathbf{y}|\mathbf{x})$ and the generator criterion $\mathcal{G}(\phi)$ saturates. To counter this, instead of minimising $\mathcal{G}(\phi) \triangleq \mathbb{E}_{p(\mathbf{x})}\left\{\mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{\log 1 - \pi_\beta(\mathbf{y}, \mathbf{x})\right\}\right\}$, one option is to train the discriminator to *maximise*

$$\mathcal{G}(\phi) \triangleq \mathbb{E}_{p(\mathbf{x})}\left\{\mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{\log \pi_\beta(\mathbf{y}, \mathbf{x})\right\}\right\} \tag{2.43}$$

which provides the same fixed point in the optimisation but with much stronger gradients earlier on in training [30].

Unlike maximum likelihood approaches, there is no guarantee that the generator or discriminator training objectives will decrease or converge as training progresses. This makes debugging and terminating training challenging.

Another common difficulty is referred to as *mode collapse*. In this case the generator $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$ resorts to predicting a single realistic example, successfully fooling the discriminator, but failing to capture any variation in $\mathbf{x}$ or $\boldsymbol{\epsilon}$.

[6] In this case guaranteed by ensuring g_ϕ has sufficient capacity

Least Squares GAN In an attempt to overcome some of these limitations, the Least Squares GAN formulation [33, 34] is adopted in chapter 4 as an alternative to the original GAN criteria proposed in [30]. Instead of considering the Binary Cross Entropy (BCE) criterion given in Eq. (2.34) a least squares formulation is adopted defining

$$\mathcal{C}(\beta) \triangleq \mathbb{E}_{p_\star(\mathbf{x})}\left\{\mathbb{E}_{p_\star(\mathbf{y}|\mathbf{x})}\left\{(\pi_\beta(\mathbf{y},\mathbf{x})-b)^2\right\} + \mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{(\pi_\beta(\mathbf{y},\mathbf{x})-a)^2\right\}\right\} \tag{2.44}$$

$$\mathcal{G}(\phi) \triangleq \mathbb{E}_{p_\star(\mathbf{x})}\left\{\mathbb{E}_{p_\phi(\mathbf{y}|\mathbf{x})}\left\{(\pi_\beta(\mathbf{y},\mathbf{x})-c)^2\right\}\right\} \tag{2.45}$$

for constants a, b, c. It can be shown that if $b-c=1$ and $b-a=2$ iteratively optimising $\mathcal{C}(\beta)$ and $\mathcal{G}(\phi)$ with respect to the discriminator β and model ϕ parameters is equivalent to minimising the $\chi^2[p_\star + p_\phi \| 2p_\phi]$ which is minimised if and only if the model $p_\phi(\mathbf{y}|\mathbf{x})$ is equal to the true data generating process $p_\star(\mathbf{y}|\mathbf{x})$ such that $p_{\phi_\star} = p_\star$ (where χ^2 denotes the χ^2 divergence) [33]. The advantage of this approach is that the gradient of the generator loss is always well-defined even early on in training and so helps to avoid the saturation problem of the original GAN formulation. The least squares GAN framework has been widely adopted by the domain transfer community [35, 36] and inspired by this is adopted in chapter 4.

Normalising Flows (An aside)

Whilst outside the scope of this thesis, the method of *normalising flows* is one exception where it is possible to evaluate the likelihood $p_\phi(\mathbf{y}|\mathbf{x})$ adopting a model of the form Eq. (2.33) [37]. If g_ϕ is *bijective* and *differentiable* then $p(\mathbf{y}|\mathbf{x})$ is given explicitly as

$$p_\phi(\mathbf{y}|\mathbf{x}) = p\left(h_\phi(\mathbf{y};\mathbf{x})\right)\left|\det \nabla_{\mathbf{y}} h_\phi(\mathbf{y};\mathbf{x})\right| \tag{2.46}$$

using the *change of variables* formula where $h_\phi(\mathbf{y};\mathbf{x}) \triangleq g_\phi^{-1}(\boldsymbol{\epsilon};\mathbf{x})$ [38].

However, a general neural network g_ϕ is not typically invertible, and for high dimensional $\mathbf{y}$ calculating $|\det \nabla_{\mathbf{y}} h_\phi(\mathbf{y};\mathbf{x})|$ (which in the worst case scales as $\mathcal{O}(n_{out}^3)$) is intractable. Instead, g_ϕ is defined using *invertible* layers and such that $\nabla_{\mathbf{y}} h_\phi(\mathbf{y};\mathbf{x})$ has a particular structure (e.g. upper diagonal) allowing the Jacobian-determinant

$|\det \nabla_{\mathbf{y}} h_\phi(\mathbf{y};\mathbf{x})|$ to be calculated efficiently. This normally comes with significant restrictions on what layers are considered permissible which in turn limits the representational power of g_ϕ. Nevertheless, normalising flows could offer a powerful approach for modelling general distributions $p_\phi(\mathbf{y}|\mathbf{x})$ in the future.

2.2.3.2 CycleGAN

We now look ahead to the problem setting of chapter 4 in which the objective is to learn a radar sensor model capable of replicating radar's sensing process in simulation. Whilst it is possible to generate radar measurements in the real world $\mathcal{Y} \triangleq \{\mathbf{y}_j\}_{j=1}^{N_y}$, and world states in simulation $\mathcal{X} \triangleq \{\mathbf{x}_i\}_{n=1}^{N_x}$, example pairs $(\mathbf{x}_i, \mathbf{y}_j)$ are now temporally and spatially *unaligned*. How might a radar sensor model be learnt in this case? When training the discriminator π_β to distinguish between real, $\mathbf{y}_j$, and simulated, $\tilde{\mathbf{y}}_i = f_{\phi_y}(\boldsymbol{\epsilon}, \mathbf{x}_i)$, examples as the real input $\mathbf{x}_j$ is no longer available, the discriminator must now be optimised using a modified version of Eq. (2.35)

$$\mathcal{C}(\beta;\mathcal{D}) \triangleq \frac{1}{N}\sum_{n=1}^{N} \log\{\pi_\beta(\mathbf{y}_n)\} + \log\{1-\pi_\beta(\tilde{\mathbf{y}}_n)\} \tag{2.47}$$

dropping the discriminators conditioning on the input $\mathbf{x}$. The discriminator is now tasked with determining whether an example $\mathbf{y}$ is real independent of the input $\mathbf{x}$. As a result, the generator f_{ϕ_y} is now significantly less constrained, and in the worse case could learn to ignore the input, $\mathbf{x}$, entirely.

The CycleGAN framework offers a valuable remedy to this problem [35]. Alongside the forward process $\mathbf{y} = f_{\phi_y}(\boldsymbol{\epsilon}_y, \mathbf{x})$ the backward process is also modelled $\mathbf{x} = f_{\phi_x}(\boldsymbol{\epsilon}_x, \mathbf{y})$[7] and a second discriminator $\pi_{\beta_x} : \mathcal{X} \to [0,1]$ introduced. The advantage of this approach is that a cyclical consistency constraint

$$\mathbf{x}' = f_{\phi_x}(\boldsymbol{\epsilon}_x, \mathbf{y}) \qquad \mathbf{y}'' = f_{\phi_y}(\boldsymbol{\epsilon}_y, \mathbf{x}') \qquad \ell_y(\phi_x, \phi_y) \triangleq \|\mathbf{y} - \mathbf{y}''\|_1 \tag{2.48}$$

$$\mathbf{y}' = f_{\phi_y}(\boldsymbol{\epsilon}_y, \mathbf{x}) \qquad \mathbf{x}'' = f_{\phi_x}(\boldsymbol{\epsilon}_x, \mathbf{y}') \qquad \ell_x(\phi_y, \phi_x) \triangleq \|\mathbf{x} - \mathbf{x}''\|_1 \tag{2.49}$$

may also be imposed between the two models too further constrain the forward and backward processes to be inverses of one another. Combining the cyclical consistency

[7] Note here we consider the general case where the generators f_{ϕ_x} and f_{ϕ_y} are conditioned on auxiliary random variables $\boldsymbol{\epsilon}_x$ and $\boldsymbol{\epsilon}_y$ rather than the deterministic case originally presented in [39]

constraints above with the training criteria given in Eq. (2.45) for ϕ_x and ϕ_y, the generative parameters $\phi \triangleq \{\phi_x, \phi_y\}$ are optimised simultaneously. Meanwhile, π_{β_x} and π_{β_y} are trained by optimising Eq. (2.44). This approach is built upon in chapter 4 to learn a model capable of generating *stochastic* radar measurements in simulation.

2.2.4 Architecture Design

When designing neural networks applied to radar for mobile robotic applications, several design constraints must be accounted for. To run on a real robot, the model must be able to run in *real-time*. The scan frequency achieved by the radar used in this thesis is 4 Hz which provides a bound on the permissible execution time of the network. As motivated in chapter 1 concatenating together multiple azimuth readings to form a radar sensor measurement $\mathbf{f} \in \mathbb{R}^{R \times \Theta}$ is desirable, providing the model with a full 360° top-down view of the scene. This allows the model to more readily overcome radars complex image formation process (outlined in Sec. 2.1.3), by providing it with power measurements in their wider scene context. However, this comes at the expense of the dimensionality of the input to the model $\mathbf{f} \in \mathbb{R}^{R \times \Theta}$.

Convolutional Neural Networks (CNNs) are particularly suited for modelling high dimensional mappings between inputs and outputs which are represented as tensors (eg. $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$). The convolution operation is readily parallelised on modern graphical processing hardware allowing CNNs to be easily executed in real time. Drawing on their wide success in the image domain for robotic applications, CNN architectures will be widely exploited for radar signal processing in chapters 3 to 6.

CNNs are defined by chaining together multiple convolutional layers

$$y_{\phi_i}(\mathbf{y}_i) = h_i\left(\mathbf{W}_i * \mathbf{y}_{i-1} + \mathbf{b}_i\right) \tag{2.50}$$

which map inputs $\mathbf{y}_{i-1} \in \mathbb{R}^{C_{i-1} \times H_{i-1} \times W_{i-1}}$ to outputs $\mathbf{y}_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ where

$$(\mathbf{W} * \mathbf{y})_{c,\mathbf{v}} \triangleq \sum_{k \in \{1 \ldots C\}} \sum_{\mathbf{u} \in \mathcal{U}} \mathbf{W}_{c,k,\mathbf{u}} \mathbf{y}_{k,\mathbf{v}+\mathbf{u}+\mathbf{u}_0} \tag{2.51}$$

is the convolution operation defining $\mathcal{U} \triangleq \{1 \ldots h_i\} \times \{1 \ldots w_i\}$ and $\mathbf{u}_0 \in \mathbb{N}^2$ as a fixed offset. The parameters of the layer $\phi_i \triangleq \{\mathbf{W}_i, \mathbf{b}_i\}$ are in this instance shared

between pixels reducing the size of the parameter space considerably. Any non-linear function may be used as the *activation* h_i such as (tanh, sigmoid, polynomials) [40], however, in recent years ReLU activations $h_i(a) \triangleq \max(0, a)$ [41, 42] have become a popular choice thanks to their well-defined gradients for any input $a > 0$.[8]

Stacking multiple convolutional layers together $y_\phi = y_{\phi_L} \circ \cdots \circ y_{\phi_1}$ allows the convolutional kernel to gradually distribute information spatially through the image. Spatial information propagation is aided further through pooling strategies. In this case the output from Eq. (2.50) is down-sampled to distribute information more quickly through the feature space using a pooling layer

$$p : \mathbb{R}^{C_i \times H_{i-1} \times W_{i-1}} \rightarrow \mathbb{R}^{C_i \times H_i \times W_i} \tag{2.52}$$

where $H_i < H_{i-1}$ and $W_i < W_{i-1}$. Different down-sampling strategies may be deployed such as taking the maximum value over a window (i.e max-pooling) [45, 46], summing over the window, or considering only one in every k pixels (using *strided* convolutions).

Whilst, pooling layers are necessary to ensure each pixel in the output covers a sufficiently *large receptive field* in the input (in a tractable number of layers), this comes at the expense of spatial resolution which is gradually reduced through the network. When considering mappings to high-dimensional outputs, such as those in chapters 3 to 6, the spatial dimension is restored using up-sampling strategies. In the simplest case bi-linear sampling can be used to up-sample feature maps. More sophisticated up-sampling approaches utilise the convolutional transpose operation (sometimes referred to as de-convolutions) [47]. In this case the network is often composed of an encoder – distributing information spatially through the feature space using a combination of convolutional layers and pooling – and a decoder – using an up-sampling strategy (e.g. sampling or de-convolutions) to restore the spatial dimension to the desired output resolution [47].

It is sometimes also helpful to provide later layers access to information from earlier layers – for example when considering outputs which rely on fine-grained

[8] Other popular choices include ELU [43] and LeakyReLU [44] which modify ReLU such that non-zero gradients are defined for $a < 0$.

spatial features in the input. This is achieved using skip connections [48, 49] where feature maps at earlier layers are concatenated with feature maps at later layers and combined using convolutions.[9] Similar to residual connections, skip connections have the added benefit of allowing gradient signals from earlier layers to easily pass through the network. Networks employing an encoder-decoder architecture with skip connections are commonly referred to as *Unets* [48]. Unet architectures are widely exploited in chapters 3 to 6.

CNNs For The Radar Datatype

Convolutional neural networks are particularly suited to inputs and outputs that can be represented as tensors $\mathbf{x} \in \mathbb{R}^{H \times W}$. The first and simplest approach to representing radar measurements as a tensor is to concatenate sensor measurements over multiple azimuths considering $\mathbf{x} \in \mathbb{R}^{R \times \Theta}$. As a second approach, it is possible to convert the radar measurement from polar to Cartesian co-ordinates (using bi-linear interpolation for example) generating an input $\mathbf{x} \in \mathbb{R}^{H \times W}$. Whilst many of radars noise artifacts are relatively simple functions in polar co-ordinates – such as saturation and halo artifacts which appear as bright streaks at a particular azimuth or range – the appearance of objects will change drastically with range in this case (and vice-versa for a Cartesian input). Both polar and Cartesian inputs are considered in later chapters.

Just as it is possible to convert between polar and Cartesian representations at the input level, it is also possible to perform this conversion at the feature level. Using differentiable interpolation methods this approach is exploited in chapter 3 to learn a mapping from a polar input to a Cartesian output. In this case, features in the encoder are passed to the decoder using polar to Cartesian Transformer Units [50].

Optimisation

When considering deep models, the parameter space is of a high dimensionality and in general – as they are non-linear in the parameters ϕ – the training objective $\mathcal{L}(\phi)$

[9] In the simple case this is achieved by only concatenating feature maps at the same spatial resolutions

is typically non-convex. Second-order non-linear optimisation methods that rely on calculating the Hessian of the loss function with respect to the network parameters are infeasible due to the high dimensionality of ϕ. Instead, first-order methods are used, using the derivative of the loss with respect to the parameters ϕ to update the parameter values at each iteration, as discussed in more detail in Sec. 8.3. In chapters chapters 3 to 6 the Adam optimiser will be the optimisation method of choice [51].

2.3 Conclusion

Several deep approaches applied to radar across a range of tasks in robotics, are now proposed. In chapter 3 a deep and probabilistic approach to radar sensor modelling is developed. Next, in chapter 4 a deep approach is proposed for replicating radars sensing process in simulation, with the ultimate goal of training of new-models for downstream tasks, using simulated radar measurements. Finally, in chapter 5 and chapter 6 deep models are explored for the radar odometry task.

In each case, chapters 3 to 6 are presented in their original manuscript format corresponding to the publications listed in Sec. 1.3. Alongside, this in chapter 5 an alternative probabilistic derivation for the proposed approach is presented which provides further technical insights, and elaborates on several of the methods proposed in the original publication.

3

Probably Unknown: Deep Inverse Sensor Modelling In Radar

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

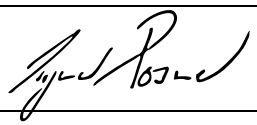
Title of Paper	Probably Unknown: Deep Inverse Sensor Modelling In Radar
Publication Status	X Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	R. Weston, S. Cen, P. Newman, I. Posner. "Probably Unknown: Deep Inverse Sensor Modelling in Radar In: *International Conference On Robotics and Automation 2019 (ICRA Montreal 2019)*

Student Confirmation

Student Name:	Robert Weston		
Contribution to the Paper	Developed Approach, Ran Experiments, Wrote Manuscript		
Signature		Date	7 / 04 / 2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Prof. Ingmar Posner		
Supervisor comments	I confirm that the candidate made the contributions specified above.		
Signature		Date	7/4/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

arXiv:1810.08151v2 [cs.RO] 10 May 2019

Probably Unknown: Deep Inverse Sensor Modelling In Radar

Rob Weston, Sarah Cen, Paul Newman and Ingmar Posner

***Abstract*— Radar presents a promising alternative to lidar and vision in autonomous vehicle applications, able to detect objects at long range under a variety of weather conditions. However, distinguishing between occupied and free space from raw radar power returns is challenging due to complex interactions between sensor noise and occlusion.**

To counter this we propose to learn an Inverse Sensor Model (ISM) converting a raw radar scan to a grid map of occupancy probabilities using a deep neural network. Our network is self-supervised using partial occupancy labels generated by lidar, allowing a robot to learn about world occupancy from past experience without human supervision. We evaluate our approach on five hours of data recorded in a dynamic urban environment. By accounting for the scene context of each grid cell our model is able to successfully segment the world into occupied and free space, outperforming standard CFAR filtering approaches. Additionally by incorporating heteroscedastic uncertainty into our model formulation, we are able to quantify the variance in the uncertainty throughout the sensor observation. Through this mechanism we are able to successfully identify regions of space that are likely to be occluded.

I. INTRODUCTION

Occupancy grid mapping has been extensively studied [1], [2] and successfully utilised for a range of tasks including localisation [3], [4] and path-planning [5]. One common approach to occupancy grid mapping uses an inverse sensor model (ISM) to predict the probability that each grid cell in the map is either *occupied* or *free* from sensor observations. Whilst lidar systems provide precise, fine-grained measurements, making them an obvious choice for grid mapping, they fail if the environment contains fog, rain, or dust [6]. Under these and other challenging conditions, FMCW radar is a promising alternative that is robust to changes in lighting and weather and detects long-range objects, making it well suited for use in autonomous transport applications.

However, two major challenges must be overcome in order to utilise radar to this end. Firstly, radar scans are notoriously difficult to interpret due to the presence of several pertinent noise artefacts. Secondly, by compressing information over a range of heights onto a dense 2D grid of power returns identifying occlusion becomes difficult. The complex interaction between occlusion and noise artefacts introduces uncertainty in the state of occupancy of each grid cell which is *heteroscedastic*, varying from one world location to another based on scene context, and *aleatoric* [7], inherent in radar data by way of the scan formation process.

In order to successfully reason about world occupancy, we posit that a model that is able to reason about scene context is essential. To this end, we formulate the problem of determining an ISM as a segmentation task, leveraging a deep network to learn the probability distribution of occupancy from raw data alone. This allows us to successfully determine regions of space that are likely to be occupied and free in light of challenging noise artefacts. Simultaneously, by explicitly modelling heteroscedastic uncertainty, we are able to quantify the latent uncertainty associated with each world cell arising through occlusion. Utilising approximate variational inference we are able to train our network using self-supervision relying on partial labels automatically generated from occupancy observations in lidar.

Authors are from the Oxford Robotics Institute (ORI)
{robw,sarah,pnewman,ingmar}@robots.ox.ac.uk

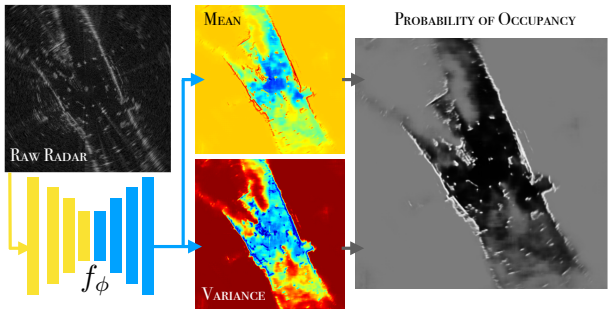


Fig. 1. Our network learns the distribution of occupancy from experience alone. By reasoning about scene context it is able to successfully identify regions of space that are likely to be occupied and free. The uncertainty associated with each grid cell is allowed to vary throughout the scene by predicting the noise standard deviation alongside the predicted logit of each grid cell. These are combined to generate a grid map of occupancy probabilities. The uncertainty predicted by our network can be used to successfully identify regions of space that are likely to be occluded.

We train our model on real-world data generated from five hours of urban driving and successfully distinguish between occupied and free space, outperforming constant false-alarm rate (CFAR) filtering in average intersection over union performance. Additionally we show that by modelling heteroscedastic uncertainty we are able to successfully quantify the uncertainty arising through the occlusion of each grid cell.

II. RELATED WORK

Inverse sensor models (ISMs) [1] are used to convert noisy sensor observation to a grid map of occupancy probabilities. For moving platforms, a world occupancy map can then be sequentially generated from an ISM, multiple observations, and known robot poses using a binary Bayes filter [8]. Using lidar data, ISMs are typically constructed using a combination of sensor-specific characteristics, experimental data, and empirically-determined parameters [9], [10], [11]. These human-constructed ISMs struggle to model challenging radar defects and often utilise limited local information to predict each cell's occupancy without accounting for scene context.

Instead, raw radar scans are often naively converted to binary occupancy grids using classical filtering techniques that distinguish between objects (or targets) and free space (or background). Common methods include CFAR [12] and static thresholding. However, both return binary labels rather than probabilities, and neither is capable of addressing all

types of radar defects or capturing occlusion. Additionally, the most popular approach, CFAR, imposes strict assumptions on the noise distribution and requires manual parameter tuning. In contrast, using deep learning methods, as first proposed by [13], allows the distribution of world occupancy to be learned from raw data alone, accounting for the complex interaction between sensor noise and occlusion through the higher level spatial context of each grid cell.

In order to capture uncertainty that varies from one grid cell to the next we incorporate heteroscedastic uncertainty into our formulation inspired by [7]. Our variational reformulation of [7] is closely related to the seminal works on variational inference in deep latent variable models [14], [15] and their extension to conditional distributions [16].

Drawing on the successes of deep segmentation in biomedical applications, [17] and vision [18] we reformulate the problem of learning an inverse sensor model as neural network segmentation. Specifically, we utilise a U-net architecture with skip connections [19]. In order to map from an inherently polar sensor observation to a Cartesian map we utilise Polar Transformer Units (PTUs) [20].

III. DEEP INVERSE SENSOR MODELLING IN RADAR

A. Setting

Let $\boldsymbol{x} \in \mathbb{R}^{\Theta \times R}$ denote a full radar scan containing Θ azimuths of power returns at R different ranges for each full rotation of the sensor. Partitioning the world into a $H \times W$ grid, $\boldsymbol{y} \in \{0,1\}^{H \times W}$ gives the occupancy state of each grid cell, where $\boldsymbol{y}^{u,v} = 1$ if cell (u,v) is *occupied* and $\boldsymbol{y}^{u,v} = 0$ if (u,v) is *free*. *Partial* measurements of occupancy $\hat{\boldsymbol{y}}$ are determined by combining the output of multiple 3D lidars and projecting the returns over a range of heights onto a 2D grid. In order to separate the region of space where no labels exist most likely as a consequence of full occlusion, from space that is likely to only be partially occluded or for which no labels exist due to a limited field of view of the lidar sensors, the observability state of each cell $\boldsymbol{o}^{u,v}$ is recorded as 0, 1 or 2 corresponding to *unobserved*, *observed* and *partially observed* space respectively. The full labelling procedure is described in Figure 2. This process is repeated for N radar-laser pairs to generate a data set $\mathcal{D} = \{\boldsymbol{x}^n, (\hat{\boldsymbol{y}}, \boldsymbol{o})^n\}_{n=1}^N$ of training examples from which we aim to learn an inverse sensor model $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}} \in [0,1]^{H \times W}$ such that $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}}^{u,v} = p(\boldsymbol{y}^{u,v} = 1|\boldsymbol{x})$ gives the probability that cell (u,v) is occupied dependent on the *full* radar scan $\boldsymbol{x}$

B. Heteroscedastic Aleatoric Uncertainty and FMCW Radar

FMCW Radar is an inherently noisy modality suffering from speckle noise, phase noise, amplifier saturation and ghost objects. These conspire to make the distinction between occupied and free space notoriously difficult. A radar's long range as well as its ability to penetrate past first returns make it attractive but also challenging. In particular, a radar's capacity for multiple returns along an azimuth implies varying degrees of uncertainty depending on scene context: the distinction between occupied and free space becomes

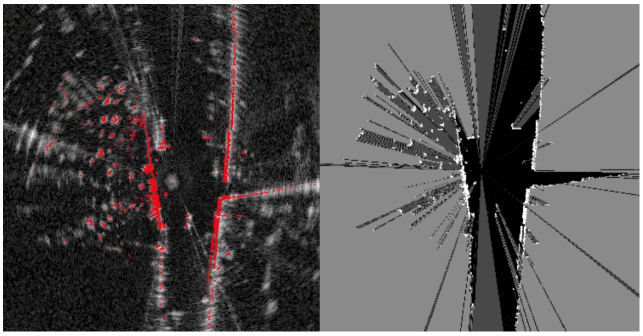


Fig. 2. Generated training labels from lidar. The image on the left shows the lidar points (red) projected into a radar scan $\boldsymbol{x}$ converted to Cartesian co-ordinates for visualisation. The right image shows the generated training labels. Any grid cell (u,v) with a lidar return is labelled as occupied $\hat{\boldsymbol{y}}^{u,v} = 1$ (white). Ray tracing along each azimuth, the space immediately in front of the first return is labelled as $\hat{\boldsymbol{y}}^{u,v} = 0$ (black), the space between the first and last return or along azimuths in which there is no return is labelled as *partially observed*, $\boldsymbol{o}^{u,v} = 2$, (dark grey) and the space behind the last return is labelled as *unobserved*, $\boldsymbol{o}^{u,v} = 0$, (light grey). Any space that is labelled as occupied or free is labelled as *observed*, $\boldsymbol{o}^{u,v} = 1$

increasingly uncertain as regions of space become partially occluded by objects. Examples of each of these problems are further explained in Figure 3. As such, high power returns do not always denote occupied and likewise, low power returns do not always denote free.

Uncertainties in our problem formulation depend on the world scene through a complex interaction between scene context and sensor noise, and are inherent in our data as a consequence of the image formation process. As such they are, heteroscedastic as they depend on scene context and aleatoric as they are ever present in our data [7]. In order to successfully determine world occupancy from an inherently uncertain radar scan we seek a model that explicitly captures heteroscedastic aleatoric uncertainty. By framing this problem as a deep segmentation task we leverage the power of neural networks to learn an ISM which accounts for scene context in order to determine – from raw data alone – *occupied* from *free* space in the presence of challenging noise artefacts. Simultaneously, as a result of our heteroscedastic uncertainty formulation we are also able to learn which regions of space are inherently uncertain because of occlusion.

C. Modelling Heteroscedastic Aleatoric Uncertainty

Instead of assuming that the uncertainty associated with each grid cell is fixed, as is typically assumed in standard deep segmentation approaches, by using a heteroscedastic model the uncertainty in each grid cell $\boldsymbol{\gamma}_\phi(\boldsymbol{x})$ is allowed to vary. This is achieved by introducing a normally distributed latent variable $\boldsymbol{z}^{u,v}$ associated with each grid cell [7] and predicting the noise standard deviation $\boldsymbol{\gamma}_\phi(\boldsymbol{x})$ alongside the predicted logit $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ of each each $\boldsymbol{z}^{u,v}$ with a neural network f_ϕ :

$$p_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\gamma}_\phi(\boldsymbol{x})\boldsymbol{I}) \tag{1}$$

$$[\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\gamma}_\phi(\boldsymbol{x})] := f_\phi(\boldsymbol{x}) \tag{2}$$

Assuming a likelihood $p(\boldsymbol{y}^{u,v} = 1|\boldsymbol{z}^{u,v}) = \mathsf{Sigmoid}(\boldsymbol{z})$, the probability that cell $\boldsymbol{y}^{u,v}$ is occupied is then given by

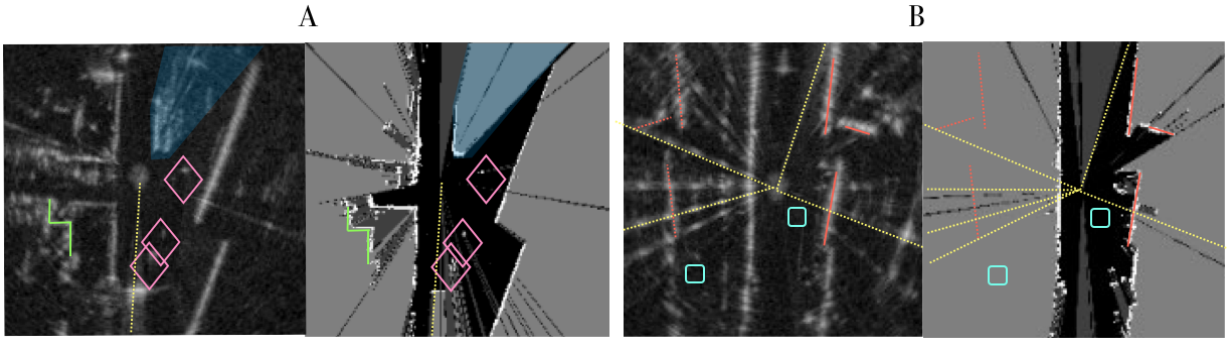


Fig. 3. Raw radar and the lidar ground truth. An ISM must be able to pick out faint objects, such as cars (pink diamonds), from the background speckle noise, in light of challenging noise artefacts such as saturation (yellow lines). In addition, an ISM must be able to determine which regions of space are likely to be occluded such as the space behind buses (highlighted blue) in light of almost identical local appearances (blue cyan boxes). Finally an ISM should be able to distinguish ghost objects (dotted orange) from true second returns (green lines).

marginalising out the uncertainty associated with $\boldsymbol{z}$:

$$p(\boldsymbol{y}^{u,v}|\boldsymbol{x}) = \int p(\boldsymbol{y}^{u,v}|\boldsymbol{z}^{u,v}) p_\phi(\boldsymbol{z}^{u,v}|\boldsymbol{x}) d\boldsymbol{z}^{u,v} \tag{3}$$

Unfortunately the integral in (3) is intractable and is typically approximated using Monte-Carlo sampling and the reparameterization trick [7]. Instead, by introducing an analytic approximation in Section III-E we show that we can accurately and efficiently approximate (3) without resorting to sampling.

One final problem remains. We expect our model to be inherently uncertain in occluded space for which no lidar training labels are available. How do we train f_ϕ whilst explicitly encoding an assumption that in the absence of training labels we expect our model to be uncertain? In Section III-D we propose to solve this problem by introducing a normally distributed prior $p(\boldsymbol{z})$ on the region of space for which no training labels exist utilising the variational inference framework.

D. Training with Partial Observations

In order to encode an assumption that in the absence of training data we expect our model to be explicitly uncertain we introduce a prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\gamma}\boldsymbol{I})$ on the uncertainty associated with the occluded scene which our network reverts back to in the absence of a supervised training signal. To do this, we begin by treating $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ as an approximate posterior to $p(\boldsymbol{z}|\boldsymbol{y})$ induced by the joint $p(\boldsymbol{z}, \boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$ where,

$$p(\boldsymbol{y}|\boldsymbol{z}) := \prod_{u,v} \mathsf{Bern}(\boldsymbol{y}^{u,v}|\boldsymbol{p}^{u,v}_{\boldsymbol{y}|\boldsymbol{z}}) \tag{4}$$

$$\boldsymbol{p}^{u,v}_{\boldsymbol{y}|\boldsymbol{z}} = p(\boldsymbol{y}^{u,v} = 1) = \mathsf{Sigmoid}(\boldsymbol{z}^{u,v}) \tag{5}$$

$$p(\boldsymbol{z}) := \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \gamma I) \tag{6}$$

$\mathsf{Sigmoid}$ and $\mathsf{Bern}(y|p) = p^y(1-p)^{1-y}$ denote the element-wise sigmoid function and Bernoulli distribution.

Next given a set of observations $\mathcal{D}$, we consider determining our parameters ϕ by maximising the variational lower bound,

$$\mathcal{L}(\phi; \mathcal{D}) = \sum_n \mathcal{L}^n(\phi) \tag{7}$$

$$\mathcal{L}^n(\phi) = \mathbb{E}_{p_\phi(\boldsymbol{z}|\boldsymbol{x}^n)}[\log p(\boldsymbol{y}^n|\boldsymbol{z})] - d_{kl}[p_\phi(\boldsymbol{z}|\boldsymbol{x}^n)||p(\boldsymbol{z})] \tag{8}$$

where d_{kl} denotes KL divergence. The first term in $\mathcal{L}^n(\phi)$ is the expected log-likelihood under the approximate posterior $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ which, when optimised, forces the network to maximise the probability of each occupancy label $\boldsymbol{y}$. The second term forces $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ towards the prior $p(\boldsymbol{z})$.

Crucially, by only evaluating the log-likelihood term in the labelled region of space and only evaluating the KL divergence term in occluded space, we are able to train our network to maximise the probability of our labels whilst explicitly encoding an assumption that in the absence of training labels we expect our network to be inherently uncertain. The latter is achieved by setting the prior to $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \gamma I)$ corresponding to an assumption that occluded space is equally likely to be free or occupied with a fixed uncertainty γ. We tested multiple values of γ and found that setting $\gamma = 1$ gave good results.

For a Gaussian prior and approximate posterior the KL divergence term can be determined analytically, whilst the expected log-likelihood is estimated using the reparameterization trick [15] by sampling $\boldsymbol{z}^l = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + \boldsymbol{\gamma}_\phi(\boldsymbol{x}) \circ \boldsymbol{\epsilon}^l$ where $\boldsymbol{\epsilon}^l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The expected log-likelihood is then approximated as $\mathbb{E}_{p_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] \approx -\frac{1}{L}\sum_l \left(\sum_{u,v} \mathbb{H}[\boldsymbol{y}^{u,v}, \boldsymbol{p}^{l,u,v}_{\boldsymbol{y}|\boldsymbol{z}}]\right)$ where $\mathbb{H}$ denotes binary cross entropy.

Finally our loss function becomes

$$\hat{\mathcal{L}}^n(\phi) = \frac{\bar{\omega}}{L} \sum_{l,u,v} \mathbb{I}(\boldsymbol{o}^{u,v} = 1) \mathbb{H}_\alpha[\hat{\boldsymbol{y}}^{n,u,v}, \boldsymbol{p}^{n,l,u,v}_{\boldsymbol{y}|\boldsymbol{z}}] + \sum_{u,v} \mathbb{I}(\boldsymbol{o}^{u,v} = 0) d_{kl}[p_\phi(\boldsymbol{z}^{u,v}|\boldsymbol{x}^n)||p(\boldsymbol{z}^{u,v})] \tag{9}$$

$$\hat{\mathcal{L}}(\phi; \mathcal{D}) = \frac{1}{N} \sum_n \hat{\mathcal{L}}^n(\phi) \tag{10}$$

where $\mathbb{I}$ denotes the indicator function which is equal to 1 if its condition is met and 0 otherwise.

In order to ensure that labelled and unlabelled data contribute equally to our loss we re-weight the likelihood term with $\bar{\omega} = \omega HW / (\sum_{uv} \mathbb{I}(\boldsymbol{o}^{u,v} = 1))$. The hyper-parameter ω is used to weight the relative importance between our prior and approximate evidence. As there is also a significant class imbalance between occupied and free space we use weighted binary cross entropy $\mathbb{H}_\alpha$ where the contribution from the occupied class is artificially inflated by weighting

each occupied example by a hyper-parameter α. Note that in the partially observed region $\boldsymbol{o}^{u,v} = 2$ there is no loss.

E. Inference

Given a trained model $p_{\phi_*}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{\phi_*}(\boldsymbol{x}), \boldsymbol{\gamma}_{\phi_*}(\boldsymbol{x}))$ we now wish to determine the probability that each cell is occupied given input $\boldsymbol{x}$ by marginalising out the uncertainty associated with the latent variable $\boldsymbol{z}$:

$$p(\boldsymbol{y}^{u,v}|\boldsymbol{x}) := \int p(\boldsymbol{y}^{u,v}|\boldsymbol{z}^{u,v}) p_{\phi_*}(\boldsymbol{z}^{u,v}|\boldsymbol{x}) d\boldsymbol{z}^{u,v} \tag{11}$$

However, for likelihood $p(\boldsymbol{y}^{u,v}|\boldsymbol{z}^{u,v}) = \mathsf{Sigmoid}(\boldsymbol{z}^{u,v})$ no exact closed form solution exists to this integral. Instead of resorting to Monte Carlo sampling we approximate the sigmoid function with a probit function and use the result that a Gaussian distribution convolved with a probit function is another probit function [21]. Following this analysis, it can be shown that,

$$p(\boldsymbol{y}^{u,v} = 1|\boldsymbol{x}) \approx \mathsf{Sigmoid}\left(\frac{\boldsymbol{\mu}^{u,v}_{\phi_*}}{\boldsymbol{s}^{u,v}_{\phi_*}}\right) \tag{12}$$

where $\boldsymbol{s}^{u,v}_{\phi_*} = (1 + (\boldsymbol{\gamma}^{u,v}_{\phi_*}\sqrt{\pi/8})^2)^{1/2}$, $\boldsymbol{\mu}^{u,v}_{\phi_*} = \boldsymbol{\mu}^{u,v}_{\phi_*}(\boldsymbol{x})$ and $\boldsymbol{\gamma}^{u,v}_{\phi_*} = \boldsymbol{\gamma}^{u,v}_{\phi_*}(\boldsymbol{x}_*)$. This allows us to efficiently calculate $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}}$ as,

$$[\boldsymbol{\mu}_{\phi_*}, \boldsymbol{\gamma}_{\phi_*}] = f_{\phi_*}(\boldsymbol{x}) \tag{13}$$

$$\boldsymbol{s}_{\phi_*} = (1 + (\boldsymbol{\gamma}_{\phi_*}\sqrt{\pi/8})^2)^{1/2} \tag{14}$$

$$\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}} := \mathsf{Sigmoid}\left(\frac{\boldsymbol{\mu}_{\phi_*}}{\boldsymbol{s}_{\phi_*}}\right) \tag{15}$$

Figure 4 shows $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}}$ approximated using (15) and Monte Carlo sampling for varying $\boldsymbol{\mu}_{\phi_*}$ and $\boldsymbol{\gamma}_{\phi_*}$. The Monte Carlo estimate takes of the order 10^4 samples to converge, whilst the analytic approximation provides a close approximation to the converged Monte Carlo estimate.

In equation (15) the predicted logit $\boldsymbol{\mu}_{\phi_*}$ can be thought of as giving the score associated with labelling an example as occupied; intuitively the higher the score the higher the probability that each cell is occupied. In contrast, the predicted deviation $\boldsymbol{\gamma}_{\phi_*}$ increases the entropy in the predicted occupancy distribution independent of the cells predicted score and captures uncertainties that cannot be easily explained by the predicted score alone.

IV. RESULTS

In this Section we show that our model, despite challenging noise artefacts, is able to successfully segment the world into occupied and free space achieving higher mean Intersection over Union (IoU) scores than cell averaging CFAR filtering approaches. In addition to this we are also able to explicitly identify regions of space that are likely to be occluded through the uncertainties predicted by our network. We provide several qualitative examples of our model operating in challenging real world environments and study the effects of our prior on our network output through an ablation study.

A. Experimental Set-Up

A Navtech CTS350x FMCW radar (without Doppler) and two Velodyne HDL32 lidars were mounted to a survey vehicle and used to generate over 78000 (90%) training examples and 8000 (10%) test examples from urban scenes. The output from the two lidars was combined from 0.7m below the roof of the vehicle to 1m above and projected onto a 600×600 grid, with a spatial resolution of 0.3m, generating a $180\text{m} \times 180\text{m}$ world occupancy map, following the procedure described in Section III-A. To account for differences in the frequency of our radar (4Hz) and lidar (10Hz) the occupancy map was ego-motion compensated such that the Cartesian map corresponds to the time stamps of each radar azimuth.

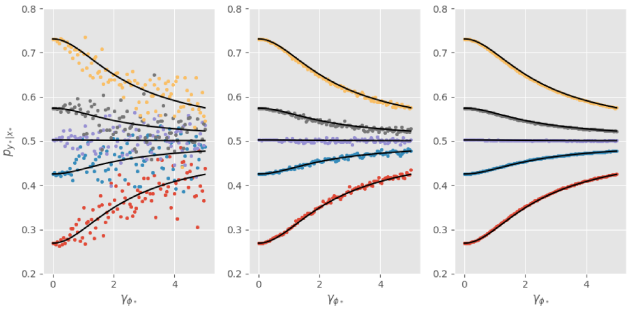


Fig. 4. Predicted occupancy probabilities $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}}$ as a function of predicted standard deviation $\boldsymbol{\gamma}_{\phi_*}$ using the analytic approximation given by (15) (black) vs Monte Carlo approximation with $L = 10^2$ (left), $L = 10^4$ (middle) and $L = 10^6$ (right) samples. Each colour corresponds to a different mean $\boldsymbol{\mu}_{\phi_*}$ with [yellow, grey, purple, blue, red] corresponding to means $[-1, -0.3, 0.01, 0.3, 1]$ respectively. It is seen that the MC estimate has high variance taking of the order 10^6 samples to converge to the analytic approximation. On the other hand the analytic approximation closely resembles the converged Monte Carlo estimate.

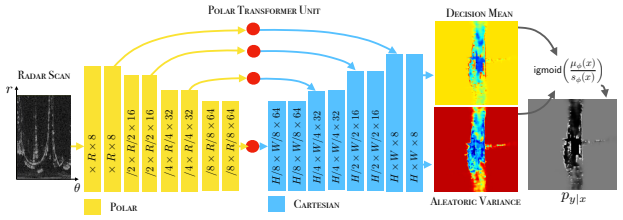


Fig. 5. Our network architecture takes in a polar radar scan $\boldsymbol{x} \in \mathbb{R}^{\Theta \times R}$ and maps it to Cartesian grids of mean utility $\boldsymbol{\mu}_\phi$ and aleatoric noise scale $\boldsymbol{s}_\phi = (1 + (\boldsymbol{\gamma}_\phi\sqrt{\pi/8})^2)^{1/2}$. Our network is composed of a polar (yellow) encoder and a Cartesian (blue) decoder. At each polar to Cartesian interface there is a polar transformer unit (red circle). Each blue rectangle corresponds to 2 convolutions followed by a max pool.

Figure 5 shows our network architecture in which a polar encoder takes the raw radar output and generates a polar feature tensor through repeated applications of 4×4 convolutions and max pooling before a Cartesian decoder maps this feature tensor to a grid of mean logits $\boldsymbol{\mu}_\phi(\boldsymbol{x}) \in \mathbb{R}^{H \times W}$ and standard deviations $\boldsymbol{\gamma}_\phi(\boldsymbol{x}) \in (0, \infty)^{H \times W}$ which are converted to a grid of probabilities through (15). Information is allowed to flow from the encoder to the decoder through skip connections, where polar features $\boldsymbol{u}$ are converted to Cartesian features $\boldsymbol{v}$ through bi-linear interpolation, with a fixed polar to Cartesian grid [20]. In all experiments we trained our model using the ADAM optimiser [22], with a learning rate of 0.001, batch size 16 for 100 epochs and randomly rotated each input output pair about the origin, minimising the loss proposed in (9) with $L = 25$ samples. Experimentally it was found that setting $\alpha = 0.5$ gave

TABLE I
COMPARING OUR APPROACH TO CLASSICAL DETECTION METHODS USING INTERSECTION OVER UNION

Method	Intersection over Union		
	Occupied	Free	Mean
CFAR (1D polar)	0.24	**0.92**	0.5
CFAR (2D Cartesian)	0.20	0.90	0.55
Static thresholding	0.19	0.77	0.48
Deep ISM (our approach)	**0.35**	0.91	**0.63**

the best results in terms of IoU performance against the lidar labels. Unless otherwise stated, the model evidence importance was set to $\omega = 1$.

B. Detection Performance of Deep ISM vs Classical Filtering Methods

We compare the detection performance of our approach against cell averaging CFAR [12] applied in 1D (along range) for polar scans and in 2D for Cartesian scans by determining the quantity of occupied and unoccupied space successfully segmented in comparison to the ground truth labels generated from lidar in observed space. Due to class frequency imbalance, we use the mean Intersection Over Union (IoU) metric [23]. The optimum number of guard cells, grid cells and probability of false alarm, for each CFAR method, was determined through a grid search maximising the mean IoU of each approach on training data. For our method, each cell was judged as occupied or free based on a 0.5 probability threshold on $\boldsymbol{p}_{\boldsymbol{y}|\boldsymbol{x}}$. A 2m square in the centre of the occupancy map, corresponding to the location of the survey vehicle, was marked as unobserved.

The results form the test data set for each approach are shown in table I and show that our approach outperforms all the tested CFAR methods, increasing the performance in occupied space by 0.11, whilst achieving almost the same performance in free space leading to a mean IoU of 0.63. Our model is successfully able to reason about occupied space in light of challenging noise artefacts. In contrast, the challenge in free space is not in identification, with free space typically being characterised by low power returns, but in distinguishing between observed and occluded regions, a challenge which is missed entirely by the IoU metric. Figure 6a shows how our model is able to successfully determine space that is likely to be unknown because of occlusion and is able to clearly distinguish features, such as cars that are largely missed in CFAR. An occupancy grid of size 600×600 can be generated at around 14Hz on a NVIDIA Titan Xp GPU. Which is significantly faster than real time for radar with a frequency of 4Hz.

C. Uncertainty Prediction

As described in Section III-E, by incorporating aleatoric uncertainty into our formulation, the latent uncertainty associated with each grid cell is allowed to vary by predicting the standard deviation of each cell $\boldsymbol{\gamma}_\phi(\boldsymbol{x})$ alongside the predicted logit $\boldsymbol{\mu}_\phi(\boldsymbol{x})$. In this section we investigate the uncertainties that are captured by this mechanism.

To do this we gradually increase a threshold on the maximum allowable standard deviation of each cell $\boldsymbol{\gamma}_\phi(\boldsymbol{x})$ labelling any cell that falls below this threshold as either occupied (white) or free (black), whilst every cell above the threshold is labelled as unknown (grey). The result of this process is illustrated in Figure 6d.

The standard deviation predicted by our network largely captures uncertainty caused by occlusion, which, independent of the true underlying state of occupancy, results in space that is inherently unknown. From least likely to most likely to be occluded, we move from high power returns labelled as occupied, to a region nearby and up to the first return, to space that lies in partial and full occlusion. This ray tracing mechanism is largely captured by the standard deviation $\boldsymbol{\gamma}_\phi(\boldsymbol{x})$ predicted by our network.

D. Qualitative Results

Finally, we provide several qualitative examples of our model operating in challenging real world environments and investigate how the strength of our prior term in (9) effects the occupancy distribution predicted by our model.

Figure 6c gives qualitative examples taken from the test set. Our network is able to successfully reason about the complex relationship between observed and unobserved space in light of challenging noise artefacts. In Figure 6b we vary the relative importance between the likelihood and KL divergence term by varying the hyper-parameter ω in (9). Increasing ω increases the relative importance of the likelihood term and leads to an ISM which is able to more freely reason about regions of space for which no labels exist during training, using the labels available in the observed scene. In the limit, of high ω the model is no longer able to successfully identify regions of space that are likely to be occluded, predicting all low power returns as free with a high probability.

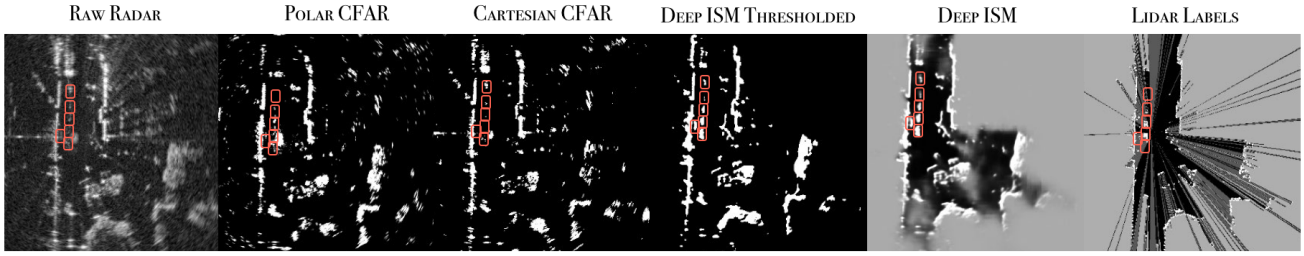
V. CONCLUSION

By using a deep network we are able to learn an inherently probabilistic ISM from raw radar data that is able to identify regions of space that are likely to be occupied in light of complex interactions between noise artefacts and occlusion. By accounting for scene context, our model is able to outperform CFAR filtering approaches. Additionally, by modelling heteroscedastic uncertainty we are able to capture the variation of uncertainty throughout the scene, which can be used to identify regions of space that are likely to be occluded. Our network is self-supervised using only partial labels generated from a lidar, allowing a robot to learn about the occupancy of the world by simply traversing an environment.

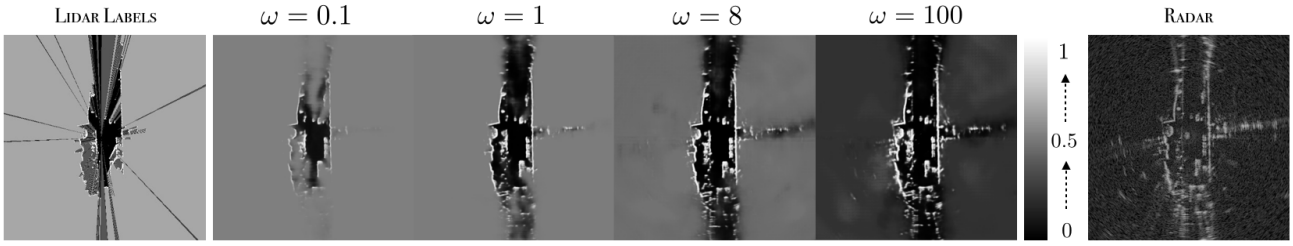
At present our approach operates under a static world assumption. In future work we hope to incorporate scene dynamics into our formulation allowing a robot to identify cells that are likely to be dynamic in addition to occupied or free.

ACKNOWLEDGMENT

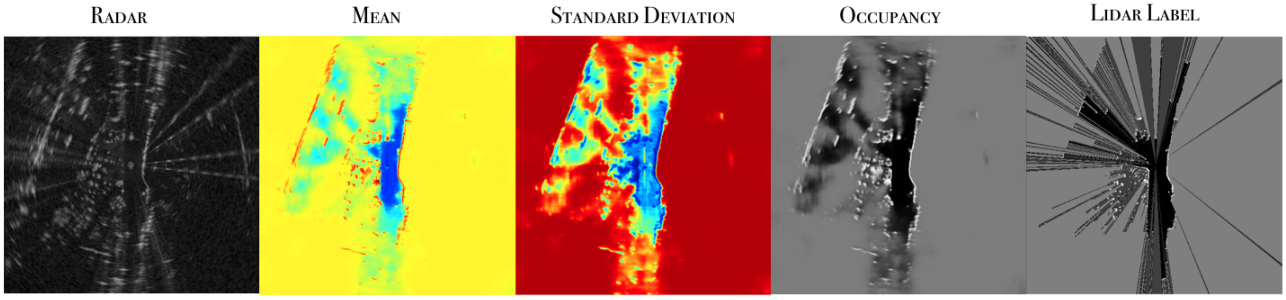
The authors would like to thank Oliver Bartlett and Jonathan Attias for proof reading a draft of the paper,



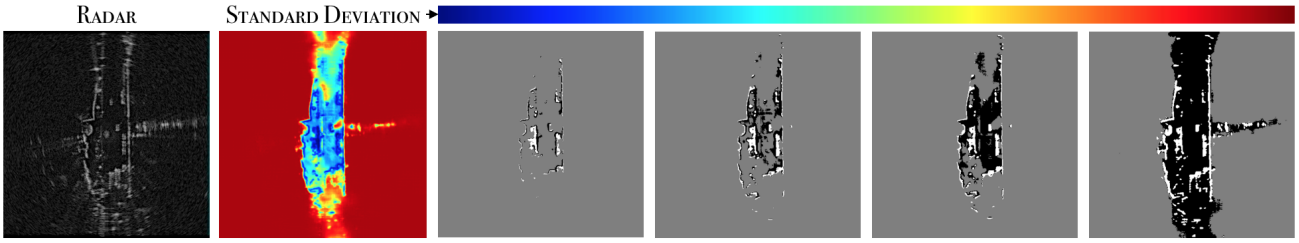
(a) The detection performance of our approach vs classical filtering methods with black representing predicted free and white representing predicted occupied by each approach. In comparison to CFAR our approach results in crisp and clean detections in observed and unobserved space. The red rectangles highlight cars that are clearly detected by our approach which are largely missed by CFAR. In addition, our model is able to successfully reason about what in the scene is likely to be unknown due to occlusion.



(b) The predicted probability of occupancy for different values of likelihood importance ω. As ω is increased our model becomes increasingly less conservative, reasoning in the unobserved region of space based on labels in the observed region.



(c) Our model successfully identifies occupied free and occluded space in challenging real world environments.



(d) A scene segmented as predicted occupied (white), unoccupied (black) and unknown (grey) for decreasing confidence thresholds (left to right) on the predicted standard deviation γ_ϕ. From most certain to most least certain, we move from high power returns labelled as occupied, to a region nearby and up to the first return, to space that lies in partial and full occlusion.

Fig. 6.

and Dan Barnes for many insightful conversations, and the reviewers for helpful feedback.

This work was supported by training grant Programme Grant EP/M019918/1. We acknowledge use of Hartree Centre resources in this work. The STFC Hartree Centre is a research collaboratory in association with IBM providing High Performance Computing platforms funded by the UKs investment in e-Infrastructure. The Centre aims to develop and demonstrate next generation software, optimised to take advantage of the move towards exa-scale computing.

REFERENCES

[1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, no. 6, pp. 46–57, 1989.
[2] K. Konolige, "Improved occupancy grids for map building," *Autonomous Robots*, vol. 4, no. 4, pp. 351–367, 1997.
[3] A. Milstein, "Occupancy grid maps for localization and mapping," in *Motion Planning*, InTech, 2008.
[4] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots:: I. a review of localization strategies," *Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.
[5] J.-A. Meyer and D. Filliat, "Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies," *Cognitive Systems Research*, vol. 4, no. 4, pp. 283–317, 2003.
[6] B. Clarke, S. Worrall, G. Brooker, and E. Nebot, "Towards mapping of dynamic environments with fmcw radar," in *Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE*, pp. 140–145, IEEE, 2013.
[7] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems*, pp. 5580–5590, 2017.
[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
[9] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations," in *Microwaves for Intelligent Mobility (ICMIM), 2015 IEEE MTT-S International Conference on*, pp. 1–4, IEEE, 2015.
[10] R. Dia, J. Mottin, T. Rakotovao, D. Puschini, and S. Lesecq, "Evaluation of occupancy grid resolution through a novel approach for inverse sensor modeling," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13841–13847, 2017.
[11] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.
[12] M. Skolnik, *Radar Handbook, Third Edition*. Electronics electrical engineering, McGraw-Hill Education, 2008.
[13] S. B. Thrun, "Exploration and model building in mobile robot domains," in *Neural Networks, 1993., IEEE International Conference on*, pp. 175–180, IEEE, 1993.
[14] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.
[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[16] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, pp. 3483–3491, 2015.
[17] H. R. Roth, C. Shen, H. Oda, M. Oda, Y. Hayashi, K. Misawa, and K. Mori, "Deep learning and its application to medical image segmentation," *Medical Imaging Technology*, vol. 36, no. 2, pp. 63–71, 2018.
[18] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artificial Intelligence Review*, pp. 1–18, 2018.
[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
[20] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar transformer networks," *arXiv preprint arXiv:1709.01889*, 2017.
[21] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, no. 4, p. 049901, 2007.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

4

There And Back Again: Learning to Simulate Radar Data for Real World Applications

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

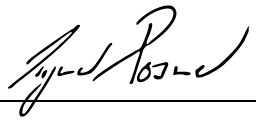
Title of Paper	There and Back Again: Learning to Simulate Radar Data for Real-World Application
Publication Status	X Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	R. Weston, O. Parker Jones, I. Posner. "There and Back Again: Learning to Simulate Radar Data for Real-World Application". In: *International Conference On Robotics and Automation 2021 (ICRA Xi'an 2021)*

Student Confirmation

Student Name:	Robert Weston		
Contribution to the Paper	Developed Approach, Ran Experiments, Wrote Manuscript		
Signature 		Date	7 / 04 / 2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Prof. Ingmar Posner		
Supervisor comments I confirm that the candidate made the contributions specified above.			
Signature 		Date	7/4/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

arXiv:2011.14389v1 [cs.RO] 29 Nov 2020

There and Back Again: Learning to Simulate Radar Data for Real-World Applications

Rob Weston, Oiwi Parker Jones and Ingmar Posner*

***Abstract*— Simulating realistic radar data has the potential to significantly accelerate the development of data-driven approaches to radar processing. However, it is fraught with difficulty due to the notoriously complex image formation process. Here we propose to learn a radar sensor model capable of synthesising faithful radar observations based on simulated elevation maps. In particular, we adopt an adversarial approach to learning a *forward* sensor model from unaligned radar examples. In addition, modelling the *backward* model encourages the output to remain aligned to the world state through a cyclical consistency criterion. The backward model is further constrained to predict elevation maps from real radar data that are grounded by partial measurements obtained from corresponding lidar scans. Both models are trained in a joint optimisation. We demonstrate the efficacy of our approach by evaluating a down-stream segmentation model trained purely on simulated data in a real-world deployment. This achieves performance within four percentage points of the same model trained entirely on real data.**

I. Introduction

The long sensing range of radar and its resilience to adverse environmental conditions make it an attractive complement to lidar and vision for robotics and autonomous driving applications. However, radar is notoriously challenging to interpret; multi-path phenomena, limited resolution, and pernicious noise artefacts arising throughout the complex and imperfect measurement pipeline pose significant challenges to radar-based perception systems. In recent years, data-driven approaches have made significant strides to overcome these challenges across a range of tasks in robotics [1], [2], [3], [4]. Central to the continuing success of such approaches is the quality, scale, and labelling of radar datasets, which in contrast to vision and lidar remain limited.

Akin to progress in the use of other sensing modalities, simulation has the potential to significantly accelerate the development and deployment of radar-based methods by reducing the need for human annotation and automating the data-gathering process. The importance of learning from simulated data can be seen across a wide range of tasks in vision and lidar [5], [6], [7], [8], [9], [10]; and it is echoed in the rapid development of multiple autonomous driving simulators capable of simulating complex worlds, designed to facilitate these approaches [11], [12], [13].

Inspired by the impact simulation has brought to the development of sophisticated vision and lidar systems, the overarching goal of our work is to enable the training of data-driven models for radar interpretation in simulation. To successfully train models in simulation, we therefore consider learning a radar sensor model that is able to faithfully simulate radar observations, such that the domain gap experienced by a down-stream system when trained on either real or synthetic data is minimal. In particular, we aim to interface with existing simulators already capable of synthesising complex scenes and adopted widely by the community. In meeting this requirement, we consider simulating radar observations given a layout of the world, supplied to our model in the form of an elevation map. Noise processes arising throughout the radar sensing pipeline give radar an inherently stochastic nature; to accurately replicate this, we adopt a probabilistic approach, using a *deep implicit model* [14] to capture a distribution over possible radar sensor measurements.

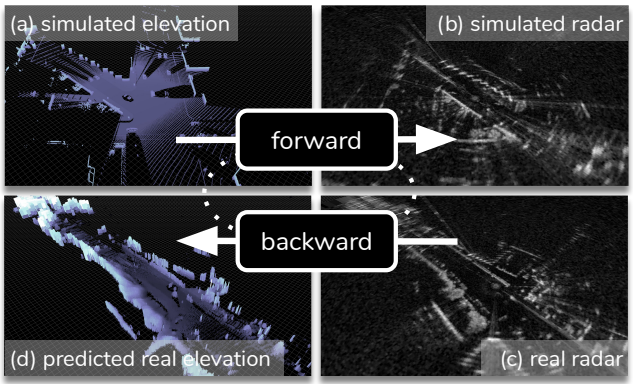


Fig. 1: Given a simulated elevation map (a) we are able to generate realistic radar in simulation (b) through a data driven approach. We achieve this by learning from unaligned real radar observations (c). Alongside the forward mapping we also learn the backward mapping from real radar (c) to predict the real world elevation (d). This allows us to further constrain training through cyclical consistency and by learning from partial lidar measurements collected in the real world.

Adopting an adversarial paradigm, we train our sensor model to explicitly generate realistic radar observations from *simulated* height maps using unaligned real-world radar data. As shown in Figure 1, alongside the forward sensor model we also learn the backward model to infer the elevation state of the real world from radar. This allows us to further constrain training by enforcing cyclical consistency [15] between the forward and backward processes. Alongside this we ground the predictions of our backward model in the real world by enforcing alignment with partial height measurements generated automatically from lidar.

Through our approach, we demonstrate the feasibility of training radar models in simulation to the best of our knowledge for the first time. In doing so, our approach achieves performance within 4% compared to that of the same model trained in the real world.

*Applied Artificial Intelligence Lab (A2I), University of Oxford {robw, oiwi, ingmar}@robots.ox.ac.uk

II. Related Work

Whilst unexplored in radar, the benefits of training in simulation have been extensively studied in both vision [10], [9], [8], [7], [6], [16] and lidar [5] across a range of perception tasks including segmentation [10], [8], detection [5], tracking [10], and optical flow [6]. In addition to training models, simulation has an important role to play in the analysis and interrogation of corner-cases, potentially too dangerous to recreate in the real world. The need for simulation, in combination with the development of powerful open source games engines [17], [18], has led to a profusion of both open source [12], [13], [11] and proprietary [17] autonomous driving simulators for vision and lidar but support for radar still remains limited. Whilst [11] provides a simple simulation of target lists as might be returned by a typical automotive radar, in this work we are concerned with simulating significantly richer—but more challenging to replicate—raw radar power measurements.

To capitalise on the similar successes that training in simulation has brought to vision and lidar, we aim to learn a radar sensor model capable of faithfully simulating the radar sensing process. But radar's complex sensing pipeline makes simulation challenging. The high frequency of radar renders direct solution of Maxwell's equations intractable. Instead, asymptotic solutions have been widely adopted, in an attempt to simulate radar, relying on a combination of geometric [19] and physical optics [20], [21], [22]. Representing objects as their characteristic scattering centers [23], [24], [25] can further help reduce computation. Whilst several approaches have demonstrated the feasibility of simulating simple driving scenes using these methods [26], [27], [28], [29], they scale poorly, relying on a precise model of the world, including material properties that are difficult to model in practice. In contrast, our approach is capable of simulating radar observations given only a simulated elevation map; this allows us to interface with and scale to the complex worlds provided in modern simulation environments, without requiring us to build a radar-specific simulation world from the ground up.

Several approaches also propose to account for noise artefacts arising through the downstream measurement process using hand-crafted phenomenological models [30], [31], [32]. More recently, data-driven approaches [33] have been proposed as a better alternative [34], learning to characterise the entire radar process, from world state to sensor observation, from raw data. Whilst [33] shows the feasibility of using a data-driven approach to replicate radar data in a controlled airfield environment with only a handful of targets, in our work we consider simulating radar in complex urban environments, where labelling exact real-world layouts is significantly more challenging. Instead of requiring exact real world layouts to train our model, as in [33], we generate radar observations from simulated height maps. As well as side-stepping the need for precise layouts of the real world, through this approach we are able to explicitly encourage our model to generate feasible radar observations *in simulation*.

Similar in flavour, therefore, to recent approaches proposed for unaligned domain transfer in vision [15], [16], [35], [36], we too consider learning *unaligned* mappings between simulated world layouts and real radar observations. We model the forward and backward model side-by-side using adversarial and cyclical consistency losses. Unlike in [15], [16], [35], [36] which consider a deterministic one-to-one mapping between domains, we adopt an inherently probabilistic approach. We capture a *distribution* over possible power returns to account for the stochastic noise processes arising throughout the radar sensing pipeline. We also further encourage our backward model to predict elevation states that are aligned to real-world measurements generated automatically in lidar, where they are available.

III. Deep Radar Simulation

A. Problem Formulation

Let $\mathbf{x}^*$ denote a real radar observation generated from real world state $\mathbf{w}^* \sim p(\mathbf{w}^*)$ as

$$\mathbf{x}^* \sim p(\mathbf{x}^*|\mathbf{w}^*), \quad \mathbf{w}^* \sim p(\mathbf{w}^*) \tag{1}$$

where $p(\mathbf{x}^*|\mathbf{w}^*)$ is the real radar sensor model. Specifically, we consider radar observations of the form $\mathbf{x}^* \in \mathbb{R}^{\Phi \times R}$ where $\mathbf{x}^*_{ij}$ gives the power returned from the world at polar-coordinate $(\phi_i, \mathrm{r}_j) \in \{\phi_i\}_{i=1}^{\Phi} \times \{\mathrm{r}_j\}_{j=1}^{R}$.

We aim to learn a radar sensor model $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ capable of generating feasible radar observations $\mathbf{x}$ from a *simulated* state $\mathbf{w} \sim p(\mathbf{w})$. We model the world state $\mathbf{w} \in \mathbb{R}^{\Phi \times \mathrm{R}}$, with the same dimensions as $\mathbf{x}$, where $\mathbf{w}_{i,j}$ gives the elevation of the world location (ϕ_i, r_i). In addition to containing the necessary information to generate $\mathbf{x}$ (such as the shape, size, and position of objects in the scene), this representation allows us to easily interface with a preexisting simulation environment, such as CARLA [11]. Whilst $\mathbf{w}$ is easily simulated it is more challenging to measure in the real world. We assume that we are able to obtain *partial* measurements $\mathbf{m}^* = m(\mathbf{w}^*)$ using lidar, where due to the limited range and number of beams of lidar many locations (ϕ_i, r_i) are without an elevation measurement. We therefore have at our disposal real world observations $\mathtt{R} = \{(\mathbf{x}^*, \mathbf{m}^*)_n\}_{n=1}^{N}$ and simulated observations $\mathtt{S} = \{\mathbf{w}_l\}_{l=1}^{L}$ with which to train our approach.

B. Stochastic Simulation Using Deep Implicit Models

For the same world state $\mathbf{w}^*$, aleatoric sensor noise results in radar sensor measurements $\mathbf{x}^*_t \sim p(\mathbf{x}^*|\mathbf{w}^*)$ and $\mathbf{x}^*_{t+1} \sim p(\mathbf{x}^*|\mathbf{w}^*)$ that are inherently stochastic. To capture the stochasticity in the mapping from $\mathbf{w}$ to $\mathbf{x}$ – mimicking the true sensing process – we introduce a latent variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; using a neural network $g_{\theta_\mathrm{x}}(\mathbf{w}, \boldsymbol{\epsilon})$ with parameters θ_x, for a fixed $\mathbf{w}$, we are able to sample multiple possible $\mathbf{x}$ by sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. This allows us to implicitly capture a distribution $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ over possible radar observations sampling $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ as

$$\mathbf{x} = g_{\theta_\mathrm{x}}(\mathbf{w}, \boldsymbol{\epsilon}; \theta_\mathrm{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p_\epsilon(\boldsymbol{\epsilon}) \ . \tag{2}$$

Through this approach we are able to leverage the representational power of a deep neural network g_{θ_x} to learn the complex mapping from world state $\mathbf{w}$ to radar sensor observation $\mathbf{x}$, whilst simultaneously implicitly learning how to characterise the uncertainty in the sensing process.

C. Learning only from Real Observations

If we could observe real-world observation-state pairs $\mathtt{R} = \{(\mathbf{x}^*, \mathbf{w}^*)_n\}_{n=1}^N$ we might train $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ by minimising

$$\mathcal{A}_\mathrm{x} = \mathbb{E}_{p(\mathbf{x}^*,\mathbf{w}^*)}[\ell(\theta_\mathrm{x}; \mathbf{x}^*, \mathbf{w}^*)] \approx \tfrac{1}{N}\textstyle\sum_n \ell(\theta_\mathrm{x}; \mathbf{x}_n^*, \mathbf{w}_n^*) \quad (3)$$

where $\ell(\theta_\mathrm{x}; \mathbf{x}^*, \mathbf{w}^*)$ is a regression loss (eg. MAE or MSE) between the real $\mathbf{x}^*$ and simulated observations $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w}^*)$. However, only partial partial measurements of the world state $\mathbf{m}^* = m(\mathbf{w}^*)$ are available. In addition, by training our model in this way, $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ is only trained on real state observations $\mathbf{w}^* \in \mathtt{R}$ and a significant domain gap between $\mathbf{w} \in \mathtt{S}$ and $\mathbf{w}^* \in \mathtt{R}$ is still likely to persist. We posit that models that explicitly incorporate simulated state observations $\mathbf{w} \in \mathtt{S}$ into the training loop are more likely to generate feasible radar observations in simulation. We now propose how this might be achieved.

D. Learning from Unaligned Real and Simulated Data

We assume that measurements of the world state $\mathbf{m}^*$ are entirely unavailable. In this case, we have the datasets $\mathtt{R} = \{\mathbf{x}_n^*\}_{n=1}^N$ and $\mathtt{S} = \{\mathbf{w}_n\}_{n=1}^N$ available to us. As $(\mathbf{x}_n^*, \mathbf{w}_n^*) \in \mathtt{R} \times \mathtt{S}$ are no longer aligned, training $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ as in (3) is no longer feasible.

To train $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to replicate $\mathbf{x}^* \sim p(\mathbf{x}^*|\mathbf{w}^*)$ using only unaligned examples $(\mathbf{x}^*, \mathbf{w})$ we adopt an adversarial approach [37]; introducing a discriminator network $d_{\beta_\mathrm{x}}(\mathbf{x})$ with parameters β_x, and training objectives,

$$\mathcal{D}_\mathrm{x} = \mathbb{E}_{p(\mathbf{x}^*)}\left[\left(d_{\beta_\mathrm{x}}(\mathbf{x}^*) - 1\right)^2\right] + \mathbb{E}_{p_{\theta_\mathrm{x}}(\mathbf{x})}\left[d_{\beta_\mathrm{x}}(\mathbf{x})^2\right] \quad (4)$$

$$\mathcal{G}_\mathrm{x} = \mathbb{E}_{p_{\theta_\mathrm{x}}(\mathbf{x})}\left[\left(d_{\beta_\mathrm{x}}(\mathbf{x}) - 1\right)^2\right] \quad (5)$$

minimising $\mathcal{D}_\mathrm{x}(\beta_\mathrm{x})$ with respect to β_x and $\mathcal{G}_\mathrm{x}(\theta_\mathrm{x})$ with respect to θ_x.[1] Here we have adopted a least-squares loss assuming a $[\mathbf{x}, \mathbf{x}^*] = [0, 1]$ coding scheme: this avoids discriminator saturation which can destabilise training [38].

In reality the expectations in (4) and (5) are estimated using $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] \approx \frac{1}{K}\sum_k f(\mathbf{x}_k)$ with $\mathbf{x}_k \sim p(\mathbf{x})$; crucially this allows us to train $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ using only unaligned samples $(\mathbf{x}^*, \mathbf{w}) \in \mathtt{R} \times \mathtt{S}$ sampling from $p_{\theta_\mathrm{x}}(\mathbf{x})$ as $\mathbf{x}_k \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w}_k)$ using (2).

E. Constraining Training Using Cyclical Consistency

However, training $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ by minimising just $\mathcal{G}_\mathrm{x}(\theta_\mathrm{x})$ with unaligned examples is a highly unconstrained process [15]. In the worst case, $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ could choose to disregard the state information $\mathbf{w}$ entirely, instead generating $\mathbf{x} = g_{\theta_\mathrm{x}}(\mathbf{w}, \boldsymbol{\epsilon})$ using only $\boldsymbol{\epsilon}$ as in a standard GAN formulation [37].

To counter this, in order to further constrain $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$, we also model the backward mapping $\mathbf{w}^* \sim p(\mathbf{w}^*|\mathbf{x}^*)$ as $\mathbf{w} = g_{\theta_\mathrm{w}}(\mathbf{x}, \boldsymbol{\kappa}; \theta_\mathrm{w})$ with $\boldsymbol{\kappa} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where g_{θ_w} is a neural network with parameters θ_w and $\boldsymbol{\kappa}$ induces the uncertainty in $p(\mathbf{w}^*|\mathbf{x}^*)$. Crucially, this allows us to impose additional cyclical consistency constraints [15], [16], [35],

$$\mathbf{w}'' \approx \mathbf{w} \quad \mathbf{w}'' \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}') \quad \mathbf{x}' \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w}) \quad (6)$$

$$\mathbf{x}'' \approx \mathbf{x}^* \quad \mathbf{x}'' \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w}') \quad \mathbf{w}' \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*) \quad (7)$$

into our training framework, explicitly encouraging $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to use $\mathbf{w}$ by enforcing (6) and $p_{\theta_\mathrm{w}}(\mathbf{w}|\mathbf{x})$ to use $\mathbf{x}$ by enforcing (7) through cyclical consistency losses,

$$\mathcal{C}_\mathrm{w}(\theta_\mathrm{x}, \theta_\mathrm{w}) = \mathbb{E}_{p(\mathbf{w})}\left[\|\mathbf{w} - \mathbf{w}''\|_1\right] \quad (8)$$

$$\mathcal{C}_\mathrm{x}(\theta_\mathrm{x}, \theta_\mathrm{w}) = \mathbb{E}_{p(\mathbf{x}^*)}\left[\|\mathbf{x}^* - \mathbf{x}''\|_1\right] . \quad (9)$$

Alongside (8) and (9), we also train $p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*)$ using an adversarial objective. Introducing another discriminator network $d_{\beta_\mathrm{w}}(\mathbf{w})$ with parameters β_w, and training objectives,

$$\mathcal{D}_\mathrm{w} = \mathbb{E}_{p(\mathbf{w})}\left[\left(d_{\beta_\mathrm{w}}(\mathbf{w}) - 1\right)^2\right] + \mathbb{E}_{p_{\theta_\mathrm{w}}(\mathbf{w}^*)}\left[d_{\beta_\mathrm{w}}(\mathbf{w}^*)^2\right] \quad (10)$$

$$\mathcal{G}_\mathrm{w} = \mathbb{E}_{p_{\theta_\mathrm{w}}(\mathbf{w}^*)}\left[\left(d_{\beta_\mathrm{w}}(\mathbf{w}^*) - 1\right)^2\right] \quad (11)$$

we minimise $\mathcal{D}_\mathrm{w}(\beta_\mathrm{w})$ with respect to β_w and $\mathcal{G}_\mathrm{w}(\theta_\mathrm{w})$ with respect to θ_w, generating samples $\mathbf{w}_k^* \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*)$ as $\mathbf{w}_k^* \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}_k^*)$ using (10) with $\mathbf{x}_k^* \in \mathtt{R} \sim p(\mathbf{x}^*)$.

We note that just as $\mathcal{G}_\mathrm{x}(\theta_\mathrm{x})$ could lead $g_{\theta_\mathrm{x}}(\mathbf{w}, \boldsymbol{\epsilon})$ to ignore $\mathbf{w}$ (as discussed previously), in the worst case $\mathcal{C}_\mathrm{x}(\theta_\mathrm{x}, \theta_\mathrm{w})$ encourages $\mathbf{x} = g_{\theta_\mathrm{x}}(\mathbf{w}, \boldsymbol{\epsilon})$ to ignore $\boldsymbol{\epsilon}$ enforcing a one-to-one mapping between $\mathbf{x}$ and $\mathbf{w}$ [39]. Whilst several extensions have been proposed to overcome this problem [40], [39], in reality we find that this does not occur in our training setup; we posit that the need to generate realistic radar observations that are capable of tricking the discriminator $\mathcal{G}_\mathrm{x}(\theta_\mathrm{x})$ far outweighs $\mathcal{C}_\mathrm{w}(\theta_\mathrm{x}, \theta_\mathrm{w})$, avoiding degeneracy.

F. Learning from Partial Lidar Measurements

Another benefit of learning the backward model $p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*)$ is that it allows us to learn from partial measurements $\mathbf{w}^* \sim p(\mathbf{w}^*)$ when they are available. This is achieved through an alignment consistency objective,

$$\mathcal{A}_\mathrm{w}(\theta_\mathrm{w}) = \mathbb{E}_{p(\mathbf{x}^*,\mathbf{w}^*)}[\ell(\theta_\mathrm{w}; \mathbf{w}^*, \mathbf{x}^*)] \quad (12)$$

where $\ell(\theta_\mathrm{w}; \mathbf{w}^*, \mathbf{x}^*) = \|(\mathbf{m}^* - g_{\theta_\mathrm{w}}(\mathbf{w}^*, \boldsymbol{\kappa})) \odot \mathbb{I}(\mathbf{m}^*)\|_1$, and $p(\mathbf{x}^*, \mathbf{w}^*) = p(\mathbf{x}^*|\mathbf{w}^*)p(\mathbf{w}^*)$ is the joint distribution over real observation-state pairs with $\mathbb{I}(\cdot)$ an element-wise indicator function returning 1 if the measurement of $\mathbf{m}_{i,j}^*$ exists or 0 otherwise.

Considering a combined training objective,

$$\mathcal{L}(\theta_\mathrm{x}, \theta_\mathrm{w}) = \mathcal{G}_\mathrm{x} + \lambda_\mathrm{gw}\mathcal{G}_\mathrm{w} + \lambda_\mathrm{cx}\mathcal{C}_\mathrm{x} + \lambda_\mathrm{cw}\mathcal{C}_\mathrm{w} + \lambda_\mathrm{aw}\mathcal{A}_\mathrm{w} \quad (13)$$

with hyper-parameters $\boldsymbol{\lambda} = [\lambda_\mathrm{gw}, \lambda_\mathrm{cx}, \lambda_\mathrm{cw}, \lambda_\mathrm{aw}]$ used to trade off the relative importance of each term, we are able to train *both* $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ and $p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*)$ – explicitly encouraging $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to generate realistic radar observations in simulation, training on $\mathbf{w} \in \mathtt{S}$, whilst also learning from aligned pairs $\mathbf{x}, \mathbf{m}^* \in \mathtt{R}$ when measurements $\mathbf{m}^*$ are available.

[1] More generally the expectations in (4) and (5) could be written with respect to the joint $p(\mathbf{x}, \mathbf{w})$ but as the terms inside the expectations are only functions of $f(\mathbf{x})$ we have $\mathbb{E}_{p(\mathbf{x},\mathbf{w})}[f(\mathbf{x})] = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})]$ which we adopt for simplicity. Here we consider generating samples $\mathbf{x} \sim p(\mathbf{x})$ by sampling from the joint distribution $\mathbf{x}, \mathbf{w} \sim p(\mathbf{x}, \mathbf{w})$ and disregarding $\mathbf{w}$

IV. Experimental Setup

A. Self-Supervised Dataset Generation

In section III-A we assumed that we had aligned real-world radar observations and partial state measurements $\mathtt{R} = \{(\mathbf{x}^*, \mathbf{m}^*)_n\}_{n=1}^N$ and unaligned but perfectly observed state observations generated in simulation $\mathtt{S} = \{\mathbf{w}_l\}_{l=1}^L$ with which to train our model. We now describe how to attain $(\mathtt{R}, \mathtt{S})$ in practice.

1) Generating $\mathtt{R}$*:* We generate $\mathtt{R}$ from the Oxford Radar RobotCar Dataset [41], [42]. We partition the dataset into train and test sets with the training set being composed of 29 10km loops (generating 222420 observations) with 3 loops being reserved for testing (23460 observations), resulting in an approximate $90 : 10$ split. Each $\mathbf{x}^*$ corresponds to the output of a Navtech CTS350x FMCW radar rotating about its vertical axis, down-sampled to a 0.35m resolution and scaled to give $\mathbf{x}^* \in [-1, 1]^{400\times 471}$. This corresponds to a 360° field-of-view with maximum observable height 5.2m. We construct partial height map measurements $\mathbf{m}^*$ by combining the output of two HDL32E Velodyne Lidars: as a result of differing sensing frequencies (radar at 4Hz and lidar at 20Hz) each $\mathbf{x}^*$ is matched to multiple lidar scans to maintain accurate labelling. The lidar pointclouds are filtered to coincide with the radar's horizontal field-of-view $(-40^o, 1.8^o)$, before being binned onto a polar grid and labelling each grid cell with the maximum height of any point falling within it. Each $\mathbf{m}^*$ is then scaled from the interval $[-2.2\text{m}, 5.2\text{m}]$ to the interval $[-1, 1]$ with any grid cell without a label assigned the value -1. Due to the limited number of lasers and range of each lidar, each $\mathbf{m}^*$ is only a partial measurement $\mathbf{m}^* = m(\mathbf{w}^*)$ of the true world state, with many cells having no observation attached to them.

2) Generating $\mathtt{S}$*:* We generate $\mathtt{S} = \{\mathbf{w}_l\}_{l=1}^L$ utilising the CARLA simulator [11] by mounting a data collection vehicle with four orthogonal depth cameras at the same height as the radar used to generate $\mathbf{w}$ (1.97m above the ground plane), each producing a $\mathbb{R}^{1024\times 1024}$ image. These are projected into a dense 3D pointcloud which is then converted to height labels $\mathbf{w} \in [-1, 1]^{400\times 471}$ in a similar approach described in the previous section. In contrast to the the real-world elevation maps generated by lidar we assume that the simulated state observations $\mathbf{w}$ are dense with each cell potentially observable in radar having a height measurement attached to it. The simulation world was spawned with 200 vehicles of random types and size and 300 pedestrians, using town layouts 1 and 2. Observations were collected by setting the vehicle into auto-pilot mode and recorded only when the vehicle was moving. The simulation was restarted after 60 seconds of no movement. Through this approach we generated 10^5 observations for training. A further $68,400$ were held out for testing.

B. Network Architectures and Training

Our network architecture and training set-up largely follow that proposed in [15]. Specifically, we use ResNet generators for g_{θ_x} and g_{θ_w} [43], with 2-strided convolution, 9 residual blocks, and 2 up-convolutions before a final $\tanh$ activation. After each convolution, batch normalisation [44] is applied before a ReLU activation. The variables $\boldsymbol{\epsilon}$ and $\boldsymbol{\kappa}$ are sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ before concatenation with $\mathbf{w}$ and $\mathbf{x}$ respectively, and passed to g_{θ_x} and g_{θ_w} as a 2-channel polar tensor. We utilize patch discriminators for d_{β_x} and d_{β_w} [45], sampling generated observations from a pool of 50 when training as in [15]. All networks are implemented in PyTorch [46] and trained for 5×10^5 steps using the Adam optimizer [47] with learning rate 2×10^{-4}, $\beta = (0.5, 0.999)$, and a batch size of 1. In all experiments we set $\boldsymbol{\lambda} = [1, 10, 10, 10]$ as given in (13).

C. Evaluation

1) Radar Simulation: One of the central motivations for our approach was to develop a radar sensor model $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to train new models $y_\alpha(\mathbf{x})$ in simulation that generalise to the real world. With this in mind, to assess the realism of radar observations $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$, we train a model $y_\alpha(\mathbf{x})$ in simulation minimising $\mathbb{E}_{p_{\theta_\mathrm{x}}(\mathbf{x},\mathbf{y})}[\ell(\mathbf{y}, y_\alpha(\mathbf{x}))]$ with respect to α where $\ell(\cdot,\cdot)$ is a loss between the actual and predicted target. We then assess the realism of $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ as, $\mathbb{E}_{p(\mathbf{x}^*,\mathbf{y}^*)}\left[m(\mathbf{y}^*, y_\alpha(\mathbf{x}^*|\theta_\mathrm{x}))\right]$ evaluating the performance of the trained model $y_\alpha(\mathbf{x}^*|\theta_\mathrm{x})$ in the real world with metric $m(\cdot,\cdot)$. Specifically, we consider training a model to segment the world into occupied, free, and unknown space. To successfully train $y_\alpha(\mathbf{x})$ this task requires $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to replicate realistic noise artefacts (which $y_\alpha(\mathbf{x})$ must learn to overcome) whilst ensuring the mapping from world state $\mathbf{w}$ to radar observation $\mathbf{x}$ is as faithful as possible to the real-world mapping $\mathbf{w}^*$ to $\mathbf{x}^*$.

We train and evaluate the segmentation model using datasets $\mathtt{S}' = \{(\mathbf{x}, \mathbf{y})\}_{n=1}^N$ and $\mathtt{R}' = \{(\mathbf{x}^*, \mathbf{y}^*)\}_{n=1}^N$, generated from the sets held out in sections IV-A.1 and IV-A.2. In both cases labels are generated automatically from either partial $\mathbf{m}^*$ or full $\mathbf{w}$ measurements of the world state in a similar approach to [1]: after extracting the ground plane, any cell with a height measurement is labelled as occupied, all cells before the first return are labelled as free, and anything else is labelled unknown. For all segmentation networks y_α, a U-Net architecture [48] is used with 6 levels, doubling the features and halving spatial resolution at each level, starting with 8 features at the input and allowing information to flow from encoder to decoder using skip connections. Each model is trained with a batch size of 8 using the Adam [47] optimizer (learning rate 1×10^{-3}) to minimise the cross-entropy loss, with an additional weighting of 50 applied to the occupied class to account for class imbalance. The model attaining the highest IoU over 4 epochs of training (tested on a 10% hold out dataset after each epoch) is used for evaluation – usually the first or second.

In keeping with [49] and as proposed in the Pascal Voc Challenge [50] each model is evaluated using the mean Intersection over Union metric [51] (mIoU)

$$\mathcal{M}(\theta_\mathrm{x}) = \frac{1}{2}\sum_c \left[\frac{\mathrm{TP}(c)}{\mathrm{FP}(c) + \mathrm{FN}(c) + \mathrm{TP}(c)}\right] \tag{14}$$

where the true positives TP, false positives FP and false negatives FN are determined for each class (occupied and free) comparing $\mathbf{y}^*_{i,j}$ and $y_\alpha(\mathbf{x})_{i,j}$ at each index (i, j) across the entire dataset $\mathtt{R}' = \{(\mathbf{x}^*, \mathbf{y}^*)\}_{n=1}^N$. Any cell that is predicted unknown but labelled as free or occupied is counted as a false negative.

	trained on			training objective						intersection over union		
				$\mathcal{A}_\mathrm{x}$	$\mathcal{A}_\mathrm{w}$	$\mathcal{G}_\mathrm{x}$	$\mathcal{G}_\mathrm{w}$	$\mathcal{C}_\mathrm{x}$	$\mathcal{C}_\mathrm{w}$	free	occ	mean
benchmark												
real world	-	-	-	-	-	-	-	-	-	0.856	0.553	0.705
ours												
(a)	$\mathbf{x}^*$	$\mathbf{m}^*$	-	✓	-	-	-	-	-	0.396 (0.00)	0.275 (0.00)	0.335 (0.00)
(b)	$\mathbf{x}^*$	$\mathbf{m}^*$	-	✓	-	✓	-	-	-	0.558 (0.11)	0.221 (0.03)	0.389 (0.07)
(c)	$\mathbf{x}^*$	-	$\mathbf{w}$	-	-	✓	-	-	-	0.385 (0.07)	0.148 (0.01)	0.266 (0.04)
(d)	$\mathbf{x}^*$	-	$\mathbf{w}$	-	-	✓	✓	✓	✓	0.845 (0.02)	0.262 (0.02)	0.553 (0.02)
(e)	$\mathbf{x}^*$	$\mathbf{m}^*$	$\mathbf{w}$	-	✓	✓	✓	✓	✓	**0.872** (0.01)	**0.455** (0.01)	**0.664** (0.01)

TABLE I: Radar simulation performance: real-world mIoU performance for segmentation models trained by different radar sensor models in simulation as outlined in Section III and discussed in Section V-A (averaged over four random seeds presented with standard deviations)

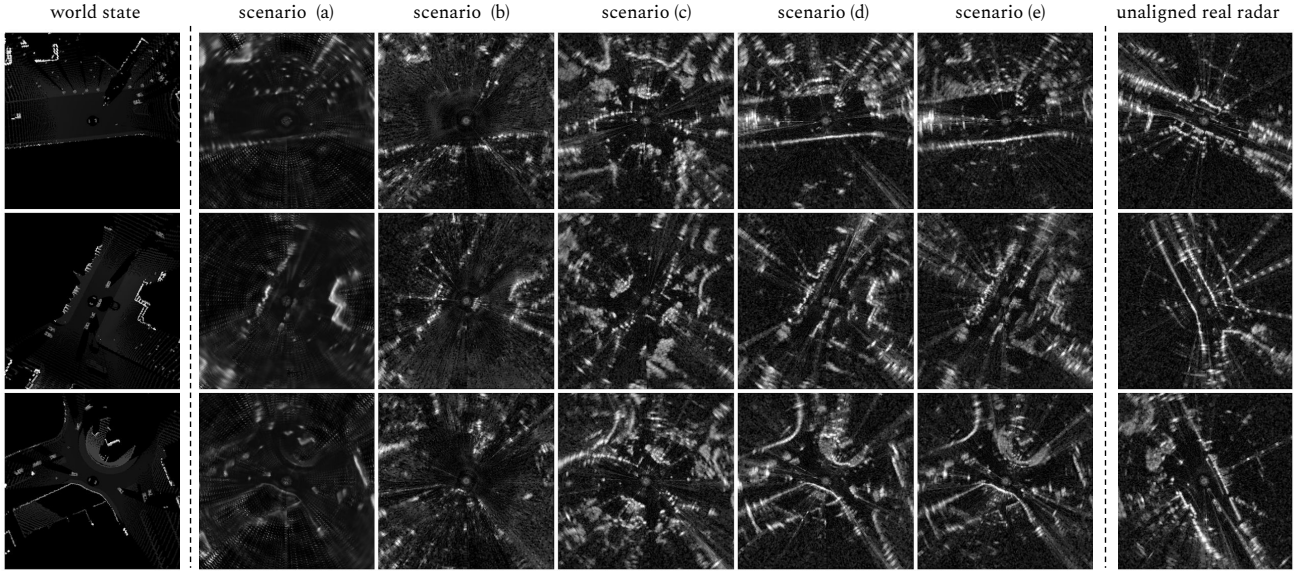


Fig. 2: Simulated radar for training scenarios given in Table I generated from elevation maps given in the first column. *Unaligned* real radar observations $\mathbf{x}^*$ are also shown for reference (last column). Simulators trained using only real data, as in (a) and (b), fail to synthesise realistic radar in simulation. Whilst at first glance simulators trained using only an adversarial criterion between simulated and real radar appear realistic, on closer inspection they are poorly aligned to the world state as can be seen in (c). Enforcing cyclical consistency helps to remedy this as can be seen in (d). The most realistic radar observations correspond to simulators trained with the full training objective, as in (e), and backed up by Table I.

2) Evaluating the Backward Model: The heights predicted by our backward model $\mathbf{w}^* \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*)$ are evaluated using absolute mean error between $\mathbf{w}^* \sim p_{\theta_\mathrm{w}}(\mathbf{w}^*|\mathbf{x}^*)$ and partial state measurements $\mathbf{m}^*$ over the test set $\mathtt{R}'$ from section IV-A.1, evaluated only where measurement $\mathbf{m}^*_{i,j}$ exist. As a result of the class imbalance between height labels corresponding to the ground plane and targets in the scene, we evaluate this metric over occupied and free space independently, presenting both alongside their average. All height evaluations were run for four models trained with different random seeds, from which we computed averages and error bounds presented as standard deviations.

V. Results

A. Radar Simulation

To assess the realism of radar observations simulated through our approach $\mathbf{x} \sim p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$, we consider how models trained using $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ perform in the real world using the evaluation method proposed in section IV-C.1. The results for an ablation of possible training setups is given in Table I with a qualitative comparison given in Figure 2. In (a) and (b) we assess whether it is possible to learn to simulate realistic radar observations given a simulated elevation map $\mathbf{w}$ whilst only training on real-world measurements $(\mathbf{x}^*, \mathbf{m}^*) \in \mathtt{R}$ as proposed in section III-C. In (a) we train $p_{\theta_\mathrm{x}}(\mathbf{x}|\mathbf{w})$ to regress to $\mathbf{x}^*$ directly from partial state measurements minimising (13) whilst in (b) we add an additional adversarial loss $\mathcal{G}_\mathrm{x}$ introducing a discriminator $d_{\beta_\mathrm{x}}(\mathbf{x})$ to distinguish between real and simulated radar observations (similar to [52]). Both (a) and (b) lead to models that poorly generalise to the real world corresponding to a mIoU of 0.335 and 0.389 respectively, and as seen qualitatively in Figure 2.

In (c) and (d) we consider learning using only unaligned observations $(\mathbf{x}^*, \mathbf{w})$ assuming that partial state measurements $\mathbf{m}^*$ are unavailable. Specifically, in (c) we train a model trained using only an adversarial loss $\mathcal{G}_\mathrm{x}$ between real $\mathbf{x}^*$ and simulated radar observations $\mathbf{x} \sim \mathbf{w}^*$ as proposed in

section III-D whilst in (d) we add in cyclical consistency constraints $\mathcal{C}_{\mathrm{x}}$ and $\mathcal{C}_{\mathrm{w}}$, also learning the backward model $p_{\theta_{\mathrm{w}}}(\mathbf{w}^*|\mathbf{x}^*)$ as proposed in section III-E. We find that (c) leads to models that perform poorly in the real world in this instance achieving a mIoU of only 0.266; training only on adversarial criteria leads to simulated radar observations that are poorly aligned to the real-world state as can be seen in Figure 2. Additionally modelling the backward model and imposing cyclical consistency, as in (d), allows us to encourage alignment between the world state and simulated radar observation – boosting performance to 0.553 and significantly outperforming (a) and (b).

However, our best performing model (e) is trained using the full training objective given in (13) as proposed in section III-F. In addition to producing the most realistic simulated observations $\mathbf{x}$ in Figure 2, in this case we are able to train a model in simulation achieving a mIoU of 0.664, only 4 percentage points off the same model trained directly on real data in the same domain as the test set 0.705.

B. Height Inference

In addition to improving the realism of the radar sensor model, as demonstrated in the last section, the backward model $p_{\theta_{\mathrm{w}}}(\mathbf{w}^*|\mathbf{x}^*)$ learnt as part of the same training setup can be used to infer the underlying elevation state of the world given a real world radar observation $\mathbf{x}^*$.

We evaluate the quality of the heights predicted by $p_{\theta_{\mathrm{w}}}(\mathbf{w}^*|\mathbf{x}^*)$ using the evaluation procedure described in section IV-C.2. We use the MAE error as compared to the partial height measurements $\mathbf{m}^*$ made in lidar evaluated separately for both both free and occupied space. The results of this process are presented in Table II for several different training configurations. In (a) we consider just training $p_{\theta_{\mathrm{w}}}(\mathbf{w}^*|\mathbf{x}^*)$ by enforcing only cyclical consistency as proposed in section III-E whilst in (b) we consider training $p_{\theta_{\mathrm{w}}}(\mathbf{w}^*|\mathbf{x}^*)$ using the full training criteria given in equation (13). By imposing alignment in (b) the system is able to more accurately infer the elevation map $\mathbf{w}^*$ with MAE 23cm – compared to a 39cm accuracy for (a). Whilst (b) performs slightly worse than our benchmark of 13cm we find that as a result of the additional adversarial constraint $\mathcal{G}_{\mathrm{w}}$, we are able to generalise to regions outside of the range of lidar (unlike our benchmark) as can be seen in Figure 3.

					Mean Absolute Error (cm)		
	data	$\mathcal{A}_{\mathrm{z}}$	$\mathcal{G}_{\mathrm{w}}$	$\mathcal{C}_{\mathrm{w}}$	free	occ	mean
Benchmark							
direct regression	R	✓	-	-	18.7	7.5	13.1
Ours							
(a)	R, S	-	✓	✓	3.9 (0.4)	74.2 (1.8)	39.0 (0.9)
(b)	R, S	✓	✓	✓	4.1 (0.6)	41.1 (5.7)	22.6 (2.6)

TABLE II: MAE for the heights predicted by the backward model evaluated against partial height measurements generated in lidar as proposed in Section V-B. Our benchmark corresponds to a model trained to regress directly to lidar measurements. (Averaged over four random seeds and presented with standard deviations.)

VI. Conclusion

This work demonstrates that it is possible to develop a data-driven approach to radar-simulation capable of training downstream systems that are readily deployable in the real world. By simulating radar sensor observations from elevation maps we are able to interface with existing simulators already capable of synthesising complex real-world scenes. We adopt an inherently stochastic and data-driven approach, capturing a mapping from state to radar sensor model (alongside sensor noise). We learn our approach from real radar measurements, simulated elevation maps, and partial elevation measurements generated in lidar. To encourage our model to simulate realistic radar observations, we adopt an adversarial approach model the backward mapping to further constrain learning through cyclical consistency losses and partial alignment to real-world elevation maps.

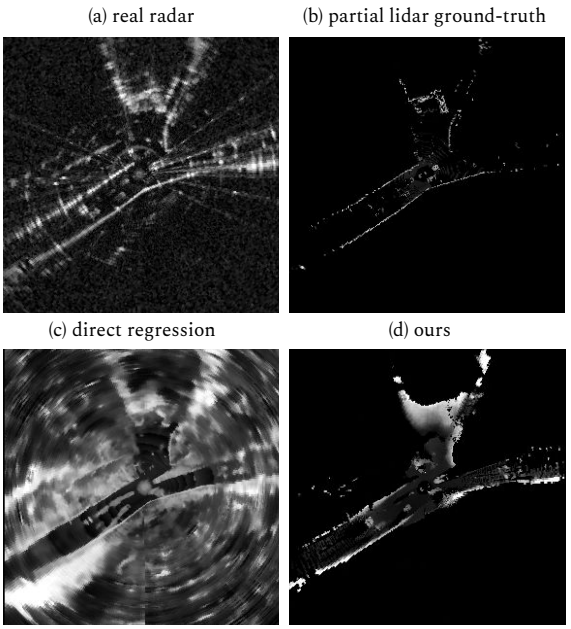


Fig. 3: Predicted elevation maps from real-world radar. As can be seen in (c) models learnt by directly regressing to partial height maps $\mathbf{m}^*$ poorly generalise to regions without labels. Adding in an additional adversarial loss forces the predicted elevation to look more like simulated height maps allowing the model in (d) to generalise to regions of space for which labels do not exist.

Using our approach to train a segmentation system in simulation, we find that when deployed in the real world, the system is able to operate with a mIoU of 0.664 performing comparably to a model trained in the real world only. To the best of our knowledge this is the first time that the feasibility of training models in simulation has been demonstrated in radar. The backward model learnt as part of the same training setup can be used to infer the height state of the world with an accuracy of 23cm, using only partial elevation measurements, whilst generalising to regions of space for which no labels exist.

Whilst our model is able to successfully train segmentation models in simulation that partition the world into occupied, free, and unknown space, early experiments (on a limited test set) found that partitioning occupied space into finer grained classes was significantly more challenging. This constitutes an interesting area for future research.

REFERENCES

[1] R. Weston, S. Cen, P. Newman, and I. Posner, "Probably unknown: Deep inverse sensor modelling radar," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5446–5452, IEEE, 2019.

[2] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," *arXiv preprint arXiv:1909.03752*, 2019.

[3] S. Saftescu, M. Gadd, D. De Martini, D. Barnes, and P. Newman, "Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Paris), 2020.

[4] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman, "RSL-Net: Localising in Satellite Images From a Radar on the Ground," *IEEE Robotics and Automation Letters*, vol. 5, pp. 1087–1094, April 2020.

[5] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, "LiDARsim: Realistic LiDAR simulation by leveraging the real world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11167–11176, 2020.

[6] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8934–8943, 2018.

[7] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4040–4048, 2016.

[8] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

[9] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, pp. 102–118, Springer, 2016.

[10] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[12] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, p. 2, 2000.

[13] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.

[14] S. Mohamed and B. Lakshminarayanan, "Learning in implicit generative models," *arXiv preprint arXiv:1610.03483*, 2016.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[16] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *arXiv preprint arXiv:1703.05192*, 2017.

[17] E. Games, "Unreal engine."

[18] J. K. Haas, "A history of the unity game engine," 2014. Worcester Polytechnic Institute.

[19] H. Ling, R.-C. Chou, and S.-W. Lee, "Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity," *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 2, pp. 194–205, 1989.

[20] F. Weinmann, "Ray tracing with po/ptd for rcs modeling of large complex objects," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 6, pp. 1797–1806, 2006.

[21] M. Domingo, F. Rivas, J. Perez, R. Torres, and M. Catedra, "Computation of the rcs of complex bodies modeled using nurbs surfaces," *IEEE Antennas and Propagation Magazine*, vol. 37, no. 6, pp. 36–47, 1995.

[22] J. M. Rius, M. Ferrando, and L. Jofre, "Greco: Graphical electromagnetic computing for rcs prediction in real time," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 2, pp. 7–17, 1993.

[23] M. Hurst and R. Mittra, "Scattering center analysis via prony's method," *IEEE Transactions on Antennas and Propagation*, vol. 35, no. 8, pp. 986–988, 1987.

[24] K. Schuler, D. Becker, and W. Wiesbeck, "Extraction of virtual scattering centers of vehicles by ray-tracing simulations," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 11, pp. 3543–3551, 2008.

[25] M. Andres, P. Feil, W. Menzel, H.-L. Bloecher, and J. Dickmann, "3D detection of automobile scattering centers using UWB radar sensors at 24/77 GHz," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 3, pp. 20–25, 2013.

[26] U. Chipengo, P. M. Krenz, and S. Carpenter, "From antenna design to high fidelity, full physics automotive radar sensor corner case simulation," *Modelling and Simulation in Engineering*, 2018.

[27] U. Chipengo and M. Commens, "A 77 ghz simulation study of roadway infrastructure radar signatures for smart roads," in *2019 16th European Radar Conference (EuRAD)*, pp. 137–140, IEEE, 2019.

[28] N. Hirsenkorn, P. Subkowski, T. Hanke, A. Schaermann, A. Rauch, R. Rasshofer, and E. Biebl, "A ray launching approach for modeling an fmcw radar system," in *2017 18th International Radar Symposium (IRS)*, pp. 1–10, IEEE, 2017.

[29] T. Machida and T. Owaki, "Rapid and precise millimeter-wave radar simulation for adas virtual assessment," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 431–436, IEEE, 2019.

[30] M. Buhren and B. Yang, "Simulation of automotive radar target lists using a novel approach of object representation," in *2006 IEEE Intelligent Vehicles Symposium*, pp. 314–319, IEEE, 2006.

[31] M. Buhren and B. Yang, "Extension of automotive radar target list simulation to consider further physical aspects," in *2007 7th International Conference on ITS Telecommunications*, pp. 1–6, IEEE, 2007.

[32] M. Bühren and B. Yang, "Simulation of automotive radar target lists considering clutter and limited resolution," in *Proc. of International Radar Symposium*, pp. 195–200, 2007.

[33] T. A. Wheeler, M. Holder, H. Winner, and M. J. Kochenderfer, "Deep stochastic radar models," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 47–53, IEEE, 2017.

[34] M. Holder, P. Rosenberger, H. Winner, T. D'hondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, *et al.*, "Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2616–2622, IEEE, 2018.

[35] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, pp. 2849–2857, 2017.

[36] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, "Surfelgan: Synthesizing realistic sensor data for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11118–11127, 2020.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[38] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[39] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville, "Augmented cyclegan: Learning many-to-many mappings from unpaired data," *arXiv preprint arXiv:1802.10151*, 2018.

[40] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Advances in neural information processing systems*, pp. 465–476, 2017.

[41] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The Oxford Radar RobotCar Dataset: A radar extension to the Oxford RobotCar Dataset," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[42] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[43] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, Springer, 2016.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[45] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, 2017.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information*

Processing Systems 32 (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241, Springer, 2015.

[49] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[50] D. Hoiem, S. K. Divvala, and J. H. Hays, "Pascal voc 2008 challenge," in *PASCAL challenge workshop in ECCV*, Citeseer, 2009.

[51] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[52] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

Architecture Diagram

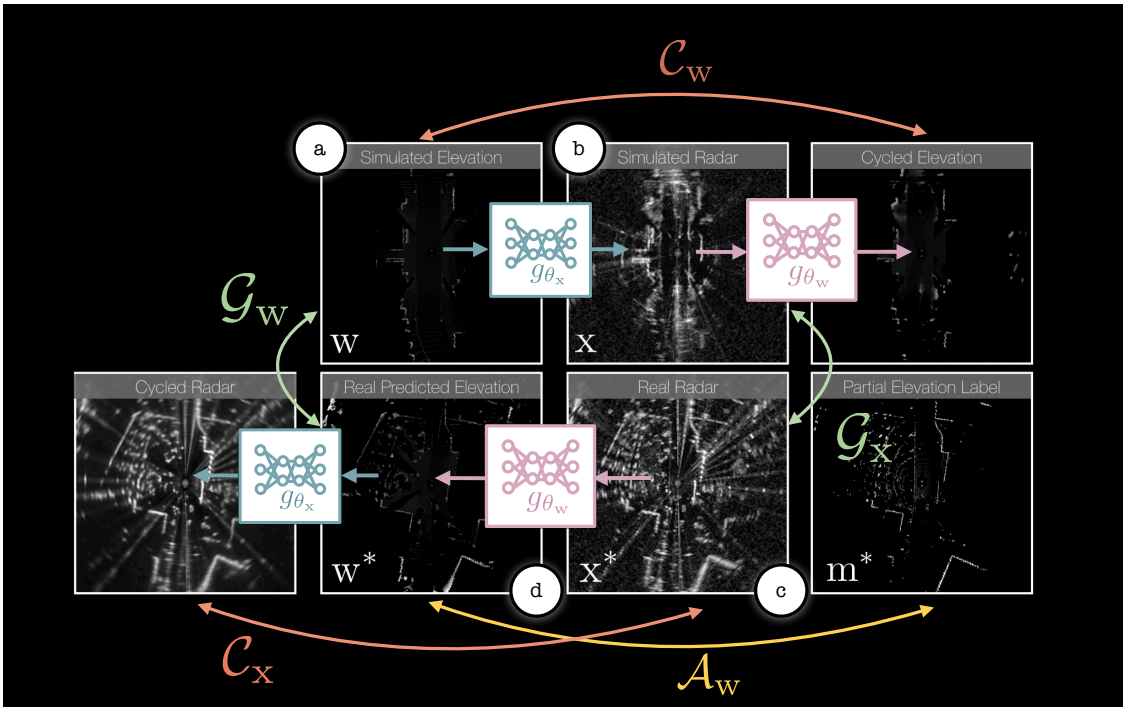


Figure 4.1 : **Network Training Setup** The forward model $\mathbf{x} = g_{\theta_{\mathrm{x}}}(\mathbf{w}, \boldsymbol{\epsilon})$ (light blue) takes as input a *simulated* elevation map ⓐ and generates a simulated radar measurement ⓑ. The backward model $\mathbf{w}^* = g_{\theta_{\mathrm{w}}}(\mathbf{x}^*, \boldsymbol{\kappa})$ (light pink) maps a *real* radar measurement ⓒ to an elevation map ⓓ. Here $\boldsymbol{\epsilon}$ and $\boldsymbol{\kappa}$ are random base variables which when sampled induce distributions over $\mathbf{x}$ and $\mathbf{w}^*$ respectively (not shown in the diagram) and $g_{\theta_{\mathrm{x}}}$ and $g_{\theta_{\mathrm{w}}}$ are neural networks. Both models are trained simultaneously through a combination of adversarial criterion, $\mathcal{G}_{\mathrm{x}}$ and $\mathcal{G}_{\mathrm{w}}$, as defined in Eq. 5 and Eq. 11, introducing discriminators $d_{\beta_{\mathrm{x}}}$ and $d_{\beta_{\mathrm{w}}}$. Cyclical consistency constraints, $\mathcal{C}_{\mathrm{x}}$ and $\mathcal{C}_{\mathrm{w}}$, as defined in Eq 8. and Eq. 9, are also introduced to ensure that the forward and backward processes are approximate inverses of one another. A partial alignment loss $\mathcal{A}_{\mathrm{w}}$ is introduced to further constrain training (as defined in Eq. 12.)

5

Masking By Moving: Learning Distraction-Free Radar Odometry from Pose Information

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

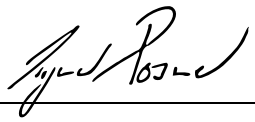
Title of Paper	Masking by Moving: Learning Distraction-Free Radar Odometry From Pose Information
Publication Status	X Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	D. Barnes, R.Weston, I. Posner. "Masking by Moving: Learning Distraction-Free radar Odometry From Pose Information". In: *Conference On Robotic Learning 2019 (CoRL Osaka 2019)*

Student Confirmation

Student Name:	Robert Weston		
Contribution to the Paper	Developed approach for quantifying the Uncertainty in pose prediction and ran experiments for uncertainty prediction in results section (section 4.4, 5.2), significant contribution to writing of manuscript, contributed ideas to the approach developed through multiple discussions.		
Signature 		Date	07 / 04 / 2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Prof. Ingmar Posner		
Supervisor comments I confirm that the candidate made the contributions specified above.			
Signature 		Date	7/4/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

arXiv:1909.03752v4 [cs.CV] 17 Jan 2020

Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information

Dan Barnes, Rob Weston, Ingmar Posner
Applied AI Lab, University of Oxford
{dbarnes, robw, ingmar}@robots.ox.ac.uk

Abstract: This paper presents an end-to-end radar odometry system which delivers robust, real-time pose estimates based on a learned embedding space free of sensing artefacts and distractor objects. The system deploys a fully differentiable, correlation-based radar matching approach. This provides the same level of interpretability as established scan-matching methods and allows for a principled derivation of uncertainty estimates. The system is trained in a (self-)supervised way using only previously obtained pose information as a training signal. Using 280km of urban driving data, we demonstrate that our approach outperforms the previous state-of-the-art in radar odometry by reducing errors by up 68% whilst running an order of magnitude faster.

Keywords: Perception, Radar, Odometry, Localisation, Deep Learning, Autonomous Driving

1 Introduction

Robust ego-motion estimation and localisation are established cornerstones of autonomy. Emerging commercial needs as well as otherwise ambitious deployment scenarios require our robots to operate in ever more complex, unstructured environments and in conditions distinctly unfavourable for typical go-to sensors such as vision and lidar. Our robots now need to see further, through fog, rain and snow, despite lens flare or when directly facing the sun. Radar holds the promise of remedying many of these shortcomings. However, it is also a notoriously challenging sensing modality: radar applications are typically blighted by heterogeneous noise artefacts such as ghost objects, phase noise, speckle and saturation. In response, previous approaches to utilising radar for robot navigation have often tried to manually extract features from noise corrupted radar scans, commonly relying on simplifying assumptions on the distribution of power returns [1], manually designed heuristics [2], or features designed for different modalities [3, 4]. Nevertheless, the recent seminal work by Cen et al. [2] has firmly established radar as a feasible alternative to complement existing navigation approaches when it comes to ego-motion estimation.

Beyond the basic methodology for pose estimation, the prevalence of vision- and lidar-based approaches in this space has given rise to a number of useful methods beyond those currently utilised for radar. State-of-the-art visual odometry, for example, leverages learnt feature representations [5] as well as attention masks filtering out potentially distracting objects [6]. Lidar-based methods using correlative scan matching [7] typically achieve highly accurate and intuitively interpretable results.

Inspired by this prior art, the aim of our work is to provide a robust radar odometry system which is largely unencumbered by either the typical radar artefacts or by the presence of potentially distracting objects. Our system is explicitly designed to provide *robust*, *efficient* and *interpretable* motion estimates. To achieve this we leverage a deep neural network to learn an essentially artefact and distraction free embedding space which is used to perform efficient correlative matching between consecutive radar scans. Our matching formulation is fully differentiable, allowing us to explicitly learn a representation suitable for accurate pose prediction. The correlative scan matching approach further allows our system to efficiently provide principled uncertainty estimates.

Training our network on over 186,000 examples generated from 216km's of driving, we outperform the previous state of the art in challenging urban environments, reducing errors by over 68% and running an order of magnitude faster. Furthermore, our pose ground truth is gathered in a self-supervised manner, automatically optimising odometry, loop closure, and location constraints, enabling us to adapt to new locations and sensor configurations with no manual labelling effort.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

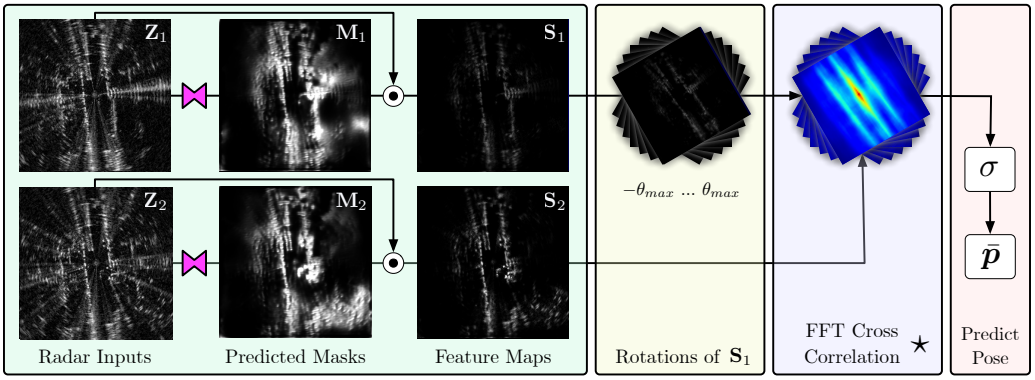


Figure 1: Using masked correlative scan matching to find the optimum pose. Each radar scan is passed to a CNN (pink) in order to generate a mask which is subsequently applied to the input radar scan generating sensor artefact and distractor free representations $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. We then calculate the 2D correlation between $\boldsymbol{S}_2$ and rotated copies of $\boldsymbol{S}_1$ using the fft2d in order to generate the correlation volume $\boldsymbol{C}$. Finally we perform a softargmax operation in order to retrieve the predicted pose. Crucially this pipeline is fully differentiable allowing us to learn the filter masks end to end. A video summary of our approach can be found at: `https://youtu.be/eG4Q-j3_6dk`

2 Related Work

Compared to other sensing modalities such as vision or lidar, radar has received relatively little attention in the context of robot navigation. Prior art in this area largely deploys a more traditional processing pipeline consisting of separate feature extraction, data association and loss minimisation steps, for example using the Iterative Closest Point (ICP) algorithm [8, 9]. For feature extraction some works deploy approaches developed in vision, such as SIFT and SURF [3, 4], others more bespoke methods such as CFAR filtering [10, 1], temporal-space continuity modelling [11, 12], and grid-map features such as Binary Annular Statistics Descriptor (BASD) [13]. Most recently the authors of [2] find point correspondences between point features extracted from raw scans using a shape similarity metric. The final pose is then found by minimising the mean squared error between point correspondences in close to real time.

By making use the Fourier transform correlation-based approaches are in contrast able to perform a dense search over possible point correspondences [14] yielding intuitively interpretable results. Similar approaches have also been applied successfully to lidar scan matching utilising efficient GPU implementations [15] [7]. In comparison to ICP, correlation-based methods have been shown to be significantly more robust to noise in pose initialisation [15]. While robustness and interpretability are desirable, correlation-based methods operate on the assumption that the power returns from a particular location are stationary over time so that a correlation operation produces meaningful results. In reality, this is often not the case – for example when dynamic objects are present in the scene. This problem is particularly pronounced in radar data due to the prevalence of noise artefacts.

Visual odometry systems, in contrast to radar-based ones, have a significant track record of successful application in robotics and beyond. While traditional processing pipelines similar to the one outlined for radar above have been widely deployed in this context (e.g. [16]) there has recently been significant interest in moving away from separate processing steps towards end-to-end approaches. Typically, a neural network is used to regress to a predicted pose directly from consecutive camera images, learning the relationship between features and point correspondences in an integrated manner (e.g. [5, 17]). In [18] the authors extend this approach by learning to predict the optimum pose from stereo images alone. As in many related fields, these end-to-end approaches demonstrate the potential for learning representations generally useful for odometry prediction. However, this comes at the expense of entangling feature representation and data association, which makes the resulting system significantly less interpretable. In contrast, the authors of [19] propose to learn a feature embedding for localising online lidar sweeps into a previously known map, whilst maintaining the interpretability, of a conventional correlative scan matching approach.

Due to the ubiquitous nature of vision-based systems researchers have also addressed challenges beyond the basic pose estimation task such as suppressing noise sources inherent in individual scenes. For example, both [20] and [6] try to mask areas of an image where non-stationary features might be found, which could corrupt the odometry estimate. Of particular relevance is [6], where a deep

neural network is trained using data from other parts of the autonomous system in order to predict human interpretable ephemerality masks indicating the presence of distractor objects in a scene.

Given the large body of evidence that end-to-end approaches tend to outperform more traditional, hand-engineered processing pipelines it is tempting to conclude that our goal here is simply to deploy a deep network to radar odometry. And we do indeed leverage deep learning in our system. However, in doing so we are cognisant that we desire a system which ideally exploits the power of representation learning offered by end-to-end approaches while at the same time leveraging the efficiency, robustness and interpretability offered by correlation-based methods. Thus, inspired by [6] and similar to [19], we deploy a correlation-based matching method as part of an end-to-end system which learns a radar embedding used to produce largely artefact and distraction-free representations optimised for pose prediction. Both the masks obtained as well as the cost-volumes considered remain as interpretable as more traditional approaches.

3 Deep Correlative Scan Matching with Learnt Feature Embeddings

Given two consecutive radar observations $(\boldsymbol{Z}_t, \boldsymbol{Z}_{t-1})$ we wish to determine the relative pose $[\boldsymbol{R}|\boldsymbol{t}] \in \mathbb{SE}(2)$ giving the transformation between the two co-ordinate systems at each time step. In achieving this we aim to harness the efficiency, interpretability and robustness of correlative scan matching assuming that the power returned from each world location is independent of the co-ordinate system it was sensed in. In reality the power returns generated from real world scenes are far from stationary, as dynamic objects move into and out of the field of view of the sensor and pertinent, random noise artefacts obscure the true power returns, limiting the performance of an out-of-the-box correlative scan matching system applied to radar data.

To address this, and inspired by the recent successes of learnt masking for pose prediction in vision [6], we instead perform correlative scan matching over a learnt feature embedding, utilising a deep, fully convolutional network to mask each radar scan as illustrated in Figure 1 (described in Section 3.1). Through this approach we are able to harness the power of deep representation learning whilst ensuring the feature representation remains interpretable through the geometrical constraints imposed by the use of a correlative scan matching procedure. Crucially, we train our network by supervising pose prediction *directly*. In doing so, our network naturally learns to attenuate distractor objects such as moving vehicles and sensor noise as they degrade pose estimation accuracy, whilst preserving features which are likely to be consistent between scans such as walls and buildings. This leads to a 68% reduction in errors over the current state-of-the-art whilst, by making use of efficient correlation computations using the Fast Fourier Transform (FFT), running an order of magnitude faster.

Even in the limit of perfectly stationary power returns, uncertainty in our pose prediction still emanates from pathological solutions arising from the underlying scene topology. In Section 5.2 we show how we are additionally able to quantify the uncertainty in our pose prediction, further aiding the interpretability of our system.

3.1 Correlative Scan Matching with Learnt Feature Embeddings

Let $(\boldsymbol{Z}_{t-1}, \boldsymbol{Z}_t) \in [0,1]^{W \times H}$ denote consecutive observations made by single sweeps of the radar sensor, converted to Cartesian co-ordinates such that $\boldsymbol{Z}_t^{u,v}$ gives the power return at Cartesian co-ordinate (x, y) at time t. Let $\boldsymbol{p} = [\Delta x, \Delta y, \Delta\theta]^T$ denote the parameters of the relative pose $[\boldsymbol{R}|\boldsymbol{t}] \in \mathbb{SE}(2)$ between the co-ordinate frames at $t-1$ and t. We aim to predict the optimum pose from consecutive radar observations harnessing the efficiency, interpretability and robustness of correlative scan matching,

$$\bar{\boldsymbol{p}} = \underset{\boldsymbol{p} \in \mathbb{SE}(2)}{\arg\max}\, \boldsymbol{Z}_t \star \boldsymbol{Z}_{t-1} \tag{1}$$

where $\boldsymbol{Z}_t \star \boldsymbol{Z}_{t-1}$ is defined as the *discrete cross correlation* between $\boldsymbol{Z}_t$ (after being warped by the pose $\boldsymbol{p}$) and $\boldsymbol{Z}_{t-1}$.

In order to solve for the predicted pose $\bar{\boldsymbol{p}}$ we consider a brute force approach: we discretise our search search space, calculating the cross correlation score for each pose on a regular grid of pose candidates before utilising a soft-argmax operation to solve for the optimum pose to sub-grid resolution accuracy. This is achieved efficiently using Algorithms 2 and 3. By utilising bi-linear interpolation for all re-size and rotation operations, and computing the cross-correlation using the highly efficient 2D Fast Fourier Transform, we are able to search for the optimum pose over a large search area, efficiently solving (1) whilst still maintaining end-to-end differentiability.

Central to this approach is an assumption that the power returned from each world location is independent of the co-ordinate system it was sensed in. This assumption rarely holds in practice. Random noise artefacts, dynamic objects and changing scene occlusion cause fluctuations in the power field, degrading the accuracy of conventional correlation-based approaches applied to radar. To counter this, we propose to learn a feature representation $\boldsymbol{S}$ specifically optimised for correlative scan matching by filtering each radar scan $\boldsymbol{S} = \boldsymbol{M} \odot \boldsymbol{Z}$ with a mask $\boldsymbol{M} = f_\alpha(\boldsymbol{Z})$ generated by a neural neural network f_α (where $\odot$ denotes Hadamard product). By limiting each element of the mask to $[0, 1]$ (using an element wise sigmoid), the network is able to learn to filter out distractor objects and noise in each sensor observation, before correlative scan matching is applied to find the optimum pose. By leveraging the differentiability of our approach for predicting $\bar{\boldsymbol{p}}$, we are able to use Algorithm 1, to learn a radar feature embedding specifically optimised for correlative scan matching by minimising the Mean Squared Error (MSE) over the training set, $\mathcal{D} = \{(\boldsymbol{Z}_t, \boldsymbol{Z}_{t-1}, \boldsymbol{p})^n\}_{n=1}^N$,

$$\alpha^* = \arg\min_\alpha \mathbb{E}_{\boldsymbol{p}\sim\mathcal{D}}\left[||\bar{\boldsymbol{p}} - \boldsymbol{p}||^2\right] \tag{2}$$

to update our network parameters α using conventional stochastic gradient descent based optimisers.

3.2 Pose Uncertainty Estimation

Pathological solutions arising from the underlying scene topology increase the uncertainty in our pose prediction even in the case of perfectly stationary power returns. In the real world identifying such cases is important in order to ensure robust operation. To this end, our approach also affords us a principled mechanism to estimate the uncertainty in each element of the predicted pose.

In performing the soft-argmax operation, we first apply a temperature controlled softmax over the correlation scores for each candidate pose, to give weights $\boldsymbol{\omega} = \text{Softmax}(\beta \boldsymbol{C})$, interpreted as the probability that each pose candidate is optimum. Assuming that our predicted pose is Gaussian distributed we can quantify the uncertainty in each pose prediction by using the weights $\boldsymbol{\omega}$ to predict both the mean pose $\bar{\boldsymbol{p}}$ *and* the predicted co-variance $\bar{\boldsymbol{\Sigma}}$,

$$\bar{\boldsymbol{p}} = \sum_s \omega_s \boldsymbol{p}_s \quad \bar{\boldsymbol{\Sigma}} = \sum_s \omega_s \boldsymbol{p}_s \boldsymbol{p}_s^T - \bar{\boldsymbol{p}}\bar{\boldsymbol{p}}^T \quad p(\boldsymbol{p}|\boldsymbol{S}_t) \approx \mathcal{N}(\boldsymbol{p}|\bar{\boldsymbol{p}}, \bar{\boldsymbol{\Sigma}}) \tag{3}$$

where we sum over all pose candidates. The softmax temperature parameter β plays an important role here: for high β our system is biased to the pose candidate with highest correlation and a low co-variance, whilst for low β to a weighted mean over a greater number of pose candidates and high co-variance.

Algorithm 1: Training

Input:

$\mathcal{D}$ // Dataset
$\boldsymbol{r}$ // Search Region giving min and max range in $\Delta x, \Delta y, \Delta\theta$
$\boldsymbol{\delta}$ // Grid resolution in each dimension $\delta_x, \delta_y, \delta_\theta$
β // Softmax Temperature Parameter
ϵ // Learning Rate
α // Initial Network Parameters

$\boldsymbol{G}_{xy\theta} = MeshGrid(\boldsymbol{r}, \boldsymbol{\delta})$
while *not converged* **do**
  $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{p} \leftarrow Sample(\mathcal{D})$
  $\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow f_\alpha(\boldsymbol{Z}_1), f_\alpha(\boldsymbol{Z}_2)$
  $\boldsymbol{S}_1, \boldsymbol{S}_2 \leftarrow \boldsymbol{M}_1 \odot \boldsymbol{Z}_1, \boldsymbol{M}_2 \odot \boldsymbol{Z}_2$
  $\boldsymbol{C} \leftarrow GetCorrelation(\boldsymbol{G}_{xy\theta}, \boldsymbol{S}_1, \boldsymbol{S}_2)$
  $\bar{\boldsymbol{p}} \leftarrow SoftArgMax(\boldsymbol{G}_{xy\theta}, \boldsymbol{C}, \beta)$
  $\alpha \leftarrow \alpha - \epsilon\nabla_\alpha \mathcal{L}(\bar{\boldsymbol{p}}; \boldsymbol{p})$
end

Algorithm 2: Correlation

function *GetCorrelation*($\boldsymbol{G}_{xy\theta}$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$) **:**
  $n_x, n_y, n_\theta \leftarrow Shape(\boldsymbol{G}_{xy\theta})$
  $\boldsymbol{C} = Zeros([n_x, n_y, n_\theta])$
  $\boldsymbol{G}_{xy}, \boldsymbol{G}_\theta \leftarrow \boldsymbol{G}_{xy\theta}$
  $\boldsymbol{X}_1, \boldsymbol{X}_2 \leftarrow Resize(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{G}_{xy})$
  par for $i \leftarrow 1$ **to** n_θ **:**
    $\boldsymbol{X}_1^{\mathsf{R}} \leftarrow Rotate(\boldsymbol{X}_1, \boldsymbol{G}_\theta[i])$
    $\boldsymbol{C}[:, :, i] \leftarrow$ $\mathsf{fft2d}^{-1}\left(\mathsf{fft2d}(\boldsymbol{X}_1^{\mathsf{R}}) \odot \mathsf{fft2d}(\boldsymbol{X}_2^{\mathsf{C}})\right)$
  return $\boldsymbol{C}$

Algorithm 3: Soft Arg Max

function *SoftArgMax*($\boldsymbol{G}_{xy\theta}$, $\boldsymbol{C}$, β) **:**
  $\boldsymbol{\omega} \leftarrow Softmax(\beta \boldsymbol{C})$
  $\boldsymbol{G}_x, \boldsymbol{G}_y, \boldsymbol{G}_\theta \leftarrow \boldsymbol{G}_{xy\theta}$
  $\Delta x \leftarrow \sum_{i,j,k} \left(\boldsymbol{\omega} \odot \boldsymbol{G}_x\right)[i, j, k]$
  $\Delta y \leftarrow \sum_{i,j,k} \left(\boldsymbol{\omega} \odot \boldsymbol{G}_y\right)[i, j, k]$
  $\Delta\theta \rightarrow \sum_{i,j,k} \left(\boldsymbol{\omega} \odot \boldsymbol{G}_\theta\right)[i, j, k]$
  return$[\Delta x, \Delta y, \Delta\theta]$

4 Experimental Setup

4.1 Dataset

To evaluate our approach we use the recently released Oxford Radar RobotCar Dataset [21], a radar extension to the Oxford RobotCar Datsset [22], which provides Navtech CTS350-X radar data as well as ground truth poses. The Navtech CTS350-X is a Frequency Modulated Continuous Wave (FMCW) scanning radar without doppler information, configured to return 3768 power readings at a resolution of 4.32cm across 400 azimuths at a frequency of 4Hz (corresponding to a maximum range of 163m). The beam spread is 2 degrees in azimuth and 25 degrees in elevation with a cosec squared beam pattern. We randomly split the traversals into training (80%) and evaluation (20%) partitions. We additionally run spatial cross validation experiments, where each split occupies a different real world region of the dataset. Further information on these results and the dataset can be found in the appendix B.2, C.1.

To validate the advantages of learning masks directly from pose supervision we compare against supervising the learnt masks directly on the proxy task of predicting temporally static occupied cells. Training data for this is generated using a similar approach to [6]. For each radar scan we warp the nearest radar sensor observation from each training traversal into the current pose before applying a static power threshold. We then form a 2D histogram counting the number of thresholded power returns that fall in each Cartesian grid cell. Any grid cell with more than 9 consistent observations is assumed to be temporally stable and is labelled with a 1, whilst every other cell is set to 0. This is repeated for every pose in every dataset. Examples of the masks generated by this approach can be found in the appendix B.1.

4.2 Network Architecture and Training

In all experiments we use a U-Net style architecture [23] in which we encode the input tensor through the repeated application of two convolutional layers (filter size 3x3) with ReLU activations before a max pooling operation. After each max pool the width and height of the tensor are reduced by a factor of 2 whilst the number of features is doubled, starting from 8 at the input to 256 at the bottleneck of the network (corresponding to 5 max pools). The feature tensor is then converted back to the original shape by the decoder through the application of bilinear upsampling followed by two convolutional layers increasing the width and height and decreasing the feature channels by a factor of 2. Skip connections at each level are implemented allowing information to flow from encoder to decoder by stacking each representation with the output from the bilinear upsampling layer in each case. The final convolutional layer has a single output channel with a sigmoid activation to limit the range to $[0, 1]$. We experiment with learning to mask both Cartesian and Polar radar representations, as well as both *single* and *dual* configurations. In the dual case radar observations are concatenated and passed as a single input producing two masks (instead of one) at the output. An architecture diagram can be found in the appendix A.1. In all cases we consider a search region of $[-50\text{m}, 50\text{m}]$ in Δx and Δy and $[-\pi/12, \pi/12]$ in $\Delta\theta$. We experiment with the three grid resolutions $[0.2\text{m}, 0.4\text{m}, 0.8\text{m}]$ for δ_x and δ_y whilst fixing δ_θ to $\pi/360$.

Our network is implemented in Tensorflow [24] and trained using the Adam Optimiser [25] (learning rate $1\text{e}-5$ and batch size 5) until the loss on a small validation set is a minimum. When training our network with pose supervision we minimise the loss proposed in (2). We performed a grid search over the optimum value of β and found setting it to 1 gave good performance.

4.3 Evaluation Metrics and Baselines

Our primary baseline is the current state of art for radar odometry [2] (implemented in C++) in which the authors extract point features from consecutive radar scans before scan matching using a global shape similarity score and refining by minimising mean squared error. Our radar was set to a range resolution of 4.32cm, whilst the original algorithm was developed for a 17.28cm resolution. As such we compare against [2] with full resolution radar scans and downsampled (with max pooling) to 17.28cm. For context we also provide visual odometry estimates (as in [2]). To assess the benefits of learning feature masks specifically optimised for pose prediction, we benchmark against scan matching on the raw radar scans without masking, as well as using the method proposed in [6] with mask labels generated as described in Section 4.1. In this setup, we supervise (using a binary cross entropy loss) the learnt masks directly (instead of supervising pose prediction). We also benchmark against taking an off the shelf deep odometry model and training this for the task of radar pose prediction. Specifically we use the UnDeepVO model proposed in [18].

For all evaluations we follow the KITTI odometry benchmark [26]. For each 100m offset up to 800m, we calculate the average residual translational and angular error for every example in the datastet normalising by the distance travelled. Finally, we average these values. Due to highly skewed error distributions we report Inter Quartile Range (IQR) for each method instead of the standard deviation. All timing statistics are calculated using a 2.7 GHz 12-Core Intel Xeon E5 CPU and Nvidia Titan Xp GPU by averaging across 1000 predictions.

4.4 Uncertainty Evaluation

To assess the quality of the uncertainty predicted by our approach we observe that if our pose distribution is Gaussian than the Mahalanobis error

$$d^2 = (\boldsymbol{p} - \bar{\boldsymbol{p}})^T \bar{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{p} - \bar{\boldsymbol{p}}), \quad d^2 \sim \chi^2(3) \tag{4}$$

should be chi-squared distributed with degrees of freedom equal to the state dimensionality of $\boldsymbol{p}$ (in this case three). As the mean of a chi-squared distribution is equal to the distributions degrees of freedom, by averaging the mean Mahalanobis distance over the test dataset $\bar{d}^2 = \frac{1}{N} \sum_n d_n^2$ we can assess to what degree the uncertainties predicted by our approach are calibrated to the test errors [7]. Specifically, if $\bar{d}^2 \ll 3$ then our model is overly conservative in its predictions whilst if $\bar{d}^2 \gg 3$ it is overly confident. In Section 5.2 we use this result to tune the temperature parameter β to provide us with realistic uncertainties, that are calibrated to the true errors in our system.

5 Results

In this section we evaluate the performance of our approach. We find by utilising correlative scan matching in combination with a learnt radar feature embedding we are able to significantly outperform the previous state of art in both prediction performance and speed. Additionally, we show how, by tuning the temperature parameter of the softargmax, we are able to predict realistic and calibrated uncertainties, further increasing the interpretability of our system and allowing us to identify pathological cases, crucial for robust operation in the real world.

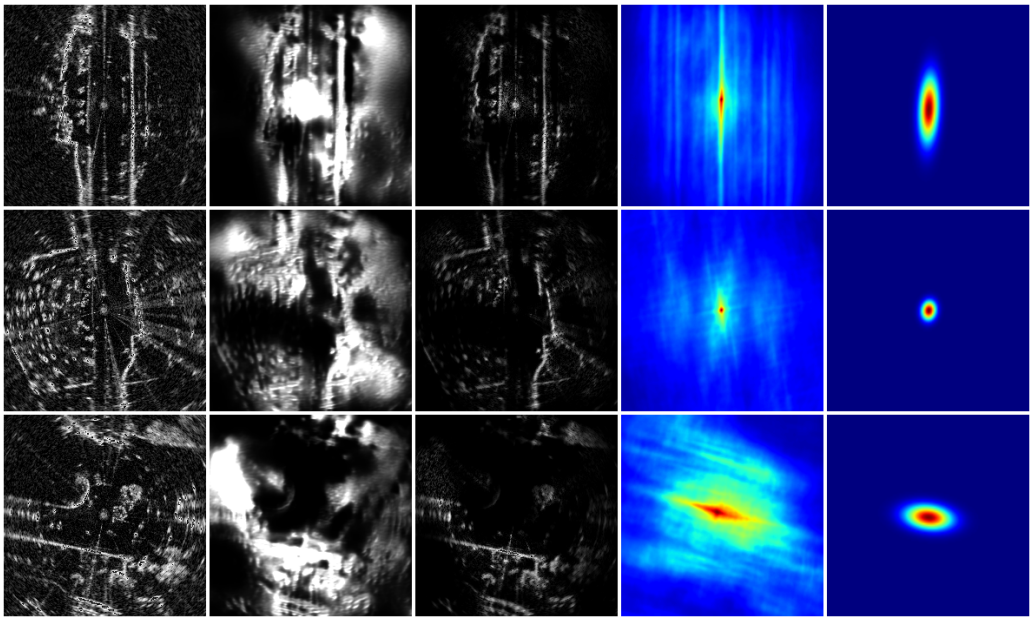


Figure 2: Qualitative examples generated from our best performing model. Our network learns to mask out noise and distractor objects whilst preserving temporally consistent features such as walls, well suited for pose prediction. Predicted co-variance is high for pathological solutions arising through a lack of constraints in the x-direction (top), whilst stationary well-constrained scenes result in low co-variance (middle). Motion blur increases the uncertainty due to ambiguous point correspondence (bottom). Further examples can be found in Figure 8 in the appendix.

Benchmarks	Resolution (m/pixel)	Translational error (%) Mean	IQR	Rotational error (deg/m) Mean	IQR	Runtime (s) Mean	Std.
RO Cen Full Resolution [2]	0.0432	8.4730	5.7873	0.0236	0.0181	*0.3059*	0.0218
RO Cen Equiv. Resolution [2]	0.1752	*3.7168*	3.4190	*0.0095*	0.0095	2.9036	0.5263
Raw Scan	0.2	8.3778	7.9921	0.0271	0.0274	0.0886	0.0006
Supervised Masks Polar	0.2	5.9285	5.6822	0.0194	0.0197	0.0593	0.0014
Supervised Masks Cart	0.2	5.4827	5.2725	0.0180	0.0186	0.0485	0.0013
Adapted Deep VO Cart [18]	0.2	4.7683	3.9256	0.0141	0.0128	0.0060	0.0003
Adapted Deep VO Polar [18]	-	9.3228	8.3112	0.0293	0.0277	0.0093	0.0002
Visual Odometry [16]	-	3.9802	2.2324	0.0102	0.0065	0.0062	0.0003
Ours							
Polar	0.8	2.4960	2.1108	0.0068	0.0052	0.0222	0.0013
	0.4	1.6486	1.3546	0.0044	0.0033	0.0294	0.0012
	0.2	1.3634	1.1434	0.0036	0.0027	0.0593	0.0014
Cartesian	0.8	2.4044	2.0872	0.0065	0.0047	0.0113	0.0012
	0.4	1.5893	1.3059	0.0044	0.0035	0.0169	0.0012
	0.2	1.1721	0.9420	0.0031	0.0022	0.0485	0.0013
Dual Polar	0.8	2.5762	2.0686	0.0072	0.0055	0.0121	0.0003
	0.4	2.1604	1.9600	0.0067	0.0053	0.0253	0.0006
	0.2	1.2621	1.1075	0.0036	0.0029	0.0785	0.0007
Dual Cart	0.8	2.7008	2.2430	0.0076	0.0054	**0.0088**	0.0007
	0.4	1.7979	1.4921	0.0047	0.0036	0.0194	0.0010
	0.2	**1.1627**	0.9693	**0.0030**	0.0030	0.0747	0.0005

Table 1: Odometry estimation and timing results. Here "RO Cen" [2] is our primary benchmark reported for 0.04m (full resolution) and, by downsampling, 0.17m (equivalent resolution for which the approach was originally developed). For comparison we also provide performance results for correlative scan matching on the *raw* power returns, for mask supervision (instead of supervising the predicted pose directly), and adapting the deep VO network proposed in [18], alongside visual odometry [16] for context. All baselines performed best at 0.2 m/pixel resolution where applicable and the rest are omitted for clarity. We experiment with both polar and Cartesian network inputs at multiple resolutions. Our approach outperforms the current state of the art, "RO Cen" (italics), for all configurations of Cartesian / polar inputs and independent / dual masking at all resolutions. Our best performing models in terms of speed and odometry performance are marked in bold.

5.1 Odometry Performance

Table 1 gives our prediction and timing results. We experiment with both Cartesian and Polar inputs to the masking network (converting the latter to Cartesian co-ordinates before correlative scan matching), as well as experimenting with single and dual configurations as detailed in Section 4.2.

At all resolutions and configurations we beat the current state of the art with our best model reducing errors by 68% in both translation and rotation, whilst running over 4 times faster. Our fastest performing model runs at over 100Hz whilst still reducing errors on the state of the art by 28% in translational and 20% in rotational error (further results exploring the accuracy-speed trade off can be found in A.2). We find that Cartesian network inputs typically outperform Polar (presumably because correlative scan matching is performed in Cartesian space). Dual input configurations also typically outperform passing single sensor observations to the masking network.

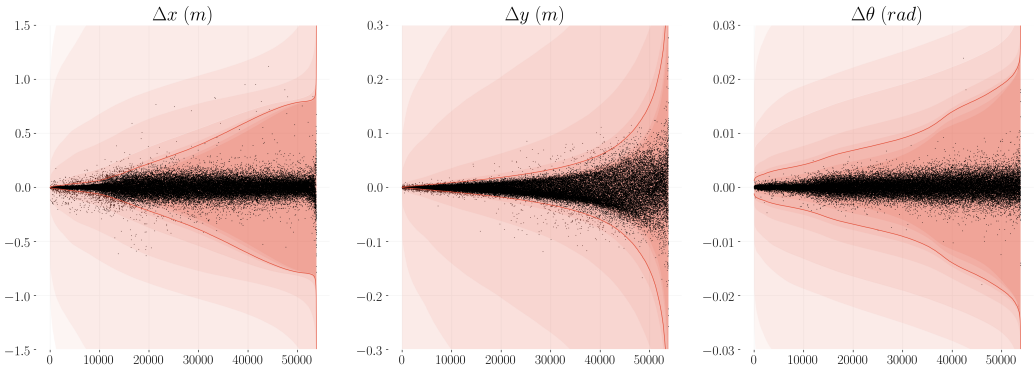
Key to our approach is learning a radar feature embedding that is optimised for pose prediction: compared to correlative scan matching on the raw radar power returns this allows us to reduce errors by over 85%. As predicted, optimising masks directly for pose prediction results in a higher prediction accuracy than mask supervision labelling the temporally stationary scene directly. We also find that simply adapting a deep odometry approach to radar results in significantly worse performance. Our approach in contrast makes use of the inherent top down representation of a radar observation which lends itself to a correlative scan matching procedure, whilst learning to mask out noise artefacts which make pose prediction in radar uniquely challenging. In addition, by adopting a correlative scan matching approach, our results remain interpretable: Figure 2 shows several qualitative examples in which the network learns to mask noise artefacts and dynamic objects in the scene whilst preserving features which are likely to be temporally stationary such as walls.

5.2 Uncertainty Prediction

In addition to the boosts in performance and speed afforded by our approach, we are also able to estimate the uncertainty in each pose prediction: by interpreting the weights generated through the temperature controlled softargmax operation as the probability that each pose candidate is optimum we predict the co-variance $\bar{\mathbf{\Sigma}}$ in our prediction as detailed in Section 3.2.

We now use the methodology proposed in Section 4.4 to tune the temperature parameter β such that the mean Mahalanobis distance $\bar{d}^2 \approx 3$ producing uncertainties $\bar{\mathbf{\Sigma}}$ that are calibrated to the errors in our system. Naively perturbing the temperature parameter away from its original value β_0 degrades pose prediction performance as the feature mask no longer corresponds to the β it was optimised for. Instead, we calculate the predicted pose using β_0, whilst varying β to tune the covariance matrix. The results of this process (for the 0.8m resolution single mode Cartesian model from Table 1) are shown in Figure 5.2 alongside the marginal distributions for the uncertainty in each pose component plotted with the true errors in our system ordered by predicted uncertainty. For a temperature parameter $\beta = 2.789$ the mean Mahalanobis distance $\bar{d}^2$ is equal to 2.99 giving us well calibrated uncertainty predictions, whilst temperature parameters above and below this value are overly certain and conservative respectively. There is a clear correlation between error and uncertainty with most errors falling within the predicted uncertainty bounds.

Figure 2 shows Gaussian heat maps generated through our approach; the results are highly intuitive with feature embeddings well constrained in each dimension having smaller and symmetric co-variance, whilst pathological solutions arising from a lack of scene constraints increase the uncertainty in Δx.



Temperature Parameter β	4.467	3.548	**2.789**	2.512	1.995	1.413	1.122
Mean Mahalanobis Distance $\bar{d}^2$	4268.471	57.099	**2.992**	1.244	0.283	0.065	0.030

Figure 3: The marginal distributions and errors (black) in each pose component for each example in our test set ordered by predicted uncertainty. The colours correspond to 1.98 standard deviation bounds plotted for each of the temperature parameters given in the table with dark to light moving through the table left to right. The red line corresponds to the standard deviation bound plotted for $\beta = 2.789$ corresponding to a mean Mahalanobis distance of $\bar{d}^2 = 2.99$. For this temperature setting the majority of the errors fall within the 1.98 standard deviation bound. Note the y axis in each case has a different scale.

6 Conclusions

By using a learnt radar feature embedding in combination with a correlative scan matching approach we are able to improve over the previous state of the art, reducing errors in odometry prediction by over 68% and running an order of magnitude faster, whilst remaining as interpretable as a conventional scan matching approach. Additionally, our method affords us a principled mechanism by which to estimate the uncertainty in the pose prediction, crucial for robust real world operation.

Our approach for attaining calibrated uncertainties currently relies on tuning a pre-trained model. An interesting direction for future work would be to incorporate this tuning process into the training pipeline, learning not only a radar feature embedding optimised for pose prediction but also for uncertainty estimation. We leave this for future work.

Acknowledgments

This work was supported by the UK EPSRC Doctoral Training Partnership and EPSRC Programme Grant (EP/M019918/1). The authors also would like to acknowledge the use of Hartree Centre resources and the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (http://dx.doi.org/10.5281/zenodo.22558).

References

[1] D. Vivet, P. Checchin, and R. Chapuis. Localization and mapping using only a rotating fmcw radar sensor. *Sensors*, 13(4):4527–4552, 2013.

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

[3] J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom. Radar slam using visual features. *EURASIP Journal on Advances in Signal Processing*, 2011(1):71, 2011.

[4] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio. Landmark based radar slam using graph optimization. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 2559–2564. IEEE, 2016.

[5] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017.

[6] D. Barnes, W. Maddern, G. Pascoe, and I. Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1894–1900. IEEE, 2018.

[7] W. Maddern, G. Pascoe, and P. Newman. Leveraging experience for large-scale lidar localisation in changing cities. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1684–1691. IEEE, 2015.

[8] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[9] E. Ward and J. Folkesson. Vehicle localization with low cost radar sensors. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*. Institute of Electrical and Electronics Engineers (IEEE), 2016.

[10] H. Rohling. Ordered statistic cfar technique-an overview. In *Radar Symposium (IRS), 2011 Proceedings International*, pages 631–638. IEEE, 2011.

[11] E. Jose and M. D. Adams. An augmented state slam formulation for multiple line-of-sight features with millimetre wave radar. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3087–3092. IEEE, 2005.

[12] E. Jose and M. D. Adams. Relative radar cross section based feature identification with millimeter wave radar for outdoor slam. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 425–430. IEEE, 2004.

[13] M. Rapp, K. Dietmayer, M. Hahn, F. Schuster, J. Lombacher, and J. Dickmann. Fscd and basd: Robust landmark detection and description on radar-based grids. In *Microwaves for Intelligent Mobility (ICMIM), 2016 IEEE MTT-S International Conference on*, pages 1–4. IEEE, 2016.

[14] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine. Radar scan matching slam using the fourier-mellin transform. In *Field and Service Robotics*, pages 151–161. Springer, 2010.

[15] E. B. Olson. Real-time correlative scan matching. *Ann Arbor*, 1001:48109, 2009.

[16] W. Churchill. *Experience Based Navigation: Theory, Practice and Implementation*. PhD thesis, University of Oxford, Oxford, United Kingdom, 2012.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[18] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[19] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun. Learning to localize using a lidar intensity map. In *Conference on Robot Learning*, pages 605–616, 2018.

[20] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman. Distraction suppression for vision-based pose estimation at city scales. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3762–3769. IEEE, 2013.

[21] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. *arXiv preprint arXiv:1909.01300*, 2019.

[22] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

A Implementation

A.1 Masking Network Architecture

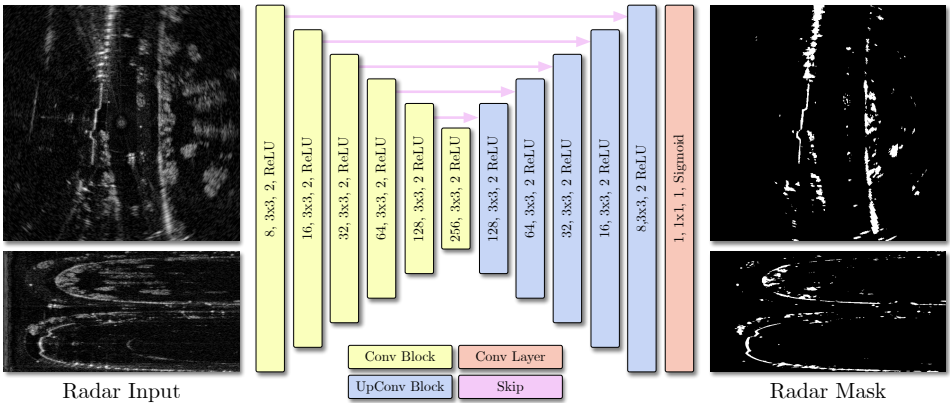


Figure 4: Architecture diagram of the radar masking network. Layers are detailed by output channels, kernel sizes, repetitions and activations respectively. The final network layer has a single output channel with a sigmoid activation to limit the masking range to $[0, 1]$. We experiment using the masking network in both Cartesian and Polar radar representations. Additionally we investigate the impact of modifying the *single* configuration shown to *dual* configuration, in which sequential radar observations used for odometry prediction are concatenated and passed as a single input producing two masks (instead of one) at the output. For more details please refer to the text in Section 4.2. The predictions shown are from a network directly supervised with baseline masks detailed in Section B.1.

A.2 Speed vs Accuracy Trade Off

By reducing our Cartesian grid resolution before calculating the correlation volume, for the same grid coverage we are able to predict the optimum pose in a shorter amount of time to the detriment of pose prediction accuracy. Estimating this trade off for our trained models is challenging and requires many training runs. Instead we investigate the speed-accuracy trade off by performing correlative scan matching on the raw power returns at a variety of grid resolutions according to Algorithms 2 and 3. The results for this process are displayed in Figure 5 which we use to choose the grid resolutions for the main results presented in Table 5.

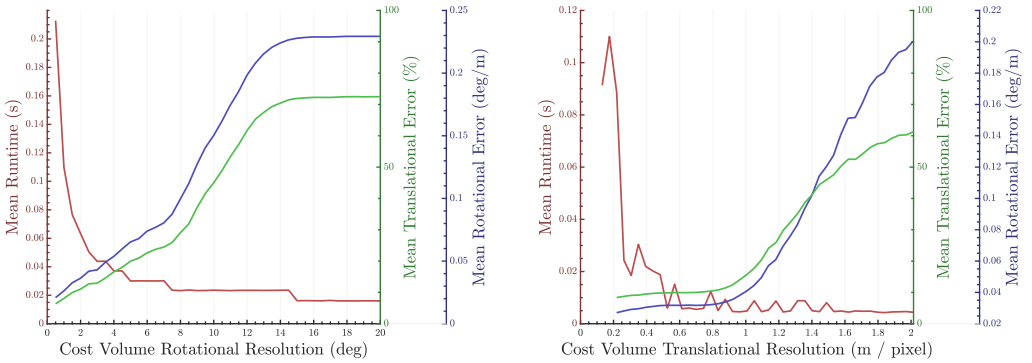


Figure 5: Translational error (green), angular error (blue) and run time (red) as a function of Cost volume resolution in degrees (left) and metres per pixel (right). In the case of limited computational resources or required pose estimate accuracy it is possible to flexibly trade off performance and computational speed.

B Data

B.1 Baseline Masks

To validate the advantages of learning masks directly from pose supervision we compare against supervising the learnt masks directly on the proxy task of predicting temporally static occupied cells. To generate static mask labels we use a similar approach to [6] as detailed in Section 4.1, whereby nearby radar scans from different traversals are warped into the current sensor frame to assess temporal stability. Even with a large corpus of accurately labelled masks identifying static structure suitable for estimating odometry, we observe increased performance by training directly on the task of pose estimation.

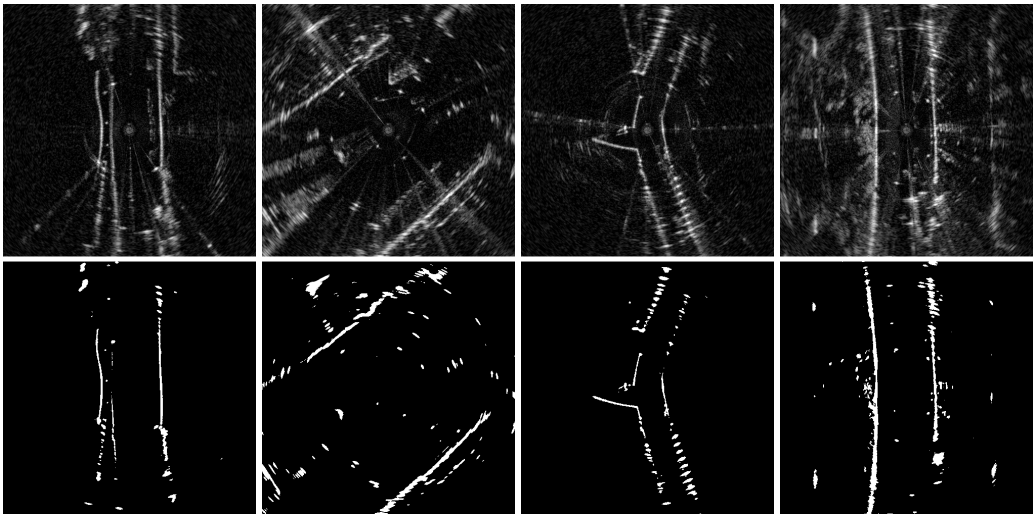
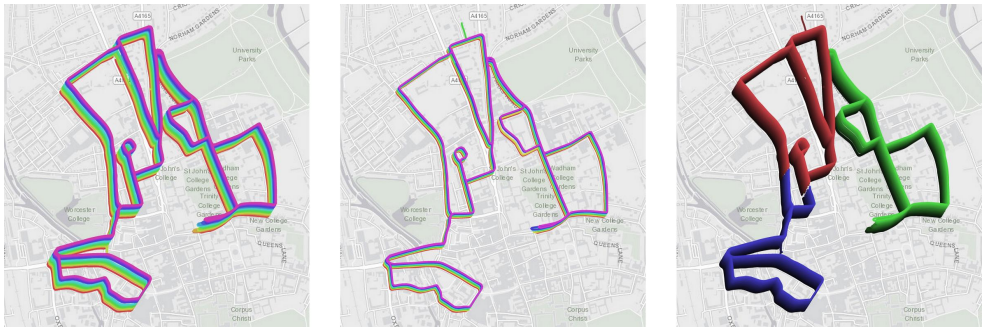


Figure 6: Example generated baseline masks used to supervise the radar masking network directly. For a given raw radar scan at time t (top) we can automatically generate high quality baseline masks identifying structure useful for pose estimation (bottom).

B.2 Dataset Splits



Training Traversals | Testing Traversals | Spatial Cross Validation

Figure 7: Trajectories of the ground truth optimised pose chains used for the 25 training (left) and 7 evaluation (middle) traversals from the Oxford Radar RobotCar Dataset [21] covering a wide variety of traffic and other challenging conditions in complex urban environments. In addition to splitting the dataset temporally we provide spatial cross validation results (right), detailed in Section C.1. Each traversal is incrementally offset with a unique colour for visualisation.

C Results

C.1 Spatial Cross Validation

In Section 5.1 we achieve radar odometry performance far exceeding the state of the art. However we train and evaluate on scenes from the same spatial locations. To assess how well our models generalise to previously un-seen scenes, in this section we train and evaluate our models using spatial cross validation: splitting our traversal loop into three, we train on two out of the three splits, evaluate performance on the third and average results across hold-out splits. Due to the computational demands of training models from scratch on each split, we train our medium resolution model (which is faster to train but has slightly worse performance than its higher resolution counterpart).

Our best model reduces average cross validation errors over the current state of the art by over 25% in translational and 11% in rotational error whilst running over 15x faster. Using this training paradigm we reduce the effective training data diversity by a third. We attribute this to the slight reduction in performance in comparison to the results presented in Section 5.1. We theorise we could significantly boost performance by moving to our highest resolution model also.

Benchmarks	Resolution (m/pixel)	Translational error (%) Mean	IQR	Rotational error (deg/m) Mean	IQR
RO Cen Full Res [2]	0.0432	6.3813	4.6458	0.0189	0.0167
RO Cen Equiv.* [2]	0.1752	*3.6349*	3.3144	*0.0096*	0.0095
Raw Scan	0.4	8.4532	8.0548	0.0280	0.0282
Adapted Deep VO Cart [18]	0.4	11.531	9.6539	0.0336	0.0307
Adapted Deep VO Polar [18]		14.446	11.838	0.0452	0.0430
Visual Odometry [16]		3.7824	1.9884	0.0103	0.0072
Ours					
Polar	0.4	2.8115	2.4189	0.0086	0.0084
Cart	0.4	3.2756	2.8213	0.0104	0.0100
Dual Polar	0.4	3.2359	2.5760	0.0098	0.0091
Dual Cart	0.4	**2.7848**	2.2526	**0.0085**	0.0080

Table 2: Spatial cross validation odometry estimation results. Our approach outperforms the benchmark (italics) in a large proportion of the experiments and we would expect a similar boost in performance to Section 5.1 by moving from our medium to highest resolution model. Our best performing model in terms odometry performance is marked in bold.

C.2 Additional Evaluation Examples

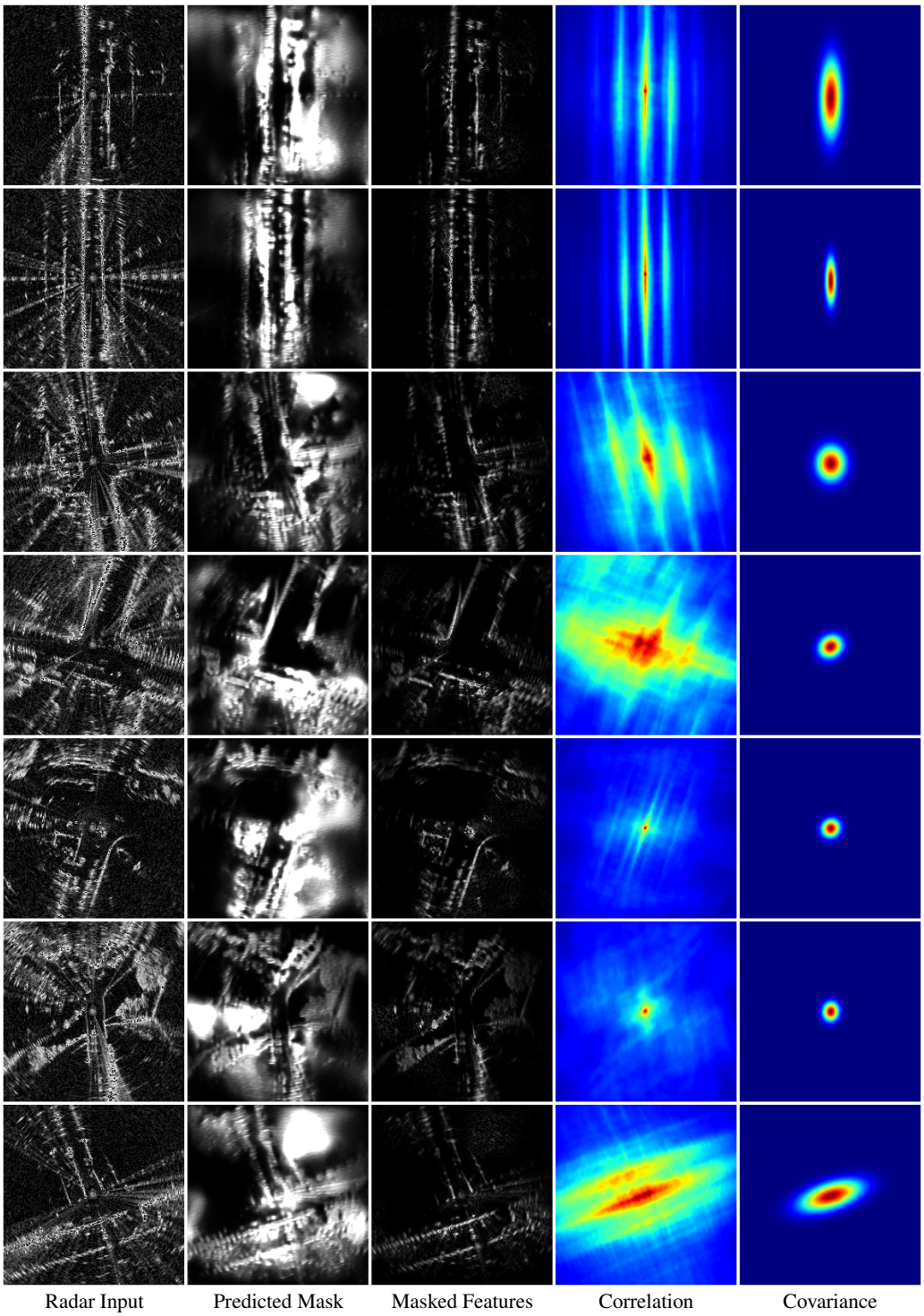


Figure 8: Additional qualitative examples generated from our best performing model. The masks generated from our network filter out noise and distractor objects in the scene whilst preserving temporally consistent features such as walls, well suited for pose prediction. From left to right the raw Cartesian radar scan, the predicted network mask, the masked radar scan, the correlation volume and the fitted gaussian to the correlation volume after temperature weighted softmax.

5.1 Probabilistic Correlative Scan

An alternative derivation for the pose estimation approach just developed is now presented, considering the problem from a Bayesian perspective. Alongside providing theoretical insights, through this consideration, a natural mechanism also emerges for accounting for *prior* information about the relative pose between two radar scans. Several of the results derived in this section rely on some understanding of the mechanisms behind Bayesian inference, a short introduction to which is provided in Sec. 8.2.2.

Bayesian Pose Estimation Let $\mathbf{x}_t \in \mathbb{SE}(2)$ denote a pose describing how a robot moves from time $t-1$ to t. The pose $\mathbf{x}_t$ is assumed unknown and the aim is to infer its value given a radar measurement $\mathbf{Z}_t \in \mathbb{R}^{H \times W}$.[1]

Adopting a Bayesian approach, this is achieved by assuming a prior for the pose, $p(\mathbf{x}_t)$, which may be formed by taking account additional sensor information (e.g. wheel odometry), or by encoding prior assumptions about the motion of the robot (e.g. vehicle models and/or velocity information), for example. The prior is then updated in light of an observation $\mathbf{Z}_t \sim p(\mathbf{Z}_t|\mathbf{x}_t)$ to form the *posterior*

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{Z}_t)} = \frac{p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{\int p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)d\mathbf{x}_t} \tag{5.1}$$

where $p(\mathbf{Z}_t) = \int p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)d\mathbf{x}_t$ is the *marginal evidence*.

Correlative Likelihood Central to correlative scan matching approaches is the assumption that high correlation between two consecutive sensor observations corresponds to a likely pose $\mathbf{x}_t$. Converting this assumption to a likelihood, one possible choice is to consider

$$p(\mathbf{Z}_t|\mathbf{x}_t) \triangleq \tfrac{1}{C}\exp\{\beta c(\mathbf{Z}_t;\mathbf{x}_t)\} \tag{5.2}$$

where $c(\mathbf{Z}_t;\mathbf{x}_t) \triangleq \mathbf{Z}_t \star \tilde{\mathbf{Z}}_t$ is the correlation between the current sensor observation $\mathbf{Z}_t$ and the sensor observation from the previous time step $\mathbf{Y}_{t-1}$ transformed into

[1] Note that the pose is now denoted $\mathbf{x}$ instead of $\mathbf{p}$ to avoid the pose being confused with distributions p

the current co-ordinate system at $\tilde{\mathbf{Z}}_t \triangleq w(\mathbf{Y}_{t-1}, \mathbf{x}_t)$ using pose $\mathbf{x}_t$.[2] Here $p(\mathbf{Z}_t|\mathbf{x}_t)$ is defined as a Gibbs distribution with temperature parameter $\beta > 0$, potential $c(\mathbf{Z}_t; \mathbf{x}_t)$ and *unknown* normalisation constant C [52].

Approximating the Posterior As the normalisation constant is *unknown*, determining the posterior distribution analytically as in Eq. (5.1) is no longer feasible. However, it is still possible to approximate the posterior in this case using an importance sampling approach (as detailed in Sec. 8.2.2.2). Here, the posterior distribution is approximated as a set of weighted particles $p(\mathbf{x}_t|\mathbf{Z}_t) \approx \sum_{s=1}^{S} \omega_s \, \mathtt{Dir}(\mathbf{x}|\mathbf{x}_s)$ with

$$\mathbf{x}_s \sim q(\mathbf{x}_s) \qquad \tilde{\omega}_s = \frac{\tilde{p}(\mathbf{x}_s)}{\tilde{q}(\mathbf{x}_s)} \qquad \omega_s = \frac{\tilde{\omega}}{\sum_s \tilde{\omega}_s} \tag{5.3}$$

given a proposal distribution $q(\mathbf{x})$ and defining the un-normalised posterior as $\tilde{p}(\mathbf{x}_t) \propto p(\mathbf{x}_t|\mathbf{Z}_t) \triangleq p(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)$.

Pose Estimation As Bayesian Risk Minimisation The optimum pose estimate in the Bayesian case is found considering minimising the *Bayesian Risk*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}'} \mathbb{E}_{p(\mathbf{x}|\mathbf{Z})}\{\ell(\mathbf{x}, \mathbf{x}')\} \tag{5.4}$$

where $\ell : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is a chosen risk function. Defining $\ell(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2^2$ it can be shown that the optimum estimate for the pose in this case is given as

$$\bar{\mathbf{x}} = \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}\{\mathbf{x}\} \approx \sum_s \omega_s \mathbf{x}_s \tag{5.5}$$

which in the last step is estimated in accordance with Eq. (8.37).

Equivalence To Masking By Moving Approach The analysis conducted so far trivially extends to radar observation masked using a neural network by defining

$$\mathbf{M}_t, \mathbf{M}_{t-1} \triangleq f_\alpha(\mathbf{Y}_t, \mathbf{Y}_{t-1}) \qquad \text{masks} \tag{5.6}$$

$$\mathbf{Z}_t, \mathbf{Z}_{t-1} \triangleq \mathbf{Z}_t \circ \mathbf{M}_t, \mathbf{Z}_{t-1} \circ \mathbf{M}_{t-1} \qquad \text{masked observations} \tag{5.7}$$

$$\tilde{\mathbf{Z}}_t \triangleq w(\mathbf{Z}_{t-1}, \mathbf{x}_t) \qquad \text{warped observation} \tag{5.8}$$

$$c_\alpha(\mathbf{Z}_t; \mathbf{x}_t) \triangleq \mathbf{Z}_t \star \tilde{\mathbf{Z}}_t \qquad \text{correlation} \tag{5.9}$$

[2] w denotes a warp function such as bi-linear interpolation for example

where $\mathbf{Y}_t, \mathbf{Y}_{t-1} \in \mathbb{R}^{H \times W}$ are now used to denote raw measurements and $\mathbf{Z}_t, \mathbf{Z}_{t-1} \in \mathbb{R}^{H \times W}$ to denote radar measurements *after* the masking process. The likelihood in this case is written as before $p_\alpha(\mathbf{Z}_t|\mathbf{x}_t) \triangleq \frac{1}{C}\exp\{\beta c_\alpha(\mathbf{Z}_t;\mathbf{x}_t)\}$ where subscript α is now introduced to indicate conditioning on the network parameters α.

Setting the proposal distribution to the prior $\tilde{q}(\mathbf{x}_t) = p(\mathbf{x}_t)$ gives

$$\tilde{\omega}_s = \frac{\tilde{p}(\mathbf{x}_t)}{\tilde{q}(\mathbf{x}_t)} = \frac{p_\alpha(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{x}_t)} = \exp\{\beta c_\alpha(\mathbf{Z}_t;\mathbf{x}_s)\} \tag{5.10}$$

(considering $\tilde{p}(\mathbf{x}_t) \triangleq p_\alpha(\mathbf{Z}_t|\mathbf{x}_t)p(\mathbf{x}_t)$ as before) and the weights may be determined as

$$\omega_s = \frac{\exp\{\beta c_\alpha(\mathbf{Z}_t;\mathbf{x}_s)\}}{\sum_s \exp\{\beta c_\alpha(\mathbf{Z}_t;\mathbf{x}_s)\}} = \texttt{softmax}(\beta \boldsymbol{c})_s \tag{5.11}$$

where $\boldsymbol{c} \in \mathbb{R}^S$ such that $\boldsymbol{c}_s \triangleq c_\alpha(\mathbf{Z}_t;\mathbf{x}_s)$. Finally, if the proposal is chosen to generate a set of pose candidates over an evenly spaced grid then the using Eq. (5.10) and Eq. (5.11) to generate $\{(\mathbf{x}_s, \omega_s)\}_{s=1}^S$ and estimating the pose using Eq. (5.5) is exactly equivalent to the pose estimation approach proposed earlier (see Algorithm 1).

Quantifying Pose Uncertainty Alongside the optimum pose, the approach proposed earlier also estimates the pose uncertainty. Using the approach just derived, this can be interpreted as approximating the posterior $p(\mathbf{x}|\mathbf{Z})$ with a Gaussian distribution

$$p_\alpha(\mathbf{x}|\mathbf{Z}) \approx \texttt{Nor}(\mathbf{x}|\bar{\mathbf{x}}, \bar{\boldsymbol{\Sigma}}) \qquad \bar{\mathbf{x}} = \sum_s \omega^s \mathbf{x}^s \qquad \bar{\boldsymbol{\Sigma}} = \sum_s \omega^s \mathbf{x}^s \mathbf{x}^{s\top} - \bar{\mathbf{x}}\bar{\mathbf{x}}^\top \tag{5.12}$$

where the mean and co-variance are set to their maximum likelihood estimates and estimated using importance sampling.

If the true posterior is a multi-variate Gaussian, the squared Mahalanobis distance

$$d^2 = (\mathbf{x} - \bar{\mathbf{x}})^\top \bar{\boldsymbol{\Sigma}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \tag{5.13}$$

will be chi-squared distributed, $d^2 \sim \chi^2(3)$, (where 3 is the dimensionality of the pose $\mathbf{x} = [x, y, \theta]^\top$) and in this case $\mathbb{E}[d^2] = 3$. This result is used to evaluate the quality of the uncertainty and to provide a mechanism for choosing the temperature parameter β in the post-training tuning step used earlier in Section 4.4.

Possible Extensions Whilst the approach developed earlier corresponds to assuming the prior is uniform over a fixed grid, this consideration also extends to other priors, formed through motion models or information from other sensing modalities, for example. Considering a prior of the form $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ the approach is easily incorporated into the Bayes filtering paradigm, readily implemented using a particle filter, for example.

6

Fast-MbyM: Leveraging Translational Invariance of the Fourier Transform for Efficient And Accurate Radar Odometry

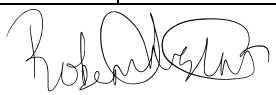
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

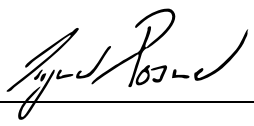
Title of Paper	Fast-MbyM: Leveraing Translational Invariance of the Fourier Transform For Efficient And Accurate Radar Odometry
Publication Status	□ Published X Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	R. Weston, M. Gadd, D. De Martini, P. Newman, I. Posner. "Fast-MbyM: Leveraing Translational Invariance of the Fourier Transform For Efficient And Accurate Radar Odometry". In: *International Conference On Robotics and Automation 2022 (ICRA Philadelphia 2022)*

Student Confirmation

Student Name:	Robert Weston		
Contribution to the Paper	Developed Approach, Ran Experiments, Wrote Manuscript		
Signature		Date	7/4/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Prof. Ingmar Posner		
Supervisor comments	I confirm that the candidate made the contributions specified above.		
Signature		Date	7/4/2022

arXiv:2203.00459v1 [cs.RO] 1 Mar 2022

Fast-MbyM: Leveraging Translational Invariance of the Fourier Transform for Efficient and Accurate Radar Odometry

Rob Weston*, Matthew Gadd†, Daniele De Martini†, Paul Newman†, and Ingmar Posner*
*Applied Artificial Intelligence Lab (A2I), †Mobile Robotics Group (MRG), University of Oxford
{robw, mattgadd, daniele, pnewman, ingmar}@robots.ox.ac.uk
https://github.com/applied-ai-lab/f-mbym

***Abstract*— Masking by Moving (MByM), provides robust and accurate radar odometry measurements through an exhaustive correlative search across discretised pose candidates. However, this dense search creates a significant computational bottleneck which hinders real-time performance when high-end GPUs are not available. Utilising the translational invariance of the Fourier Transform, in our approach, Fast Masking by Moving (f-MByM), we decouple the search for angle and translation. By maintaining end-to-end differentiability a neural network is used to mask scans and trained by supervising pose prediction directly. Training faster and with less memory, utilising a decoupled search allows f-MbyM to achieve significant run-time performance improvements on a CPU (168 %) and to run in real-time on embedded devices, in stark contrast to MbyM. Throughout, our approach remains accurate and competitive with the best radar odometry variants available in the literature – achieving an end-point drift of 2.01 % in translation and 6.3 deg /km on the *Oxford Radar RobotCar Dataset*.**

I. Introduction

In recent years, Radar Odometry (RO) has emerged as a valuable alternative to lidar and vision based approaches due to radar's robustness to adverse conditions and long sensing horizon. However, noise artefacts inherent in the sensor imaging process make this task challenging. The work of Cen and Newman [1] first demonstrated the potential of radar as an alternative to lidar and vision for this task and since then has sparked significant interest in RO.

Whilst sparse point-based RO methods such as [1], [2], [3], [4], [5], [6], [7] have shown significant promise, Barnes *et al.* [8] recently established the benefits that a dense approach brings to this problem setting. By masking radar observations using a DNN before adopting a traditional brute-force scan matching procedure, `mbym` *learns* a feature embedding explicitly optimised for RO. As robust and interpretable as a traditional scan matching procedure, `mbym` was able to significantly outperform the previous state of the art [1].

However, as our experiments demonstrate, `mbym` in its original incarnation is unable to run in *real-time* on a laptop at all but the smallest resolutions and not at all on an embedded device. The requirement for a high-end GPU for real-time performance represents a significant hindrance for deployment scenarios where the cost or power requirements of such hardware is prohibitive.

In this work we propose a number of modifications to the original `mbym` approach which result in significantly faster run-time performance, enabling real-time performance at higher resolutions on both CPUs and embedded devices. In particular, instead of performing a brute-force search over all possible combinations of translation and angle, we exploit properties of the Fourier Transform to search for the angle between the two scans *independent* of translation. By adopting this decoupled approach, we significantly reduce computation. Our approach, `f-mbym`, retains end-to-end differentiability and thus the use of a CNN to mask radar scans, learning a radar scan representation explicitly optimised for RO. `f-mbym`, is shown in Fig. 1. Like `mbym` our model is trained end-to-end in a supervised fashion. However, our modifications allow `f-mbym` to be trained much more rapidly and with much less memory.

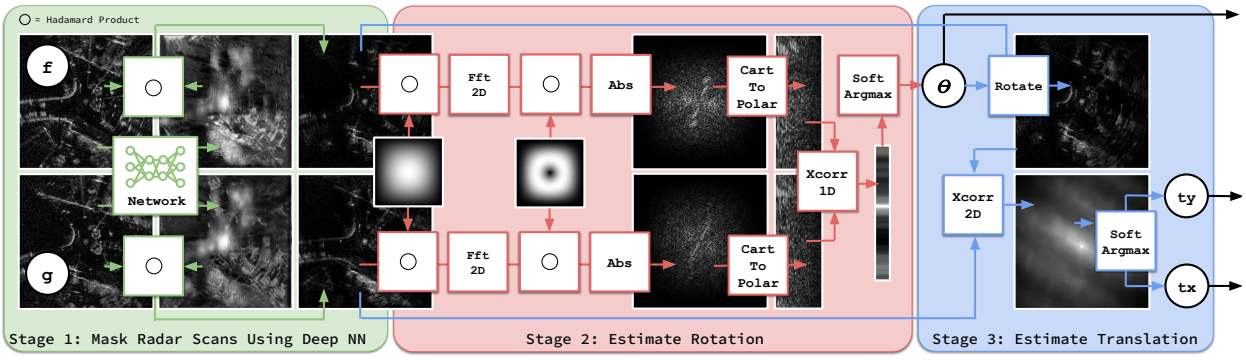


Fig. 1: Given radar scans $\mathbf{f}$ and $\mathbf{g}$ our system outputs the relative pose $[\theta, \mathtt{t_x}, \mathtt{t_y}]$ between them in three phases: (1) each radar scan is masked using a deep neural network; (2) the rotation θ is determined by maximising the correlation between the magnitude of their fourier transforms in polar co-ordinates; (3) the translation $[\mathtt{t_x}, \mathtt{t_y}]$ is determined by maximising the correlation between $\mathbf{f}$ and $\mathbf{g}$ rotated by the now known angle θ. Using our approach we are able to determine θ independently of $[\mathtt{t_x}, \mathtt{t_y}]$ allowing us to achieve real-time performance on a CPU and embedded devices. Crucially, this entire procedure is end-to-end differentiable allowing us to explicitly optimise our network for radar pose estimation.

By providing a greater run-time efficiency at higher resolutions our best performing real-time model achieves an end-point error of $2.01\,\%$ in translation and $6.3\,\mathrm{deg}\,/\mathrm{km}$ in rotation on the *Oxford Radar RobotCar Dataset* [9], outperforming the best real-time `mbym` model in accuracy whilst running 168% faster on a CPU and in real-time (at $6\,\mathrm{Hz}$) on a *Jetson* GPU. Our approach remains competitive with the current state-of the-art, point-based methods.

II. Related Work

In recent years the work of Cen *et al.* [1], [2] has demonstrated the potential of RO as an alternative to vision and lidar, sparking a significant resurgence in interest in RO. Cen and Newman [1] propose a global shape similarity metric to match features between scans whilst in their subsequent work [2] a gradient-based feature detector and a new graph matching strategy are shown to improve performance.

Since then several methods have been proposed [8], [6], [5], [3], [10], outperforming [1], [2] with gains attained through a combination of motion compensation [4], [5], [7], fault diagnosis and filtering [10], as well as new learnt [6], [5], [3] and rule-based [7] feature representations. In [3], as an alternative to hand-crafted feature extraction proposed in [1], Aldera *et al.* propose to extract temporally consistent radar feature points using a DNN. In this approach labels for stable points are generated by accumulating a histogram of points across time and over wide baselines. Instead, Barnes and Posner [6] extract and learn radar feature representations by supervising pose prediction directly. This results in a significant reduction in end-point error when compared to [2]. In [4] Burnett *et al.* find that motion compensating scans yields significant boosts in RO performance (when compared to [1]). Combining this with an unsupervised adaptation of [6], in [5], Burnett *et al.* are able to slightly outperform [6] without requiring ground truth odometry measurements to train their system. In an alternative and recently proposed approach [7] a robust point-to-line metric is used, in combination with motion compensation and estimation over a sliding window of past observations.

In contrast to the sparse methods mentioned above, Masking by Moving [8] adopts a dense approach; using a correlative scan matching procedure in combination with a learnt feature space supervised for pose prediction, the optimal pose is searched for across a dense grid of candidates. Through this approach `mbym` is able to outperform sparse variants [1], [2], [5], [4]. However, while a dense search results in excellent performance it comes with a significant computational cost. This cost may be offset when high-end modern graphical processing hardware is available as demonstrated by the timing results shown in [8], but means that `mbym` struggles to run online when the cost or power requirements of such hardware is prohibitive. The learnt element of `mbym` can also lead to geographical overfitting, where the model performs better in the areas it has been trained. In this work our aim is to tackle the former of these problems, noting that as larger scale and more varied radar odometry datasets become available models should become less prone to overfitting. Nonetheless, further investigation into combating geographical overfitting in the low data regime, remains an interesting area for future research.

Building upon [8] we also adopt a dense scan matching procedure with a learnt feature representation supervised directly for pose estimation. However, we propose to overcome the computational burden of the dense search by decoupling the search for angle and translation between scans, exploiting the translational invariance of the Fourier Transform [11]. This property alongside the scale invariance property of the Mellin Transform (MT) are combined to form the Fourier-Mellin Transform (FMT) [11]. The FMT has been widely exploited for image registration [12], [13] as well as for visual odometry [14], [15].

In the radar domain, Checchin and Gérossier [16] proposed to use the FMT for RO over a decade ago. More recently [17] proposes to use a similar approach in their RO system. In contrast, as the scale between scans is known, in our own work we rely on *only* the Fourier translation property. In contrast to [16], [17] we propose to mask radar observations using a CNN. Using a differentiable implementation of the decoupled scan matching procedure allows us to learn a radar feature representation supervised for pose prediction without resorting to hand-crafted filtering or feature extraction and results in superior performance.

III. Approach

We begin by formulating the problem (Sec. III-A) and discuss the limitations of a naïve correlative scan matching procedure (Sec. III-B). Next we show how by using properties of the Fourier Transform we are able to more efficiently search for the optimum pose by decoupling the search for rotation and translation (Sec. III-C). In Sec. III-D we propose a discrete and differentiable implementation. Finally, to improve performance the radar scans are filtered using a Deep Neural Network (Sec. III-E) which is trained explicitly for pose prediction (leveraging the differentiability of our scan matching implementation).

A. Problem Formulation

Let the signals $f(\mathbf{x}) \in \mathbb{R}$ and $g(\mathbf{x}') \in \mathbb{R}$ denote radar power measurements in two coordinate systems $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$ related by a rigid-body transformation $[\mathtt{R}^*|\mathtt{t}^*] \in \mathbb{SE}(2)$

$$\mathbf{x}' = \mathtt{R}^*\mathbf{x} + \mathtt{t}^* \tag{1}$$

where $\mathtt{R}^* = \mathtt{R}^*(\theta^*) \in \mathbb{SO}(2)$ is a 2D rotation matrix parameterised by yaw $\theta^* \in [0, 2\pi]$ and $\mathtt{t}^* = [\mathtt{t_x}^*, \mathtt{t_y}^*]^\top \in \mathbb{R}^2$ is a translational offset. In this case the radar power measurements $f(\mathbf{x})$ and $g(\mathbf{x}')$ are related as $f(\mathbf{x}) \approx g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*)$, where this relationship is only approximate due to appearance change between the two frames. The aim of our approach is to estimate the pose $[\mathtt{R}^*|\mathtt{t}^*] \in \mathbb{SE}(2)$ between the two coordinate systems, given access to $f(\mathbf{x})$ and $g(\mathbf{x}')$.

B. Correlative Scan Matching

In a *correlative scan matching* approach (such as in [8]) the optimum pose $[\mathtt{R}^*|\mathtt{t}^*]$ is found by maximising the correlation between the two scans

$$\mathtt{R}^*, {\mathtt{t_x}}^*, {\mathtt{t_y}}^* = \mathrm{argmax}_{\mathtt{R},\mathtt{t_x},\mathtt{t_y}} (f \star g)(\mathtt{R}, \mathtt{t_x}, \mathtt{t_y}) \tag{2}$$

where $(f \star g)(\mathtt{R}, \mathtt{t_x}, \mathtt{t_y})$ is the *cross-correlation* operation:

$$(f \star g)(\mathtt{R}, \mathtt{t_x}, \mathtt{t_y}) = \int_{\mathbb{R}^2} f(\mathbf{x}) g(\mathtt{R}\mathbf{x} + [\mathtt{t_x}, \mathtt{t_y}]^\top) d\mathbf{x} \ . \tag{3}$$

The optimum pose is found through a brute force approach partitioning the space $\mathtt{R}, \mathtt{t} \in \mathbb{SO}(2) \times \mathbb{R}^2$ into discrete and evenly spaced pose candidates $\mathtt{R}, \mathtt{t_x}, \mathtt{t_y} \in \{\mathtt{R}_i\}_{i=1}^{n_\theta} \times \{\mathtt{x}_i\}_{i=1}^{n_x} \times \{\mathtt{y}_i\}_{i=1}^{n_y}$ and choosing the pose that maximises correlation between f and g. However, searching over every possible combination of $\mathtt{R}^*$, $\mathtt{t_x}^*, \mathtt{t_y}^*$ creates a significant computational bottleneck hindering real-time performance when high-end compute is not available.

C. Exploiting Translational Invariance of the Fourier Transform for Efficient Pose Estimation

We therefore utilise properties of the Fourier Transform to search for $\mathtt{R} \in \{\mathtt{R}_i\}_{i=1}^{n_\theta}$ *independently* of $\mathtt{t} \in \{\mathtt{x}_i\}_{i=1}^{n_x} \times \{\mathtt{y}_i\}_{i=1}^{n_y}$. This is the key to the efficiency of our approach. With the radar signals related as $f(\mathbf{x}) = g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*)$ their Fourier Transforms are related as

$$\hat{f}(\mathbf{u}) := \mathcal{F}[f(\mathbf{x})] := a_f(\mathbf{u}) e^{j\phi_f(\mathbf{u})} \tag{4}$$

$$\hat{g}(\mathbf{u}') := \mathcal{F}[g(\mathbf{x}')] := a_g(\mathbf{u}') e^{j\phi_g(\mathbf{u}')} \tag{5}$$

$$\hat{f}(\mathbf{u}) = \hat{g}(\mathtt{R}^*\mathbf{u}) e^{2\pi j \mathtt{t}^{*\top} \mathtt{R}^* \mathbf{u}} \tag{6}$$

(see proof in Sec. VII) where $\mathcal{F} : \mathbb{R} \to \mathbb{C}$ denotes the one-sided 2D Fourier Transform and $\mathbf{u} = [u_1, u_2]^\top \in \mathbb{R}^2$ is the spatial frequency. Here, their magnitudes $a_f(\mathbf{u}) := |\hat{f}(\mathbf{u})|$, $a_g(\mathbf{u}') := |\hat{g}(\mathbf{u}')|$ differ *only* by a rotation, $a_f(\mathbf{u}) = a_g(\mathtt{R}^*\mathbf{u})$ and are *independent* of $\mathtt{t}^*$[1]. Exploiting this result, an efficient algorithm for determining the optimum pose $[\mathtt{R}^*|\mathtt{t}^*]$ emerges:

1) Determine $\mathtt{R}^*$: Considering $a_f(\mathbf{u})$ and $a_g(\mathbf{u}')$ in polar coordinates $\tilde{a}_f(\boldsymbol{\omega})$ and $\tilde{a}_g(\boldsymbol{\omega}')$, where $\boldsymbol{\omega}(u_1, u_2) = \left[\tan^{-1}(\frac{u_2}{u_1}), \sqrt{u_1^2 + u_2^2}\right]$ is the polar representation of the 2D spatial frequency plane, the rotation between $\mathbf{u}$ and $\mathbf{u}'$ will manifest as a translation between $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$: the angle θ between the two signals can therefore be recovered as,

$$\theta^* = \text{argmax}_\theta(\tilde{a}_f \star \tilde{a}_g)(\mathtt{I}, \theta, 0) \tag{7}$$

where $\mathtt{I} = \text{diag}([1, 1])$ and $\text{argmax}_\theta(\tilde{a}_f \star \tilde{a}_g)$ is the correlation as per Eq. (3) between the magnitudes of the two signals after mapping to polar coordinates.

2) Determine $\mathtt{t}^*$: Once $\mathtt{R}^* = \mathtt{R}^*(\theta^*)$ is known we are able to recover $\mathtt{t}^* = [\mathtt{t_x}^*, \mathtt{t_y}^*]^\top$ as,

$$\mathtt{t_x}^*, \mathtt{t_y}^* = \text{argmax}_{\mathtt{t_x}, \mathtt{t_y}}(f \star g)(\mathtt{R}^*, \mathtt{t_x}, \mathtt{t_y}) \tag{8}$$

where g is rotated by the rotation solved for in the previous step. Compared to the naïve approach, where this last step must be performed for every yaw candidate $\mathtt{R} \in \{\mathtt{R}_i\}_{i=1}^{n_\theta}$, this reduces computation by a factor of n_θ.

D. Implementation

Whilst the approach so far was developed for continuous signals $f(\mathbf{x})$ and $g(\mathbf{x}')$ in reality we only have access to discrete sets of power measurements $\mathbf{f} \in \mathbb{R}^{n_x \times n_y}$ and $\mathbf{g} \in \mathbb{R}^{n_x \times n_y}$ measured at locations $\mathbf{x}, \mathbf{x}' \in \{\mathtt{x}_i\}_{i=1}^{n_x} \times \{\mathtt{y}_i\}_{i=1}^{n_y}$ (assumed to fall over an evenly spaced grid). Alg. 1 therefore gives a discrete approximation to the approach developed up to this point. The function $ScanMatch$ takes as input $\mathbf{f}$ and $\mathbf{g}$ and returns the estimated pose $[\mathtt{R}, \mathtt{t_x}, \mathtt{t_y}]$. A diagram of our approach is found in Fig. 1.

[1]This can intuitively be understood by noting that translating the original 2D signal does not change the overall frequency content, merely shifts it to a new location (resulting in a phase shift between the two signals)

Algorithm 1: Fourier Scan Matching Procedure

1 **function** *ScanMatch*($\mathbf{f}$, $\mathbf{g}$, $n_\theta = 733$, $\delta_\theta = \pi/733$, $T_\theta = 2$, $n_{xy} = 255$, $\delta_{xy} = 0.4$, $T_{xy} = 1$)**:**
  `/* Determine the pose` $[\mathtt{R}^*, \mathtt{t}^*]$ `between the two scans` $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{n_{xy} \times n_{xy}}$ `*/`
2 $\{\theta_k\} = Linspace(-\frac{1}{2}\delta_\theta n_\theta, \frac{1}{2}\delta_\theta n_\theta, n_\theta)$
3 $\{\mathbf{x}_i\} = Linspace(-\frac{1}{2}\delta_{xy} n_{xy}, \frac{1}{2}\delta_{xy} n_{xy}, n_{xy})$
4 $\{\mathbf{x}_j\} = Linspace(-\frac{1}{2}\delta_{xy} n_{xy}, \frac{1}{2}\delta_{xy} n_{xy}, n_{xy})$
  `/* Stage 1: Determine` θ^* `*/`
5 $\mathbf{h}_{Hann} = HanningFilter(Shape(\mathbf{f}))$
6 $\mathbf{f}, \mathbf{g} = \mathbf{h}_{Hann} \circ \mathbf{f}, \mathbf{h}_{Hann} \circ \mathbf{g}$
7 $\hat{\mathbf{f}}, \hat{\mathbf{g}} = FFT2d(\mathbf{f}), FFT2d(\mathbf{g})$
8 $\mathbf{h}_{Band} = BandPassFilter(Shape(\mathbf{f}))$
9 $\hat{\mathbf{f}}, \hat{\mathbf{g}} = \mathbf{h}_{Band} \circ \hat{\mathbf{f}}, \mathbf{h}_{Band} \circ \hat{\mathbf{g}}$
10 $\mathbf{a}_f, \mathbf{a}_g = Abs(\hat{\mathbf{f}}), Abs(\hat{\mathbf{g}})$
11 $\tilde{\mathbf{a}}_f, \tilde{\mathbf{a}}_g = Cart2Pol(\mathbf{a}_f), Cart2Pol(\mathbf{a}_g)$
12 $\tilde{\mathbf{a}}_f, \tilde{\mathbf{a}}_g = WrapPad(\mathbf{a}_f), WrapPad(\mathbf{a}_g)$
13 $\mathbf{c}_{\theta r} = iFFT2d(FFT2d(\tilde{\mathbf{a}}_f) \circ FFT2d(\tilde{\mathbf{a}}_g))$
14 $\mathbf{c}_\theta = Mean(\mathbf{c}_{\theta r}, \text{dim} = \text{'r'})$
15 $\theta = SoftArgMax(\mathbf{c}_\theta, T_\theta, \{\theta_k\})$
  `/* Determine` $[\mathtt{t_x}^*, \mathtt{t_y}^*]$ `*/`
16 $\mathbf{g}' = Rotate(\mathbf{g}, \theta^*)$
17 $\mathbf{f}, \mathbf{g}' = ZeroPad(\mathbf{f}), ZeroPad(\mathbf{g}')$
18 $\mathbf{c}_{xy} = iFFT2d(FFT2d(\mathbf{f}) \circ FFT2d(\mathbf{g}'))$
19 $\mathtt{t_x}', \mathtt{t_y}' = SoftArgMax(\mathbf{c}_{xy}, T_{xy}, \{\mathbf{x}_i\} \times \{\mathtt{y}_j\})$
20 $\mathtt{R}' = BuildSO2(-\theta)$
21 $\mathtt{t_x}, \mathtt{t_y} = MatMul(\mathtt{R}', [\mathtt{t_x}', \mathtt{t_y}'])$
22 **return** $\theta, \mathtt{t_x}, \mathtt{t_y}$

The 2D correlation operator defined in Eq. (3) is approximated in Alg. 1 by its discrete counterpart and is implemented as a multiplication in the Fourier domain using the highly efficient $FFT2d$ and inverse $iFFT2d$ (lines 13 and 18). The argmax operation in Eqs. (7) and (8) is replaced with a soft approximation $SoftArgMax$ in lines 15 and 19 to ensure that the scan matching procedure maintains end-to-end differentiability. Here, a temperature controlled softmax is applied to the 2D correlation scores before a weighted sum is performed over its coordinates. This property will be exploited in Sec. III-E to *learn* a radar embedding optimised for pose prediction. It was found that applying specific filtering and padding strategies was important to ensure correct operation. A Hanning filter [18] is applied before performing the 2D FFT of $\mathbf{f}$ and $\mathbf{g}$ to reduce boundary artefacts and a band-pass filter was applied thereafter to reduce the impact of uninformative low and high frequencies. As the angular dimension in polar-coordinates is periodic, applying circular padding to the power spectra along the angular dimension ($WrapPad$ in Alg. 1) significantly reduces boundary artefacts; on the translational directions, instead, we padded the spectra with zeros ($ZeroPad$ in Alg. 1). The functions $Rotate$ and $Cart2Pol$ are implemented using bilinear interpolation in a similar approach to [19]. The number of range readings is set to n_{xy}.

E. Learnt Radar Embeddings For Improved Odometry

Central to the success of our approach was an assumption that $f(\mathbf{x}) \approx g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*)$. Of course there are several reasons why this condition might not hold in practice: dynamic objects, motion blur, occlusion, and noise all result in a

power field that fluctuates from one time-step to the next. To counteract this, in a similar approach to [8], we propose to mask the radar power returns using a neural network h_α to filter the radar scans before scan matching:

$$[\mathbf{m}_f, \mathbf{m}_g] = h_\alpha(\mathbf{f}, \mathbf{g}) \tag{9}$$

$$\tilde{\mathbf{f}} = \mathbf{f} \circ \mathbf{m}_f \quad \text{and} \quad \tilde{\mathbf{g}} = \mathbf{g} \circ \mathbf{m}_g \tag{10}$$

$$[\theta, \mathtt{t_x}, \mathtt{t_y}] = ScanMatch(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) \tag{11}$$

where $\circ$ denotes the Hadamard product and $ScanMatch$ is defined in Alg. 1. Given a dataset $\mathcal{D} = \{(\mathbf{f}, \mathbf{g}, \theta^*, \mathtt{t_x}^*, \mathtt{t_y}^*)_n\}_{n=1}^N$ the network parameters α are found by minimising:

$$\mathcal{L}(\alpha) = \mathbb{E}_\mathcal{D}\left\{|\theta^* - \theta|_1 + |\mathtt{t_x}^* - \mathtt{t_x}|_1 + |\mathtt{t_y}^* - \mathtt{t_y}|_1\right\} \tag{12}$$

Note that instead of minimising the Mean Square Error (MSE) as in [8] we consider minimising the Mean Absolute Error (MAE) which is less sensitive to outliers. The network architecture for h_α is discussed further in Sec. IV-B.

IV. Experimental Setup

A. Datasets

We evaluate our approach using the *Oxford Radar RobotCar Dataset* [9] featuring a CTS350-X Navtech FMCW radar with 4 Hz scan rate which defines our requirement for real-time. In a similar approach to [8], [5], [6] we partition the data in *time* rather than geography. Tab. I details the specific train, validation and test sets used.

Split	Pattern	Examples	Percentage
Train	`2019-01-1[1-8]*`	197900	85%
Validate	`2019-01-10-12-32-52*`	8617	4%
Test	`2019-01-10-1[24]*`	25707	11%

TABLE I: All *Oxford Radar RobotCar Dataset* loops which match the split pattern are used for each split.

B. Network Architecture And Training

As our primary benchmark we compare against the `mbym` model proposed in [8] which we train from scratch using the splits from Sec. IV-A. To ensure a fair comparison, the masking network architecture and masking strategy are kept consistent for both `mbym` and `f-mbym` (see Tab. II).

In	Skip	Down	Conv c_i	Conv c_o	Norm	Act	Conv c_i	Conv c_o	Norm	Act	Up	Out
Encoder												
$\mathbf{f}, \mathbf{g}$	–	–	2	8	BN	Relu	8	8	BN	Relu	-	$\mathbf{h}_1$
$\mathbf{h}_1$	–	MP	8	16	BN	Relu	16	16	BN	Relu	-	$\mathbf{h}_2$
$\mathbf{h}_2$	–	MP	16	32	BN	Relu	32	32	BN	Relu	-	$\mathbf{h}_3$
$\mathbf{h}_3$	–	MP	32	64	BN	Relu	64	64	BN	Relu	-	$\mathbf{h}_4$
$\mathbf{h}_4$	–	MP	64	128	BN	Relu	128	128	BN	Relu	-	$\mathbf{h}_5$
$\mathbf{h}_5$	–	MP	128	256	BN	Relu	256	256	BN	Relu	BL	$\mathbf{h}_6$
Decoder												
$\mathbf{h}_6$	$\mathbf{h}_5$	–	384	128	BN	Relu	128	128	BN	Relu	BL	$\mathbf{h}_7$
$\mathbf{h}_7$	$\mathbf{h}_4$	–	192	64	BN	Relu	64	64	BN	Relu	BL	$\mathbf{h}_8$
$\mathbf{h}_8$	$\mathbf{h}_3$	–	96	32	BN	Relu	32	32	BN	Relu	BL	$\mathbf{h}_9$
$\mathbf{h}_9$	$\mathbf{h}_2$	–	48	16	BN	Relu	16	16	BN	Relu	BL	$\mathbf{h}_{10}$
$\mathbf{h}_{10}$	$\mathbf{h}_1$	–	24	8	BN	Relu	8	8	BN	Relu	BL	$\mathbf{h}_{11}$
$\mathbf{h}_{11}$	–	–	8	2		Sigmoid	–	–	–	–	–	$\mathbf{m}_{fg}$

TABLE II: The network architecture h_α used to generate masks $\mathbf{m}_f, \mathbf{m}_g$ from radar scans $\mathbf{f}, \mathbf{g}$ in Sec. III-E. `MP` is max-pool, `BN` is batch-norm and `BL` is for bi-linear upsampling.

The scans $\mathbf{f}$ and $\mathbf{g}$ are concatenated to form a two channel tensor and passed to our network as a single input (adopting the best-performing *dual* method from [8]). A U-Net architecture [20] is used to increase the feature dimension and decrease the spatial dimension through the repeated application of convolutions and max-pooling before this process is reversed through bi-linear up-sampling (`BL`) and convolutions [21]. Information is allowed to flow from the encoder to the decoder using skip connections which are concatenated with the input feature map at each decoder level. Batch Norm (`BN`) and ReLu activation (`Relu`) are applied after each convolution. The masks $\mathbf{m}_f, \mathbf{m}_g$ output by our network are generated using a single convolution with a sigmoid activation.

As there is an intrinsic balance between run-time performance and input resolution with reference to Alg. 1 input parameters, we train both models at three resolutions $\delta_{xy} \in \{0.8, 0.4, 0.2\}$ corresponding to input sizes $n_{xy} \in \{127, 255, 511\}$, similarly to [8], with a batch size of 128, 64 and 32 respectively. All networks are trained minimising the loss of Eq. (12) for 80 epochs on the training set with no augmentation applied to the input data. Translational drift (see Sec. IV-C) is calculated on the validation set at each epoch and the model with the smallest drift over all epochs is selected, before the accuracy is calculated over the *test* set. We experimented with learning rates 1×10^{-3} and 1×10^{-4} using the Adam optimiser [22], finding that all models perform best when training with a learning rate of 1×10^{-4} with the exception of `f-mbym@511` where 1×10^{-3} was slightly better. For completeness we also include results which are available from the original implementation and splits, quoting directly from [8]. We find that our implementation of `mbym` outperforms the original as presented in [8] as shown in Tab. IIIb. We attribute this to our introduction of batch-norms after every convolution, experimenting with slightly different resolutions ($127, 255, 511$ vs $125, 251, 501$) as well as a different training objective ($L1$ as opposed $L2$). These observations may be useful when re-implementing our work and that of [8].

C. Metrics

To assess odometry accuracy we follow the KITTI odometry benchmark [23]. For each 100 m segment of up to 800 m long trajectories, we calculate the average residual translational and angular error for every test set sequence, normalising by the distance travelled. The performance across each segment and over all trajectories is then averaged to give us our primary measure of success.

As a core objective of this work, we also provide timing statistics using both a laptop without GPU as well as an embedded device with limited graphics capability. These test beds include a *Lenovo ThinkPad* with Intel Core i7 2.9 GHz processor and 8 GB RAM and a *NVIDIA Jetson Nano* with a Quad-Core ARM Cortext-A57 1.42 GHz processor, 128 CUDA cores (472 GFLOPS), and 4 GB RAM. During *ThinkPad* and *Jetson* tests, timing is measured by passing through the network tensors of batch size 1 which are populated by noise. For *Jetson*, we use event profiling provided by PyTorch/CUDA, while for *ThinkPad*, we use the standard Python library. All timing statistics stated are

calculated by averaging between 2000 and 10 000 forward passes. We discard results from an initial "burn-in" of 50 to 100 steps in order to let computation stabilise.

D. Baselines

As our primary benchmark we compare our approach, `f-mbym`, against `mbym`, as per [8]. Both models share the same masking network architecture and training setup (Sec. IV-B) and differ in how they solve for the pose (see Sec. III). We also include results for `mbym` and `f-mbym` without masking, denoted as `raw` and `f-raw` respectively. This allows us to further investigate the benefits that adopting a decoupled search brings to run-time performance. Comparing `f-raw` to `f-mbym` also allows us to compare our approach to a conventional decoupled procedure *without* a learnt radar feature space, similar to [17].

V. Results

In Sec. V-A, Sec. V-B and Sec. V-C we respectively investigate what impact a decoupled search has on run-time efficiency, real-time performance, and training. In Sec. V-D we compare our approach with and without a masking network. Finally, in Sec. V-E we investigate how our approach fairs in comparison to several sparse point-based baselines.

A. Run-Time Performance

Comparing the run time efficiency of `f-mbym` to `mbym` in Tab. IIIa the benefits of adopting a decoupled approach becomes clear; considering a like-for-like comparison at each resolution we are able to achieve speedups of 372 % to 800 % on a CPU and 424 % to 470 % on the *Jetson* (it is worth noting that the memory footprint of the 511 resolution `mbym` means it is unable to run on the *Jetson* entirely).

Further insights into run-time efficiency are gained by considering the efficiency of the brute-force and decoupled scan matching procedure in isolation from the time taken to mask each radar scan. The former is determined by measuring the run-time performance of `mbym` and `f-mbym` operating on *raw* radar scan (without masking) and is given by `raw` and `f-raw` in Tab. IIIa. The latter is provided by measuring the time it takes for a forward pass through the masking network and is given by `mask`. Considering `raw` it becomes clear that the brute-force search for $\mathtt{t_x}, \mathtt{t_y}, \theta$ is a significant computational bottleneck; even without masking only the lowest resolution model is able to run in real-time (>4 Hz, the radar scan rate) on the *ThinkPad* and not at all on the *Jetson*. In contrast the majority of `f-mbym` models are currently throttled by the forward pass through the network, as can be seen by comparing `mask` to `f-raw` (where in the majority of cases the time taken for masking each radar scan is greater than that spent on the scan matching procedure).

B. Real-Time Odometry Accuracy

As our approach runs faster we are able to use a model at a higher resolution whilst still maintaining real-time operation. Considering Tab. IIIb, we note that whilst increasing the resolution from 127 to 255 results in a significant reduction in end-point error we experience only a marginal reduction in error when increasing from a resolution 255 to 511 (e.g. 2.01 % to 2.00 %). As `f-mbym@255` runs significantly faster than `f-mbym@511` we therefore consider `f-mbym@255` as our best performing model.

	Timing Results					
	Think Pad (Hz)			**Jetson (Hz)**		
	127	255	511	127	255	511
Baseline						
`mask`	96.2	33.4	7.6	24.7	8.7	2.4
`raw`	14.3	3.7	0.8	6.6	2.1	-[1]
`f-raw`	83.2	58.2	21.0	28.3	22.3	9.4
`mbym`	12.2	3.4	0.7	3.7	1.4	-[1]
Ours						
`f-mbym`	45.4	20.6	5.6	15.7	6.6	1.9

(a)

	Kitti Odometry Error					
	127		255		511	
	Tra	Rot	Tra	Rot	Tra	Rot
Baseline						
`raw`	9.55	30.93	6.39	20.87	5.13	17.39
`f-raw`	9.58	29.60	8.46	27.75	7.95	26.86
`mbym` [8]	2.70	7.6	1.80	4.7	1.16	3.0
`mbym`	2.15	6.46	1.36	3.98	-[2]	-[2]
Ours						
`f-mbym`	2.77	8.74	2.01	6.3	2.00	6.3

(b)

TABLE III: Timing results (a) and Kitti Odometry Metrics (b). Timing results are in Hz while translational (Tra) and rotational errors (Rot) are in % and deg/km respectively. [1]Failed to run entirely on the Jetson. [2]Due to training time constraints and resource limitations values for `mbym@511` are not reported for our own re-implementation as the run-time performance of this model fell significantly below real-time as shown in Tab. IIIa (see [8] for estimate).

On the *ThinkPad*, `f-mbym@255` outperforms the best performing (and only) real-time `mbym` model `mbym@127` in terms of end-point error (2.01 %, 6.3 deg/km vs. 2.14 %, 6.4 deg/km) whilst running 168 % faster. For *Jetson* tests `f-mbym@255` is still able to run in real-time at 6.6 Hz. This is in stark contrast to `mbym` which is unable to achieve *real-time* performance at any of the tested resolutions.

C. Training Comparisons

By adopting a decoupled search for angle and translation we are able to train significantly faster and with much less memory. We average the time for each training step (excluding data loading) for `mbym` and `f-mbym` running on 255 resolution inputs across an epoch. This process is repeated, doubling the batch size each time, until a *12GB Nvidia Titan X GPU* runs out of memory. The results are shown in Fig. 2. Whilst `mbym` is only able to fit a batch size of 4 into memory, `f-mbym` manages 64. We also find that a training step for `f-mbym` is $\sim 4-7$ times faster than for `mbym` (a like-for-like comparison at each batch size).

D. Masking

We now compare the performance of our approach with (`f-mbym`) and without (`f-raw`) the masking network. Comparing the odometry accuracy (Tab. IIIb) vs run-time perfor-

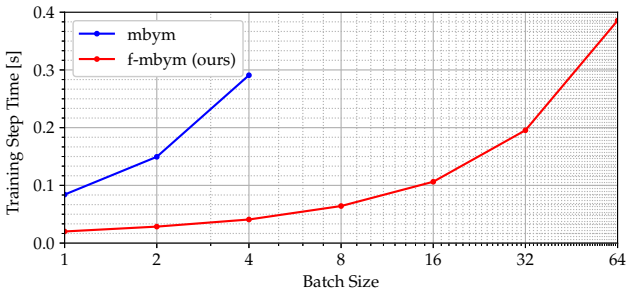


Fig. 2: Training step time comparison. Note that batch size is displayed using a log scale.

mance (Tab. IIIa) of each method it is clear that increasing odometry accuracy is worth the added penalty to run-time performance. In the majority of cases `f-mbym` is still able to run in real-time whilst increasing odometry accuracy by between $345\,\%$ to $390\,\%$ across each resolution. We posit that conventional decoupled search approaches, as in [17] could experience similar boosts in performance by adopting a learnt feature representation as in our approach.

E. Comparison To Sparse Point Based Methods

Finally, we compare our approach to several existing point-based RO systems on the *Oxford Radar RobotCar Dataset* [9], including: Cen RO [1], MC-RANSAC [4], HERO [5], Under The Radar [6], CFEAR [7]. For direct comparison we re-train our method using the splits from [6], [5]. As shown in Tab. IV we perform competitively with other approaches. We outperform Cen RO and MC-RANSAC by a significant margin. We also slightly outperform Under the Radar and HERO in rotational error. Only, CFEAR outperforms us in both translational and rotational error.

Method	**Type**	**Kitti Odometry Error** Tra (%)	Rot (deg/km)
Sparse Point-Based			
Cen RO [1]	classical	3.7168	9.50
MC-RANSAC [4]	classical	3.3190	10.93
Under The Radar [6]	supervised	2.0583	6.70
HERO [5]	unsupervised	1.9879	6.52
CFEAR [7]	classical	1.7600	5.00
Dense			
`mbym`[8]	supervised	*1.1600*	*3.00*
`f-mbym` (ours)	supervised	2.0597	6.269

TABLE IV: Comparison to other recent RO methods.

VI. Conclusion

In contrast to the brute force search over all possible combinations of translation and angle proposed in `mbym` [8], we propose to decouple the search for angle and translation, exploiting the Fourier Transform's invariance to translation. Doing so allows our approach to be trained faster and with less memory as well as to run significantly faster at inference time. By providing a greater run-time efficiency at higher resolutions our best performing real-time model achieves an end-point error of $2.01\,\%$ in translation and $6.3\,\mathrm{deg/km}$, outperforming the best real-time Masking by Moving model in accuracy whilst running 168% faster on a CPU and in real-time (at $6\,\mathrm{Hz}$) on a *Jetson* GPU. Our approach is competitive with the current state of the art achieved by sparse, point-based methods, challenging the conventional wisdom that a sparse point-based method is necessary for real-time performance.

As per Sec. V-A the run-time performance of our approach is currently limited by the time taken to mask each radar scan using a neural network. We also note that whilst our model achieves more accurate real-time performance in comparison to [8] when considering a like-for-like comparison at each resolution a significant gap exists in odometry accuracy. Closing this gap further could allow a dense method to surpass the performance of a sparse method whilst running in real-time. Investigating whether this is achievable with the modifications to the proposed formulation alongside faster masking strategies constitute interesting areas for future research. Finally, the decoupled search developed in our approach, could also be used to efficiently search for larger rotations, and so utilised for metric localisation where the rotational offset can be arbitrary.

VII. Appendix

As the affine transformation property of the Fourier Transform (FT) in Eq. (6) is crucial to this work and the original description by Bracewell [24] is not readily available, we derive it here again for completeness, starting with the definition of the 2D FT

$$\hat{g}(\mathbf{u}') = \int_{\mathbb{R}^2} g(\mathbf{x}') e^{-2\pi j \mathbf{u}'^\top \mathbf{x}'} d\mathbf{x}' \tag{13}$$

$$= \int_{\mathbb{R}^2} g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*) e^{-2\pi j \mathbf{u}'^\top (\mathtt{R}^*\mathbf{x} + \mathtt{t}^*)} d\mathbf{x} \tag{14}$$

$$= e^{-2\pi j \mathbf{u}'^\top \mathtt{t}^*} \int_{\mathbb{R}^2} g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*) e^{-2\pi j \mathbf{u}'^\top \mathtt{R}^* \mathbf{x}} d\mathbf{x} \tag{15}$$

$$= e^{-2\pi j (\mathtt{R}^*\mathbf{u})^\top \mathtt{t}^*} \int_{\mathbb{R}^2} f(\mathbf{x}) e^{-2\pi j \mathbf{u}^\top \mathbf{x}} d\mathbf{x} \tag{16}$$

$$= e^{-2\pi j \mathtt{t}^{*\top} \mathtt{R}^* \mathbf{u}} \hat{f}(\mathbf{u}) \tag{17}$$

Eq. (14) follows from a change of variables $\mathbf{x}' = \mathtt{R}^*\mathbf{x} + \mathtt{t}^*$ noting $d\mathbf{x}' = |\mathtt{R}^*| d\mathbf{x} = d\mathbf{x}$ and Eq. (15) by expanding the exponent and from the linearity of the Fourier transform. Eq. (16) follows by defining $\mathbf{u}' = \mathtt{R}^*\mathbf{u}$ and substituting $f(\mathbf{x}) = g(\mathtt{R}^*\mathbf{x} + \mathtt{t}^*)$ as in Sec. III-A. Finally, Eq. (17) follows from the definition of the 2D Fourier transform. Substituting $\mathbf{u}' = \mathtt{R}^*\mathbf{u}$ and rearranging terms finally gives Eq. (6):

$$\hat{f}(\mathbf{u}) = \hat{g}(\mathtt{R}^*\mathbf{u}) e^{2\pi j \mathtt{t}^{*\top} \mathtt{R}^* \mathbf{u}} \tag{18}$$

Acknowledgments

This work was supported by EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1) as well as by the Assuring Autonomy International Programme, a partnership between Lloyd's Register Foundation and the University of York. The authors would like to acknowledge the use of Hartree Centre resources and the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work `http://dx.doi.org/10.5281/zenodo.22558`. We gratefully acknowledge our partners at Navtech Radar and the support of Scan UK in this research.

REFERENCES

[1] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6045–6052, IEEE, 2018.

[2] S. H. Cen and P. Newman, "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.

[3] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "Fast radar motion estimation with a learnt focus of attention using weak supervision," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1190–1196, IEEE, 2019.

[4] K. Burnett, A. P. Schoellig, and T. D. Barfoot, "Do we need to compensate for motion distortion and doppler effects in spinning radar navigation?," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 771–778, 2021.

[5] K. Burnett, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Radar odometry combining probabilistic estimation and unsupervised feature learning," in *Robotics: Science and Systems*, 2021.

[6] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9484–9490, IEEE, 2020.

[7] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Cfear radarodometry-conservative filtering for efficient and accurate radar odometry," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5462–5469, IEEE, 2021.

[8] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," *arXiv preprint arXiv:1909.03752*, 2019.

[9] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[10] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "What could go wrong? introspective radar odometry in challenging environments," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2835–2842, IEEE, 2019.

[11] J. S. Lim, "Two-dimensional signal and image processing," *Englewood Cliffs*, 1990.

[12] D. Casasent and D. Psaltis, "Position, rotation, and scale invariant optical correlation," *Applied optics*, vol. 15, no. 7, pp. 1795–1799, 1976.

[13] X. Guo, Z. Xu, Y. Lu, and Y. Pang, "An application of fourier-mellin transform in image registration," in *The Fifth International Conference on Computer and Information Technology (CIT'05)*, pp. 619–623, 2005.

[14] T. Kazik and A. H. Göktoğan, "Visual odometry based on the fourier-mellin transform for a rover using a monocular ground-facing camera," in *2011 IEEE International Conference on Mechatronics*, pp. 469–474, IEEE, 2011.

[15] H. T. Ho and R. Goecke, "Optical flow estimation using fourier mellin transform," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[16] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine, "Radar scan matching slam using the fourier-mellin transform," in *Field and Service Robotics*, pp. 151–161, Springer, 2010.

[17] Y. S. Park, Y.-S. Shin, and A. Kim, "Pharao: Direct radar odometry using phase correlation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2617–2623, IEEE, 2020.

[18] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[19] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[21] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[24] R. Bracewell, K.-Y. Chang, A. Jha, and Y.-H. Wang, "Affine theorem for two-dimensional fourier transform," *Electronics Letters*, vol. 29, no. 3, pp. 304–304, 1993.

7 Discussion

7.1 Contributions

This thesis has explored the potential of deep data-driven approaches applied to radar across a range of tasks in robotics. To counter the lack of high-quality and large-scale datasets currently available, self-supervised approaches were presented as a promising alternative. Probabilistic approaches were also identified as important, for managing the natural uncertainty that arises when working with radar sensor measurements as a result of radar's complex image formation process. With these motivations in mind, in chapters 3 to 6 specific solutions were developed for inverse sensor modelling, simulation and odometry.

Inverse Sensor Modelling In chapter 3 a deep and data-driven approach for inverse sensor modelling was developed, mapping raw radar measurements to an occupancy grid-map. In contrast to classical approaches which typically rely on filtering power measurements based on their local context, a deep model was used to model this mapping in an end-to-end fashion accounting for a wider scene context. This allowed the proposed approach to significantly outperform classical methods in partitioning the world into occupied and free space. In addition, adopting a probabilistic approach allowed uncertainty in the occupancy state to

be successfully quantified, allowing regions of space occluded from the sensor view to be identified. The approach was trained without requiring manual labelling, using partial occupancy labels generated from lidar.

Simulation In chapter 4 a deep approach was developed which replicates radar's complex sensing process in simulation, without requiring expensive ray-tracing methods or specialist simulation environments. Using a pre-existing simulation world to generate world states in the form of elevation maps, a deep implicit model was used to capture the inherent uncertainty arising in the radar sensing process, allowing a stochastic simulation to be learnt from data. To train the model using *unaligned* datasets of simulated elevation maps and real-world radar measurements, the forward and backward processes were learnt side-by-side (such that the backward model could be used to infer the elevation state from real radar measurements). Both processes are trained in a joint optimisation using a combination of adversarial and cyclical consistency constraints, alongside an alignment loss with partial elevation labels generated in lidar. All training data can therefore be generated in simulation or automatically through data-collection in the real world. To test the realism of radar measurements generated in simulation, downstream models operating on radar data were trained using simulated radar measurements. When deployed in the real world we find that models trained in simulation are able to perform within 4 percentage points of a model trained purely in the real world. The backward model learnt as part of the same optimisation process can also be used to recast radar measurements back to a 2.5D representation of the world with reasonable accuracy.

Odometry Finally, in chapters 5 and 6 deep approaches were explored for the radar odometry task. In chapter 5 the robustness and interpretability of a correlative scan matching procedure was combined with a learnt radar feature representation, using a neural network to mask out distractor objects and noise artifacts. The entire approach remained differentiable, allowing the masking network to be explicitly optimised for the odometry task, supervising pose prediction directly. Through this approach the previous state-of-the-art was surpassed in accuracy by a significant

margin. In addition, through a probabilistic consideration, the uncertainty in the predicted pose was characterised and calibrated to real world errors (through a post training tuning step).

Building upon this approach, in chapter 6 properties of the Fourier Transform were used to decouple the search for translation and angle. Adopting a decoupled search vastly reduced the computational requirements of the scan matching procedure, and allowed real-time performance to be achieved at higher resolutions even when running on CPUs and limited embedded devices. This allowed the best real-time decoupled approach to outperform the best real-time approach proposed in chapter 5 in accuracy (on a CPU) whilst running significantly faster.

7.2 Future Research

The approaches developed in chapters 3 to 6 raise several interesting areas for future research.

Inverse Sensor Modelling In chapter 3 sparse training labels were generated automatically using lidar. Developing denser labelling strategies covering a greater range constitutes one potential avenue for future research. This could be achieved by integrating lidar sensor measurements over a longer time horizon into the labelling procedure, for example, whilst accounting for scene dynamics and changing occlusion.

Early experiments suggested some benefit to using Polar Transformer units to convert polar features in the encoder into Cartesian features in the decoder. Further, investigating network architectures specifically designed for the radar datatype, remains an interesting area for future research, so far under-explored in the literature.

Finally, the probabilistic inverse sensor modelling approach developed in chapter 3 could naturally be incorporated into a Bayesian filtering paradigm to allow sensor measurements to be integrated over time. Incorporating the learnt sensor model into a conventional mapping system for a larger scale mapping deployment is another interesting avenue to be explored. Much work has also been devoted to developing localisation and planning solutions based on occupancy grids which may also offer promising solutions when applied to radar.

Simulation Generalising the simulation approach developed in chapter 4 to facilitate the learning of down-stream models for more complex tasks remains an interesting area for future work and a prerequisite for fully capitalising on the full potential of TIS applied to radar. Whilst the simulator was able to train segmentation models – successfully partitioning the model into free, occupied and unknown space – early experiments on a limited number of hand-labelled examples, found that partitioning the world into finer-grained classes was significantly more challenging. Here, the development of larger scale radar datasets with a greater degree of labelling would be useful in providing more sophisticated testing scenarios.

Another limitation of the current approach is its generalisation to new sensor configurations – the radar's configuration is never explicitly passed to the model and is instead accounted for implicitly. In the worst case this requires a new simulator to be learnt for each configuration. Whilst new training data in this instance is easily generated, developing an approach in which the radar configuration may be used as a conditional input to the model may be a better solution.

Finally, the application of alternative GAN training objectives and network architectures may also bring additional benefits to this problem setting. Attention-based discrimination models may help with learning finer-grained details, for example.

Odometry The approach in chapter 5 appears to have a propensity to overfit to the geographical locations over which it was trained, as demonstrated by the drop in accuracy for the spatial cross-validation results presented in Table 2 in chapter 5. The release of larger scale radar odometry datasets with ground truth poses, offers a potential solution, allowing the feature space to be optimised over a wider array of environments and problem scenarios. Alternatively, developing specific schemes to counter overfitting in the low-data regime is also an interesting avenue of exploration.

At higher resolutions, the accuracy of the decoupled approach developed in chapter 6 appears to saturate. Whilst the approach remains competitive with other sparse odometry systems recently proposed in the literature, overcoming this limitation could result in additional gains in radar odometry accuracy, and constitutes an interesting area for future research.

7.3 Concluding Remarks

Moving beyond the classical rule-based approaches which have dominated to date, this thesis set out to explore the potential of deep and data-driven methods applied to radar across a range of tasks in robotics. As shown by the approaches developed for inverse sensor modelling, simulation and odometry, deep models offer significant advantages, allowing the performance of previous approaches to be surpassed, whilst running in real-time, and opening up new avenues for exploration. Meanwhile, through probabilistic approaches the natural uncertainty arising when working with radar sensor measurements may be successfully identified.

With a sustained interest from the research community, deep models and probabilistic approaches applied to radar may yet allow radar to reach its full potential in robotics as a first class member of the sensing suite.

8 Appendix

8.1 Probability Theory

8.1.1 Random Variables

Let $x \in \mathcal{X}$ denote a random variable belonging to the sample space $\mathcal{X}$ and distributed as $x \sim p(x)$ where $p(x)$ is the *marginal distribution*. To be a valid distribution we require that $p(x) \geq 0$ for every $x \in \mathcal{X}$ and that the sum of $p(x)$ across the entire sample space is equal to 1. Three cases are of particular interest:

- **Discrete Random Variables** Considering $x \in \{x_1 \dots x_K\}$ the *normalisation constraint* is written as $\sum_{x \in \mathcal{X}} p(x) = 1$. In this case combining the normalisation constraint with the *non-negativity constraint* $p(x) \geq 0$ implies that $p(x)$ must lie in the interval $[0, 1]$ for all $x \in \{x_1 \dots x_K\}$. The distribution $p(x)$ gives the probability of a particular outcome and is referred to as the *probability mass function* (p.m.f).

- **Continuous Random Variables** Considering $x \in \mathbb{R}$ the sum in the normalisation constraint becomes an integral $\int_{x \in \mathcal{X}} p(x)dx = 1$ and $p(x)$ is now a *probability density function* (p.d.f). The probability of x taking one outcome in $\mathbb{R}$ is now un-defined. Instead, the *cumulative density function* (c.d.f) $F(x) = \int_{\infty}^{x} p(x')dx'$ is used to calculate the probability that x lies in

the interval $[-\infty, a]$. Given that $p(x)$ satisfies the normalisation and non-negativity constraints it can be shown that $F(x)$ is a non-decreasing function in x and is bounded above and below such that $F(-\infty) \to 0$ and $F(+\infty) \to 1$.[1]

- **Multivariate Random Variables** In the case of a multi-variate random variable $\mathbf{x} \in \mathbb{R}^n$ then the normalisation constraint becomes $\int_{\mathbb{R}^n} p(\mathbf{x})d\mathbf{x} = 1$ and the non-negativity constraint is satisfied if $p(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. The distribution $p(\mathbf{x})$ is now the *joint probability density function* and captures the likelihood of each of the elements of $\mathbf{x} \triangleq (x_1, ..., x_n)$ occurring simultaneously.

8.1.2 The Rules of Probability

Considering two interacting random variables x and y, the *joint* $p(x, y)$ describes the outcome of two events x *and* y occurring simultaneously, whilst the *conditional* $p(x|y)$ describes the outcome of x given that y is already known[2]. Ensuring that the normalisation constraint and non-negativity constraint is satisfied for $p(x|y)$, $p(x, y)$, $p(x)$ and $p(y)$ gives rise to the *product* and *sum* rules

$$p(x, y) = p(x|y)p(y) \qquad \text{product rule} \tag{8.1}$$

$$p(y) = \int_{\mathcal{Y}} p(x, y)dx \qquad \text{sum rule} \tag{8.2}$$

which taken together form the foundational operations of probability. They provide a consistent framework for reasoning about random variables x and y and can be thought of as a natural extension of boolean logic extended to random events. Combining the product and sum rule in combination with the symmetry property of the joint $p(x, y) = p(y, x)$ gives rise to *Bayses* rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx} \tag{8.3}$$

(Note, equivalent results for the discrete case are retrieved replacing the integral in the sum rule with a summation.)

[1] Note, in some instances it is possible to define a c.d.f $F(x)$ satisfying these constraints which does not correspond to a valid $p(x)$. As a result in an axiomatic defintion of continuous random variables it is common to define the distribution of continous random variables using $F(x)$ instead. When $F(x)$ is also differentiable then $F(x)$ *will* correspond to a valid p.d.f $p(x)$.

[2] Whilst, the joint is a distribution in both x and y (satisfying $p(x, y) \geq 0$ for all x, y and $\int p(x, y)dxdy = 1$), the conditional $p(x|y)$ is only a distribution in x

Independent Events In the limit when knowledge of y provides no information about the event x then x and y are *independent*; $p(x|y) = p(x)$ collapses to the marginal for x and (from the product rule) the joint becomes $p(x, y) = p(x)p(y)$.

Multiple Interacting Random Variables Generalisations of the sum and product rule for the genral ND case are easily derived considering repeated applications of the sum and product rule for the 2D case. In this case, the sum, product and bayes rule all hold replacing x, y and dx with $\mathbf{x}$, $\mathbf{y}$ and $d\mathbf{x}$ in ... .

8.1.3 Expectations

Expectations of a function $f(\mathbf{x})$ with respect to some distribution $p(\mathbf{x})$ are defined as:

$$\mathbb{E}_{p(\mathbf{x})}\{f(\mathbf{x})\} \triangleq \int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \ . \tag{8.4}$$

8.1.3.1 The Law Of Large Numbers

Whilst, in some case it may be possible to determine a closed form solution to the expectation $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})]$ for general $f(\mathbf{x})$ and $p(\mathbf{x})$ this is not the case. The *law of large numbers* states

$$\mathbb{E}_{p(\mathbf{x})}\{f(\mathbf{x})\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n) \quad \text{with} \quad \mathbf{x}_n \sim p(\mathbf{x}) \tag{8.5}$$

and provides a powerful alternative allowing expectations to be estimated as the *empirical average* $\mathbb{E}_{p(\mathbf{x})}\{f(\mathbf{x})\} \approx \bar{f}$ where $\bar{f} \triangleq \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n)$ is referred to as the *Monte-Carlo estimate* of f.

8.1.3.2 The Reparameterisation Trick

In subsequent chapters the gradients $\nabla_\psi \mathbb{E}_{p_\psi(\mathbf{z})}\{f(\mathbf{z})\}$ will also be required where the distribution $p_\psi(\mathbf{z})$ depends on the parameters ψ. The *re-parmaterisation* trick offers a powerful solution in this case.

Provided it is possible to express the random variable $\mathbf{z} \sim p_\psi(\mathbf{z})$ as a deterministic variable $\mathbf{z} = g_\psi(\boldsymbol{\epsilon})$ induced from an auxillary variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$, the re-parameterisation trick allows expectations in the parameter $\mathbf{z}$ to be re-written

in the parameter $\boldsymbol{\epsilon}$

$$\mathbb{E}_{p_\psi(\mathbf{z})}\{f(\mathbf{z})\} = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left\{f\left(g_\psi\left(\boldsymbol{\epsilon}\right)\right)\right\} \tag{8.6}$$

where g_ψ is a function depending on the parameters ψ. Provided that g_ψ is also differentiable low-variance gradients $\nabla_\psi \mathbb{E}_{p_\psi(\mathbf{z})}\{f(\mathbf{z})\}$ can now be estimated as

$$\begin{aligned}
\nabla_\psi \mathbb{E}_{p_\psi(\mathbf{z})}\{f(\mathbf{z})\} &= \nabla_\psi \mathbb{E}_{p_\psi(\mathbf{z})}\{f(\mathbf{z})\} && (8.7)\\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left\{\nabla_\psi f\left(g_\psi(\boldsymbol{\epsilon})\right)\right\} && (8.8)\\
&\approx \tfrac{1}{K}\textstyle\sum_{i=1}^{K} \nabla_\psi f\left(g_\psi(\boldsymbol{\epsilon}_i)\right) && (8.9)
\end{aligned}$$

with $\boldsymbol{\epsilon}_i \sim p(\boldsymbol{\epsilon}_i)$

Many common distributions, can be re-parameterised in this way, including location scale distibutions (Eg. Gaussian and Laplace) and any distribution with a tractable inverse CDF (Eg. Cauchy, Logistic, Rayleigh, Gumbel, etc.) [29].

8.1.4 The Kullback Leibler Divergence

The Kulback Leibler (KL) divergence

$$d[p,q] \triangleq \mathbb{KL}[p||q] = \mathbb{E}_p\left\{\log p - \log q\right\} \tag{8.10}$$

satisfies

$$\begin{aligned}
&\mathbb{KL}[p||q] \geq 0 && \text{non-negativity} && (8.11)\\
&\mathbb{KL}[p||q] = 0 \Leftrightarrow p = q && \text{identity of indiscernibles} && (8.12)\\
&\mathbb{KL}[p||q] \neq \mathbb{KL}[q||p] && \text{non-symmetric} && (8.13)
\end{aligned}$$

and provides a natural measure of the similarity between two discrete, continuous or multivariate distributions p and q (using the respective definition of the expectation operator in each case).[3]

From an information perspective the KL divergence is the expected number of bits that need to be sent to a receiver in order to communicate the density p given

[3] As it is not symmetric $\mathbb{KL}[p||q] \neq \mathbb{KL}[q||p]$, it does not satisfy the triangle equality and so is not a metric.

that the receiver already knows the density q []. In general, however, the base of the log is arbitrary and when working with probability distributions – typically defined as functions involving the natural exponential – a base of e is often a more natural choice. In this case $\mathbb{KL}[p||q]$ is said to be measured in nats rather than bits.

The KL divergence has a key role to play in what is to follow; maximum likelihood – an approach adopted throughout machine learning – can be shown to be the same as minimising the KL divergence between the model and the true data generating process. This provides the guarantee that if the model has the capacity to represent the true process then in the limit of expectation it will. It also has a key role to play in variational Bayes for determining approximate but tractable posteriors in Bayesian inference problems as described in Sec. 8.2.2.1.

8.2 Inference

Up to this point, an axiomatic approach has been taken to how possibly multiple interacting real and/or discrete random variables may be defined, characterised and manipulated. Little attention has been given however to *what* probability represents which is an interesting and controversial question. How one answers this question has a profound impact on *inference* – the process of estimating an unknown θ given observations $\mathcal{D} \sim p_\star(\mathcal{D}|\theta)$ generated from the true process $p_\star$. In general opinion may be broadly divided down the lines of *frequentist* and *Bayesian* viewpoints.

8.2.1 Frequentist Inference

Frequentists view probability as representing the frequency with which an event occurs. This point of view is natural for repeatable events. For example in the case of a coin toss the fraction of times the coin comes up heads h in the limit of an infinite number of trials n is used to define the probability of a coin toss p such that $p \triangleq \lim_{n\to\infty} h/n$.

8.2.1.1 Maximum Likelihood

In a frequentist approach unknown θ is assumed fixed, and is inferred using the principle of *maximum likelihood estimation*

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathcal{D}|\theta) \tag{8.14}$$

choosing θ which maximises the likelihood of the *model* $p(\mathcal{D}|\theta)$ given observed data $\mathcal{D}$ from the *true* process $\mathcal{D} \sim p_\star(\mathcal{D}|\theta_\star)$.

Consistency of The Maximum Likelihood Estimator When the data $\mathcal{D} \triangleq \{\mathtt{D}_n\}_{n=1}^N$ corresponds to a series of independent and identically distributed

observations such that $p_\star(\mathcal{D}|\theta_\star) = \prod_{n=1}^{N} p_\star(\mathtt{D}_n|\theta_\star)$ we have

$$\tilde{\theta} = \arg\max_\theta \mathbb{E}_{p_\star(\mathtt{D}|\theta_\star)}\{\log p(\mathtt{D}_n|\theta)\} \tag{8.15}$$

$$\approx \arg\max_\theta \frac{1}{N}\textstyle\sum_n \log p(\mathtt{D}_n|\theta) \qquad \text{law of large numbers} \tag{8.16}$$

$$\approx \arg\max_\theta \log p(\mathcal{D}|\theta) \qquad \text{independence and log properties} \tag{8.17}$$

$$\approx \hat{\theta} \tag{8.18}$$

and so in the limit as $N \to \infty$ we have $\hat{\theta} \to \tilde{\theta}$. Provided that the model likelihood subsumes the true likelihood such that $p_\star(\mathtt{D}|\theta_\star) = p(\mathtt{D}|\theta_\star)$ this guarantees $\tilde{\theta} = \theta_\star$ and so in the limit $\hat{\theta} \to \theta_\star$.

In reality when $N < \infty$ sampling different datasets $\mathcal{D} \sim p_\star(\mathcal{D}|\theta)$ will lead to different estimates for $\hat{\theta}$. The uncertainty in $\hat{\theta}$ is quantified by using uncertainty in the data as a proxy (such as p-values).

Maximum Likelihood and KL Divergence An alternative derivation for maximum likelihood estimation can be derived by minimising the KL divergence between the true process $p_\star(\mathcal{D}|\theta_\star)$ and the model $p(\mathcal{D}|\theta)$

$$\hat{\theta} = \arg\min_\theta \mathbb{KL}[p_\star||p] \qquad \text{substitution} \tag{8.19}$$

$$= \arg\min_\theta \mathbb{E}_{p_\star(\mathcal{D}|\theta_\star)}\left\{\log p_\star(\mathcal{D}|\theta_\star) - \log p(\mathcal{D}|\theta)\right\} \qquad \text{definition} \tag{8.20}$$

$$= \arg\max_\theta \mathbb{E}_{p_\star(\mathcal{D}|\theta_\star)}\left\{\log p(\mathcal{D}|\theta)\right\} \qquad \text{keep terms in } p \tag{8.21}$$

$$\approx \arg\max_\theta \frac{1}{N}\textstyle\sum_n \log p(\mathtt{D}_n|\theta) \qquad \text{law of large numbers} \tag{8.22}$$

$$\approx \arg\max_\theta \log p(\mathcal{D}|\theta) \qquad \text{i.i.d} \tag{8.23}$$

in which maximum likelihood estimation naturally emerges. Once again in the limit $N \to \infty$ this approximation becomes exact and the positive defiteness of the KL divergence guarantees $p(\mathcal{D}|\theta) \to p_\star(\mathcal{D}|\theta_\star)$.

8.2.1.2 Empirical Risk

In the case of IID samples the maximum likelihood estimate can be re-written as

$$\hat{\theta} = \arg\min_\theta \frac{1}{N}\sum_n \ell(\theta; \mathtt{D}_n) \approx \arg\min_\theta \mathbb{E}_{p_\star(\mathtt{D}|\theta_\star)}\{\ell(\theta; \mathtt{D})\} \tag{8.24}$$

where $\ell(\theta; \mathtt{D}) \triangleq -\log(p(\mathtt{D}|\theta)$. Whilst the *loss function* $\ell(\theta; \mathtt{D}_n)$ was derived by considering the log-likelihood $\log p(\mathtt{D}_n|\theta)$ other loss functions $\ell : \Theta \to \mathbb{R}$ may be defined. Considering Eq. (8.24) with other loss functions $\ell(\theta; \mathtt{D})$ gives a general approach to parameter estimation – of which maximum likelihood is a special case – referred to as *empirical risk minimisation*. A loss function is *consistent* if in the limit as $N \to \infty$ solving Eq. (8.24) gives $\theta \to \theta_\star$.

Improper Likelihoods Note that as $\ell : \Theta \to \mathbb{R}$, defining $\tilde{p}(\mathcal{D}|\theta) \triangleq \exp -\ell(\theta)$ we have $\tilde{p}(\mathcal{D}|\theta) \geq 0$ for every θ. Whilst, the distribution $\tilde{p}(\mathcal{D}|\theta) \geq 0$ satisfies the non-negativity constraint, it is not necessarily normalised (doesn't integrate to 1) and is referred to as an *improper* likelihood. Reversing this line of reasoning *any* loss $\ell : \Theta \to \mathbb{R}$ can be thought of as deriving from an *improper* likelihood $\ell(\theta; \mathcal{D}) \triangleq -\log \tilde{p}(\mathcal{D}|\theta)$.

8.2.2 Bayesian Inference

In the *Bayesian* view probability quantifies *belief* and does not necessarily correspond to the actual frequency with which an event occurs. When it comes to inference the unknown θ is assumed *random* $\theta \sim p(\theta)$ with *prior* $p(\theta)$. In light of observations $\mathcal{D} \sim p(\mathcal{D}|\theta)$ the belief in θ is updated using Bayes rule

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta} = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \ . \tag{8.25}$$

to form the *posterior* $\theta \sim p(\theta|\mathcal{D})$. Uncertainty in θ is naturally captured by $\theta \sim p(\theta|\mathcal{D})$.

To estimate the posterior $p(\theta|\mathcal{D})$ the marginal evidence $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$ must be determined. For suitable choices of likelihood $p(\mathcal{D}|\theta)$ and prior $p(\theta)$ the marginal evidence can be calculated in closed form; this is the case when $p(\mathcal{D}|\theta)$ and $p(\theta)$ are conjugate to one another (as is the case in Bayesian Linear Regression) where the posterior distribution will have the same distributional form as the prior. For other choices of prior – where inference is no longer tractable – approximate Bayesian methods must be employed such as *Variational Bayes* or *Sampling Approaches*.

8.2.2.1 Variational Bayes

Adopting a *variational approach*, an approximate posterior distribution $q(\theta)$ is introduced and fitted to the true posterior $p(\theta|\mathcal{D})$. This is achieved by minimising the KL divergence

$$q_\theta = \arg\min_{q_\theta} \mathbb{KL}[q_\theta \| p_{\theta|\mathcal{D}}] \ . \tag{8.26}$$

Re-writing the KL divergence gives

$$\begin{aligned}
\mathbb{KL}\left[q_\theta \| p_{\theta|\mathcal{D}}\right] &= \mathbb{E}_{q_\theta}\left\{\log q_\theta - \log p_{\theta|\mathcal{D}}\right\} && \text{definition} && (8.27)\\
&= \mathbb{E}_{q_\theta}\left\{\log q_\theta - \log p_{\theta,\mathcal{D}} + \log p_{\mathcal{D}}\right\} && \text{Bayses rule} && (8.28)\\
&= \mathbb{E}_{q_\theta}\left\{\log q_\theta - p_{\theta,\mathcal{D}}\right\} + \log p_{\mathcal{D}} && \text{normalisation} && (8.29)\\
&\triangleq -\mathbb{ELBO}[\log q_\theta \| p_{\theta,\mathcal{D}}] + \log p_{\mathcal{D}} && \text{definition} && (8.30)
\end{aligned}$$

and so the identity

$$\mathbb{ELBO}[q_\theta \| p_{\theta,\mathcal{D}}] = \log p_{\mathcal{D}} - \mathbb{KL}\left[q_\theta \| p_{\theta|\mathcal{D}}\right] \tag{8.31}$$

where $\mathbb{ELBO}[\log q_\theta, p_{\theta,\mathcal{D}}]$ is the *evidence lower bound*.

Variational Bayes The otpimisation problem in Eq. (8.26) can therefore be equivalently re-written as

$$q_\theta = \arg\max_{q_\theta} \mathbb{ELBO}[q_\theta \| p_{\theta,\mathcal{D}}] \tag{8.32}$$

noting that the model evidence $\log p_{\mathcal{D}}$ does not depend on θ. Crucially, Eq. (8.32) in contrast to Eq. (8.26) only requires evaluation of the joint distribution $p_{\theta,\mathcal{D}}$.

Approximating the Model Evidence In addition to allowing an approximation $q(\theta)$ to the true posterior $p(\theta|\mathcal{D})$ to be established, $\mathbb{ELBO}[q_\theta \| p_{\theta,\mathcal{D}}]$ also provides an estimate for the model evidence

$$\mathbb{ELBO}[q_\theta \| p_{\theta,\mathcal{D}}] \approx -\log p_{\mathcal{D}} \tag{8.33}$$

considering Eq. (8.30) and letting $q(\theta) \to p(\theta|\mathcal{D}) \implies \mathbb{KL}\left[q_\theta \| p_{\theta|\mathcal{D}}\right] \to 0$.

Re-interpreting the ELBO Further insight can also be gained into this approach by re-writing the definition of $\mathbb{ELBO}[q_\theta \| p_{\theta,\mathcal{D}}]$

$$\begin{aligned}\mathbb{ELBO}[\log q_\theta \| p_{\theta,\mathcal{D}}] &= \mathbb{E}_{q_\theta}\left\{\log q_\theta - \log p_{\theta,\mathcal{D}}\right\} && (8.34)\\ &= \mathbb{E}_{q_\theta}\left\{\log \frac{q_\theta}{p_{\theta|\mathcal{D}}} - \log p_{\theta|\mathcal{D}}\right\} && (8.35)\\ &= \mathbb{KL}\left[q_\theta \| p_\theta\right] - \mathbb{E}_{q_\theta}\left\{\log p_{\theta|\mathcal{D}}\right\} && (8.36)\end{aligned}$$

and so minimising $\mathbb{ELBO}[\log q_\theta \| p_{\theta,\mathcal{D}}]$ is also seen to be equivalent to minimising the KL divergence between q_θ and the prior p_θ whilst maximising the *approximate model evidence* $\mathbb{E}_{q_\theta}\left\{p_{\theta|\mathcal{D}}\right\}$.

8.2.2.2 Sampling Approaches

Another approach is to approximate the posterior distribution as an empirical distribution $p(\theta|\mathcal{D}) \approx \sum_{s=1} \omega_s \delta_{\theta_s}(\theta)$ such that

$$\mathbb{E}_{p(\theta|\mathcal{D})}\{f(\theta)\} \approx \int \textstyle\sum_s \omega_s \delta_{\theta_s}(\theta) f(\theta) d\theta = \sum_s \omega_s f(\theta_s) \tag{8.37}$$

where $\Theta = \{\theta_s\}_{s=1}^S$ is a set of particles and $\{\omega_s\}_{s=1}^S$ are particle weights such that $\sum_s \omega_s = 1$. When sampling from the posterior $\theta_s \sim p(\theta|\mathcal{D})$ directly then $\omega_s = 1/S$.

Importance Sampling However, sampling from other distributions $\theta_s \sim q(\theta)$ is also possible. Noting that Eq. (8.37) can be re-written

$$\begin{aligned}\mathbb{E}_{p(\theta|\mathcal{D})}\{f(\theta)\} &= \int p(\theta|\mathcal{D}) f(\theta) d\theta && (8.38)\\ &= \int q(\theta) \frac{p(\theta|\mathcal{D})}{q(\theta)} f(\theta) d\theta && (8.39)\\ &\approx \frac{1}{S}\sum_s \frac{p(\theta_s|\mathcal{D})}{q(\theta_s)} f(\theta_s) \qquad \theta_s \sim q(\theta) && (8.40)\\ &\approx \frac{1}{L}\sum_s \frac{\tilde{p}(\theta_s)}{\tilde{q}(\theta_s)} f(\theta_s) \qquad \tilde{p}(\theta) \propto p(\theta|\mathcal{D}),\ \tilde{q}(\theta) \propto q(\theta) && (8.41)\\ &\approx \frac{1}{K}\sum_s \tilde{\omega}_s f(\theta_s) \qquad \tilde{\omega}_s \propto \frac{\tilde{p}(\theta_s)}{\tilde{q}(\theta_s)} && (8.42)\end{aligned}$$

and that $\mathbb{E}_{p(\theta|\mathcal{D})}\{1\} = 1 \implies K = \sum_s \tilde{\omega}_s$ this gives

$$\mathbb{E}_{p(\theta|\mathcal{D})}\{f(\theta)\} \approx \sum_s \omega_s f(\theta_s) \qquad \tilde{\omega}_s = \frac{\tilde{p}(\theta_s)}{\tilde{q}(\theta_s)} \qquad \omega_s = \frac{\tilde{\omega}_s}{\sum_s \tilde{\omega}_s} \tag{8.43}$$

where $\theta_s \sim q(\theta)$ are sampled from a *proposal distribution* $q(\theta)$. Note, that the weights are calculated without evaluating $p(\theta|\mathcal{D})$, only requiring that it is possible to evaluate an *unormalised* density $\tilde{p}(\theta) \propto p(\theta|\mathcal{D})$ which is readily provided by the joint distribution $\tilde{p}(\theta) \triangleq p(\theta, \mathcal{D}) = p(\mathcal{D}|\theta)p(\theta)$. Methods exploiting Eq. (8.43) are referred to as *importance sampling*.

Grid Sampling Setting the proposal distribution to a uniform density and considering sampling θ_s over an evenly space grid

$$\theta_s \sim \texttt{Uni}(\|\theta\|_2 < d) \qquad\qquad \omega_s = p(\mathcal{D}|\theta)p(\theta) \tag{8.44}$$

Sampling From The Posterior Distribution Directly If samples are required *directly* from the posterior distribution $\theta_s \sim p(\theta|\mathcal{D})$ then a *resampling* step can be performed in which new particles are sampled from the empirical distribution $p(\theta|\mathcal{D}) \approx \sum_{s=1} \omega_s \delta_{\theta_s}(\theta)$ using weights ω_s as discrete probabilities.[4]

Sampling will be inefficient if the proposal distribution $q(\theta)$ is poorly aligned to the posterior $p(\theta|\mathcal{D})$ with $\omega_s \approx 0$. This problem is exacerbated as the dimensionality of θ increases. Markov Chain Monte Carlo (MCMC) techniques such as Metropolis Hastings approaches – of which Gibbs sampling is a famous example – have been designed in response to this problem. Whilst, an interesting and powerful approach to Bayesian inference MCMC approaches come with a significant increase in complexity and remain outside the scope of this thesis.

[4] Another simple alternative to sampling particles directly from the posterior is *rejection sampling*. In the case where $\theta \in \mathbb{R}$ the un-normalised posterior is *bounded* by a proposal distribution such that $\tilde{q}(\theta) = kq(\theta)$ for some constant k such that $\tilde{q}(\theta) \geq \tilde{p}(\theta)$ for all possible θ. A particle is then proposed $\theta_s \sim q(\theta)$ and rejected if $s > \tilde{p}(\theta_s)$ where $s \sim \texttt{Uni}\left(|s| < kq(\theta_s)\right)$. All particles which are not rejected can be shown to be distributed as $\theta_s \sim p(\theta|\mathcal{D})$ and are added to the set Θ. This approach is easily generalised to $\theta \in \mathbb{R}^d$ by sampling each element in $\theta = [\theta_1, ..., \theta_d]^\top$ independently. One significant challenge with rejection sampling however is determining the constant k such that $\tilde{q}(\theta) \geq \tilde{p}(\theta)$. In contrast, k is implicitly estimated using importance sampling by ensuring that all the weights sum to 1.

8.3 Optimisation

When considering deep models, the parameter space is of a high dimensionality and in general – as they are non-linear in the parameters ϕ – the training objective $\mathcal{L}(\phi)$ is typically non-convex. Second-order non-linear optimisation methods that rely on calculating the Hessian of the loss function with respect to the network parameters are infeasible due to the high dimensionality of $\boldsymbol{\phi}$. Instead, first-order methods are used where the estimate of the parameters ϕ is iteratively refined using a linear update rule

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} - \delta\boldsymbol{\phi}_k \quad \text{for} \quad k = 0...K \tag{8.45}$$

exploiting local *gradient* information about the current estimate $\boldsymbol{\phi}_k$. Note, in this case all the parameters from the network $\phi = \{\phi_1, \ldots, \phi_L\}$ are flattened and concatenated to give $\boldsymbol{\phi} \in \mathbb{R}^d$.

Gradient Descent For a smooth loss $\mathcal{L}(\boldsymbol{\phi})$ the method of *gradient descent* [53, 54] defines an update rule

$$\delta\boldsymbol{\phi}_k \triangleq \epsilon\mathbf{g}_{k-1} \quad \text{with} \quad \mathbf{g}_{k-1} \triangleq \nabla_\phi\mathcal{L}(\boldsymbol{\phi}_{k-1}) \tag{8.46}$$

where $\epsilon \in [0, \infty)$ is referred to as the *step size* or *learning rate*.

Considering the first-order Taylor expansion for $\mathcal{L}(\boldsymbol{\phi})$ about the point $\boldsymbol{\phi}_{k-1}$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\phi}_k) &\approx \mathcal{L}(\boldsymbol{\phi}_{k-1}) + (\boldsymbol{\phi}_k - \boldsymbol{\phi}_{k-1})^\top\mathbf{g}_{k-1} && \text{first-order taylor series} && (8.47)\\
&= \mathcal{L}(\boldsymbol{\phi}_{k-1}) - \epsilon\mathbf{g}_{k-1}^\top\mathbf{g}_{k-1} && \text{Eq. (8.45) \& Eq. (8.46)} && (8.48)\\
&= \mathcal{L}(\boldsymbol{\phi}_{k-1}) - \epsilon\|\mathbf{g}_{k-1}\|_2^2 && \text{definition of } \|\cdot\|_2 && (8.49)
\end{aligned}$$

it becomes clear that provided that ϵ is small enough so that the approximation in Eq. (8.47) is valid, the loss will continue to decrease until the first-order condition for optimality $\lim_{k\to\infty}\mathbf{g}_k = \mathbf{0}$ is met giving $\hat{\boldsymbol{\phi}} = \lim_{k\to\infty}\boldsymbol{\phi}_k$ [27, 54].

When ϵ is small, an impractically large number of steps is needed for convergence. On the other hand, when ϵ is large then the approximation in Eq. (8.47) is no longer

valid – there is no guarantee that $\boldsymbol{\phi}$ will even converge. In practice the step size ϵ is typically found by performing a hyper-parameter sweep choosing the best ϵ from amongst a set of possible candidates $\{\epsilon_1, ..., \epsilon_k\} \subset [a, b]$.

Stochastic Gradient Descent For large datasets $\mathcal{D}$ calculating the loss $\mathcal{L}(\boldsymbol{\phi}_k; \mathcal{D})$ and its gradient $\mathbf{g}_{k-1}$ may require an intractable amount of time and/or memory. Instead, a batch of the full dataset $\mathcal{B} \subset \mathcal{D}$ is used to estimate $\mathcal{L}(\boldsymbol{\phi}_k; \mathcal{D}) \approx \mathcal{L}(\boldsymbol{\phi}_k; \mathcal{B})$ at each iteration. Using this approximation in conjunction with the iterative update scheme defined through Eq. (8.46) and Eq. (8.45) is referred to as the method of *stochastic gradient descent* (SGD) [55, 56].

This comes at the expense of a higher variance gradient and as a result a loss which converges more slowly and may even fail to converge. On the other hand, when the number of parameters d is relatively large compared to the number of training examples N, the added stochasticity of SGD with $B < N$ has been posited to have a regularising effect [57, 58], leading to better generalisation.

Momentum As an attempt to reduce the variance in the gradient estimate SGD with *momentum* [59] uses an update rule

$$\boldsymbol{\mu}_k = \gamma_1 \boldsymbol{\mu}_{k-1} + (1 - \gamma_2)\mathbf{g}_{k-1} \tag{8.50}$$

$$\delta\boldsymbol{\phi}_k = \epsilon \boldsymbol{\mu}_k \tag{8.51}$$

where $\boldsymbol{\mu}_k$ provides a running average of the gradient. The exponetial decay factors $\gamma_1, \gamma_2 \in [0, 1]$ control how quickly $\boldsymbol{\mu}_{k-1}$ decays whilst ϵ is the gradient step size or learning rate (as before). By setting $\gamma_1 = 1$ and $\gamma_2 = 0$ the SGD update rule is recovered.

Adaptive Learning Rates One downside to stochastic gradient descent is that a single learning rate is used to update all the parameters. Modern optimisers such as RMSProp [60], AdaGrad [61] and Adam [51] consider adding in *adaptive* learning

rates for each parameter. For RMSProp the update rule is given as

$$\boldsymbol{\nu}_k = \beta\boldsymbol{\nu}_{k-1} + (1-\beta)\mathbf{g}_{k-1}^2 \tag{8.52}$$

$$\delta\boldsymbol{\phi}_k = \epsilon\frac{1}{\sqrt{\boldsymbol{\nu}_k} + 10^{-8}}\mathbf{g}_{k-1} \tag{8.53}$$

where $\boldsymbol{\nu}_k$ gives a moving average of the second moment of the gradient of each parameter. The Adam update rule, on the other hand, combines the update rules for RMSProp and Momentum to give

$$\boldsymbol{\mu}_k = \gamma\boldsymbol{\mu}_{k-1} + (1-\gamma)\mathbf{g}_{k-1} \tag{8.54}$$

$$\boldsymbol{\nu}_k = \beta\boldsymbol{\nu}_{k-1} + (1-\beta)\mathbf{g}_{k-1}^2 \tag{8.55}$$

$$\delta\boldsymbol{\phi}_k = \epsilon\frac{c_k}{\sqrt{\boldsymbol{\nu}_k} + 10^{-8}}\boldsymbol{\mu}_k \tag{8.56}$$

where $c_k = (1-\gamma^k)^{-1}(1-\beta^k)^{-1/2}$ is a bias correction term (which will tend to 1 as $k \to \infty$).

In comparison to other optimisers Adam has been posited to have faster convergence rates and is the default optimisation method used in this thesis.

Backpropagation

Central to the first order methods described so far is an assumption that it is possible to calculate the gradient $\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi})$. Whilst, layers $\mathbf{y}_i = y_{\boldsymbol{\phi}_i}(\mathbf{y}_{i-1})$ are designed to be differentiable in their parameters $\boldsymbol{\phi}_i$, care needs to be taken when calculating gradients $\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\boldsymbol{\phi})$. Central to the first order methods described so far is an assumption that it is possible to calculate the gradient $\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi})$. Whilst, layers $\mathbf{y}_i = y_{\boldsymbol{\phi}_i}(\mathbf{y}_{i-1})$ are designed to be differentiable in their parameters $\boldsymbol{\phi}_i$, care needs to be taken when calculating gradients $\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\boldsymbol{\phi})$. Naïve application of the chain rule

$$\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\boldsymbol{\phi}) \triangleq \frac{\partial\mathcal{L}}{\partial\boldsymbol{\phi}_i} = \underbrace{\frac{\partial\mathcal{L}}{\partial\mathbf{y}_L}}_{1\times n_L}\underbrace{\frac{\partial\mathbf{y}_L}{\partial\mathbf{y}_{L-1}}}_{n_L\times n_{L-1}}\cdots\underbrace{\frac{\partial\mathbf{y}_{i+1}}{\partial\mathbf{y}_i}}_{n_{i+1}\times n_i}\underbrace{\frac{\partial\mathbf{y}_i}{\partial\boldsymbol{\phi}_i}}_{n_i\times d_i} \tag{8.57}$$

involves the calculation and multiplication of matrix jacobians which for modern neural networks soon requires an intractable amount of time and memory (particularly when considering networks typically composed of 100s of layers and

mapping high-dimensional inputs – such as images – to high-dimensional outputs) [62]. Instead, writing

$$\mathbf{g}_i \triangleq \frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_i} = \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i}}_{1\times n_i}\underbrace{\frac{\partial \mathbf{y}_i}{\partial \boldsymbol{\phi}_i}}_{n_i \times d_i} = \frac{\partial}{\partial \boldsymbol{\phi}_i}\Big(\underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i}}_{1\times n_i}\underbrace{\mathbf{y}_i}_{n_i}\Big) \triangleq \nabla_{\boldsymbol{\phi}_i}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle \tag{8.58}$$

where $\boldsymbol{\delta}_i \triangleq \nabla_{\mathbf{y}_i}\mathcal{L}(\phi)$ allows the gradient $\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\boldsymbol{\phi})$ to be calculated as the derivative of the inner product $\mathbf{g}_i = \nabla_{\boldsymbol{\phi}_i}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle$ side-stepping the calculation of large matrix Jacobians entirely [62, 63].[5]

In a similar way

$$\boldsymbol{\delta}_{i-1} \triangleq \frac{\partial \mathcal{L}}{\partial \mathbf{y}_{i-1}} = \underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i}}_{1\times n_i}\underbrace{\frac{\partial \mathbf{y}_i}{\partial \mathbf{y}_{i-1}}}_{n_i \times n_{i-1}} = \frac{\partial}{\partial \mathbf{y}_{i-1}}\Big(\underbrace{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i}}_{1\times n_i}\underbrace{\mathbf{y}_i}_{n_i}\Big) \triangleq \nabla_{\mathbf{y}_{i-1}}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle \tag{8.59}$$

which allows the gradient $\boldsymbol{\delta}_i \triangleq \nabla_{\mathbf{y}_i}\mathcal{L}(\phi)$ at each layer to be calculated using the iterative scheme

$$\boldsymbol{\delta}_{i-1} = \nabla_{\mathbf{y}_{i-1}}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle \quad \text{for} \quad i = L...1 \tag{8.60}$$

where in this case the gradients are gradually update from layer $i = L$ back to the input $i = 1$.

Calculating gradients $\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\phi)$ and $\boldsymbol{\delta}_{i-1} = \nabla_{\mathbf{y}_{i-1}}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle$ using Eq. (8.58) and Eq. (8.60) is referred to as the *backpropagation* algorithm [63]. In reality this procedure is easily automated by ensuring that each layer implements both a forward mode $\mathbf{y}_i = y_{\boldsymbol{\phi}_i}(\mathbf{y}_{i-1})$ and a backward mode $\mathbf{g}_i, \boldsymbol{\delta}_{i-1} \triangleq y'_{\boldsymbol{\phi}_i}(\boldsymbol{\delta}_i, \mathbf{y}_i)$, calculating the gradient of the loss with respect to both the layer parameters $\mathbf{g}_i \triangleq \nabla_{\boldsymbol{\phi}_i}\mathcal{L}(\phi) = \nabla_{\boldsymbol{\phi}_i}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle$ and the input to the layer $\boldsymbol{\delta}_{i-1} \triangleq \nabla_{\mathbf{y}_{i-1}}\mathcal{L}(\phi) = \nabla_{\mathbf{y}_{i-1}}\langle \boldsymbol{\delta}_i, \mathbf{y}_i\rangle$.

Modern deep learning libraries [64, 65] provide both forward and backward implementations for a wide range of functions and layers allowing for the automatic differentiation of a plethora of loss functions and models, in many cases without further thought. Alongside developments in parallel computing hardware and technical breakthroughs, the availability of such libraries – readily implemented

[5] The final part of equation Eq. (8.58) follows from the product rule, noting that layers $\mathbf{y}_k$ for $L \leq k < i$ do not depend on $\boldsymbol{\phi}_i$ either implicitly or explicitly such that $\nabla_{\boldsymbol{\phi}_i}\{\nabla_{\mathbf{y}_i}\mathcal{L}(\phi)\} = \mathbf{0}$.

as a result of the layer abstraction brought by the back-propagation algorithm – can be at least partially accredited with the rise and success of deep learning approaches in recent years.

8.4 Derivations

8.4.1 Loss Functions For Assumed Density Models

As described in Sec. 2.2.1 many common loss functions used to train deep neural networks can be derived as maximum likelihood objectives $\ell(\phi) = -\log p_\phi(\mathbf{y}|\mathbf{x})$ corresponding to different choices of density $p_\phi(\mathbf{y}|\mathbf{x})$. Proofs for the results stated in Sec. 2.2.1 can be found below.

Gaussian Model For $p_\phi(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Nor}\left(\mathbf{y}|y_\theta(\mathbf{x}), \sigma^2\mathbf{I}\right)$ with $y_\theta : \mathcal{X} \to \mathbb{R}^{n_{out}}$:

$$\hat{\phi} = \arg\min_\theta -\sum_n \log p_\theta(\mathbf{y}_n|\mathbf{x}_n) \tag{8.61}$$

$$= \arg\min_\theta -\sum_n \log\left\{|2\pi\sigma^2\mathbf{I}|^{-\frac{1}{2}} \exp -\frac{1}{2\sigma^2}\|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_2^2\right\} \tag{8.62}$$

$$= \arg\min_\theta \frac{1}{2}\sum_n \log|2\pi\sigma^2\mathbf{I}| + \frac{1}{\sigma^2}\|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_2^2 \tag{8.63}$$

$$= \arg\min_\theta \frac{1}{N}\sum_n \|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_2^2 \tag{8.64}$$

$$= \frac{1}{N}\arg\min_\phi \sum_n \ell_2(\theta; \mathbf{x}_n, \mathbf{y}_n) \tag{8.65}$$

the Mean Squared Error (MSE) training objective is recovered similar.

Laplace Model When $\mathbf{y} \in \mathbb{R}^{n_{out}}$ and $p_\theta(\mathbf{y}|\mathbf{x}) \triangleq \texttt{Lap}\left(\mathbf{y}|y_\theta(\mathbf{x}), \sigma^2\right)$, with $y_\theta : \mathcal{X} \to \mathbb{R}^{n_{out}}$:

$$\hat{\phi} = \arg\min_\theta -\sum_n \log p_\theta(\mathbf{y}_n|\mathbf{x}_n) \tag{8.66}$$

$$= \arg\min_\theta -\sum_n \log\left\{|2\sigma\mathbf{I}|^{-1} \exp\frac{1}{\sigma}\|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_1\right\} \tag{8.67}$$

$$= \arg\min_\theta \sum_n \log|2\sigma\mathbf{I}| + \frac{1}{\sigma}\|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_1 \tag{8.68}$$

$$= \arg\min_\theta \frac{1}{N}\sum_n \|\mathbf{y}_n - y_\theta(\mathbf{x}_n)\|_1 \tag{8.69}$$

$$= \arg\min_\phi \frac{1}{N}\sum_n \ell_1(\theta; \mathbf{x}_n, \mathbf{y}_n) \tag{8.70}$$

gives Mean Absolute Error (MSE) training objective.

Bernoulli Model $p(y|\mathbf{x}) \triangleq \texttt{Ber}(y|\pi_\phi(\mathbf{x}))$ with $\pi_\phi : \mathcal{X} \to [0,1]$:

$$\hat{\phi} = \arg\min_\phi \sum_n -\log p(y_n|\mathbf{x}_n) \tag{8.71}$$

$$= \arg\min_\phi \sum_n -\log \left(\texttt{Ber}(y_n|\pi_\phi(\mathbf{x}))\right) \tag{8.72}$$

$$= \arg\min_\phi \sum_n -\log \left(\pi_\phi(\mathbf{x}_n)^{y_n}(1-\pi_\phi(\mathbf{x}_n))^{1-y_n}\right) \tag{8.73}$$

$$= \frac{1}{N} \arg\min_\phi \sum_n -y_n \log\left(\pi_\phi(\mathbf{x}_n)\right) - (1-y_n)\log\left(1-\pi_\phi(\mathbf{x}_n)\right) \tag{8.74}$$

$$= \arg\min_\phi \frac{1}{N} \sum_n \ell_{bce}(\theta; \mathbf{x}_n, \mathbf{y}_n) \tag{8.75}$$

and the *Binary Cross Entropy* $\ell_{bce}(\theta; \mathbf{x}_n, \mathbf{y}_n)$ naturally emerges.

References

[1] Alexey Dosovitskiy et al. “Flownet: Learning optical flow with convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766.

[2] Charles R Qi et al. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[4] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[6] Pei Sun et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[7] R. Kesten et al. *Level 5 Perception Dataset 2020*. https://level-5.global/level5/data/. 2019.

[8] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] 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)*. IEEE. 2018, pp. 6045–6052.

[10] Daniel Adolfsson et al. “CFEAR Radarodometry-Conservative Filtering for Efficient and Accurate Radar Odometry”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 5462–5469.

[11] Martin Adams, Martin David Adams, and Ebi Jose. *Robotic navigation and mapping with radar*. Artech House, 2012.

[12] Martin Holder et al. “Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 2616–2622.

[13] Dipak L Sengupta and Tapan K Sarkar. “Maxwell, Hertz, the Maxwellians, and the early history of electromagnetic waves”. In: *IEEE Antennas and Propagation Magazine* 45.2 (2003), pp. 13–19.

[14] Christian Hüelsmeyer. *Telemobiloscope, Patent Number FR343846A (expired.* 15th October 1904.

[15] Merrill Ivan Skolnik. “Radar handbook”. In: (1970).

[16] Mirosław Adamski et al. “Effects of transmitter phase noise on signal processing in FMCW radar”. In: *Proc. 2000 International Conf. Signals & Electronic Systems,(Ustronie, Poland)*. 2000, pp. 51–56.

[17] Ebi Jose et al. “Predicting millimeter wave radar spectra for autonomous navigation”. In: *IEEE Sensors Journal* 10.5 (2010), pp. 960–971.

[18] Fabrizio Argenti et al. “A tutorial on speckle reduction in synthetic aperture radar images”. In: *IEEE Geoscience and remote sensing magazine* 1.3 (2013), pp. 6–35.

[19] Patrick DL Beasley. “The influence of transmitter phase noise on FMCW radar performance”. In: *2006 European Microwave Conference*. IEEE. 2006, pp. 1810–1813.

[20] HM Finn. “Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates”. In: *Rca Rev.* 29 (1968), pp. 414–465.

[21] GB Goldstein. “False-alarm regulation in log-normal and Weibull clutter”. In: *IEEE Transactions on Aerospace and Electronic Systems* 1 (1973), pp. 84–92.

[22] N Levanon and M Shor. “Order statistcs CFAR for Weibull background”. In: *IEE Proceedings F (Radar and Signal Processing)*. Vol. 137. 3. IET. 1990, pp. 157–162.

[23] Peter Weber and Simon Haykin. “Ordered statistic CFAR processing for two-parameter distributions with variable skewness”. In: *IEEE Transactions on Aerospace and Electronic Systems* 6 (1985), pp. 819–821.

[24] Hermann Rohling. “Radar CFAR thresholding in clutter and multiple target situations”. In: *IEEE transactions on aerospace and electronic systems* 4 (1983), pp. 608–621.

[25] V Gregers Hansen and James H Sawyers. “Detectability loss due to" greatest of" selection in a cell-averaging CFAR”. In: *IEEE Transactions on Aerospace and Electronic Systems* 1 (1980), pp. 115–118.

[26] Rob Weston, Oiwi Parker Jones, and Ingmar Posner. “There and Back Again: Learning to Simulate Radar Data for Real-World Applications”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021), pp. 12809–12816.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[28] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems* 30 (2017).

[29] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[30] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[31] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[32] Ian Goodfellow. “Nips 2016 tutorial: Generative adversarial networks”. In: *arXiv preprint arXiv:1701.00160* (2016).

[33] Xudong Mao et al. “Least squares generative adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2794–2802.

[34] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. “f-gan: Training generative neural samplers using variational divergence minimization”. In: *Advances in neural information processing systems* 29 (2016).

[35] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[36] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[37] George Papamakarios et al. “Normalizing flows for probabilistic modeling and inference”. In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64.

[38] Christina Winkler et al. “Learning likelihoods with conditional normalizing flows”. In: *arXiv preprint arXiv:1912.00042* (2019).

[39] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

[40] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[41] Takeshi Fukushima and Jon C Nixon. “Analysis of reduced forms of biopterin in biological tissues and fluids”. In: *Analytical biochemistry* 102.1 (1980), pp. 176–188.

[42] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Icml*. 2010.

[43] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[44] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3.

[45] Kouichi Yamaguchi et al. “A neural network for speaker-independent isolated word recognition.” In: *ICSLP*. 1990.

[46] Maximilian Riesenhuber and Tomaso Poggio. “Hierarchical models of object recognition in cortex”. In: *Nature neuroscience* 2.11 (1999), pp. 1019–1025.

[47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[49] Michal Drozdzal et al. "The importance of skip connections in biomedical image segmentation". In: *Deep learning and data labeling for medical applications*. Springer, 2016, pp. 179–187.

[50] Carlos Esteves et al. "Polar transformer networks". In: *arXiv preprint arXiv:1709.01889* (2017).

[51] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[52] Tamir Hazan, Subhransu Maji, and Tommi Jaakkola. "On sampling from the gibbs distribution with random maximum a-posteriori perturbations". In: *Advances in Neural Information Processing Systems* 26 (2013).

[53] Augustin Cauchy et al. "Méthode générale pour la résolution des systemes d'équations simultanées". In: *Comp. Rend. Sci. Paris* 25.1847 (1847), pp. 536–538.

[54] Haskell B Curry. "The method of steepest descent for non-linear minimization problems". In: *Quarterly of Applied Mathematics* 2.3 (1944), pp. 258–261.

[55] Jack Kiefer and Jacob Wolfowitz. "Stochastic estimation of the maximum of a regression function". In: *The Annals of Mathematical Statistics* (1952), pp. 462–466.

[56] Herbert Robbins and Sutton Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[57] Grégoire Montavon, Geneviève Orr, and Klaus-Robert Müller. *Neural networks: tricks of the trade*. Vol. 7700. springer, 2012.

[58] Samuel L Smith et al. "On the origin of implicit regularization in stochastic gradient descent". In: *arXiv preprint arXiv:2101.12176* (2021).

[59] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods". In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17.

[60] Geoffrey Hinton. *Coursera Neural Networks For Machine Learning (Lecture 6)*. Coursera, Jan. 2018.

[61] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7 (2011).

[62] Andrea Vedaldi. *A convolutional neural network primer, For the Oxford C18 and AIMS Big Data courses*. Online, Sept. 2019.

[63] Henry J Kelley. "Gradient theory of optimal flight paths". In: *Ars Journal* 30.10 (1960), pp. 947–954.

[64] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[65] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.