Towards Robust Radar Perception in Autonomous Vehicles

SIMIN ZHU

Towards Robust Radar Perception in Autonomous Vehicles

Deep Learning Methods for Motion Estimation, Radar Calibration, and Scene Segmentation

Towards Robust Radar Perception in Autonomous Vehicles

Deep Learning Methods for Motion Estimation, Radar Calibration, and Scene Segmentation

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Wednesday 11 March 2026 at 17:30

by

SIMIN ZHU

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. A. Yarovoy	Delft University of Technology, promotor
Dr. F. Fioranelli	Delft University of Technology, promotor
Independent members:	
Prof. dr. D. Gavrila	Delft University of Technology
Prof. dr. G.C.H.E de Croon	Delft University of Technology
Prof. dr. A. Pandharipande	Eindhoven University of Technology / NXP, NL
Prof. dr. M. Antoniou	University of Birmingham / UK
Dr. ir. A. Coraddu	Delft University of Technology
Dr. M.A. Zuñiga Zamalloa	Delft University of Technology, reserve member

This research has been carried out at the Delft University of Technology in the Microwave, Signals, and Systems (MS3) group.

Keywords: Automotive Radar, Motion Estimation, Localization, Radar Segmentation, Deep Learning, Radar Calibration, Point Cloud.

Printed by: Proefschriftspecialist, 1506RZ Zaandam, The Netherlands.

Front & Back: Design by Simin Zhu.

ISBN/EAN: 978-94-6384-920-3 (Paperback/Softback)
ISBN/EAN: 978-94-6518-255-1 (E-Book/PDF)

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Author e-mail: simmyzhu1993@gmail.com

To my homeland, my people, and my family, I miss you deeply.

I await the dawn with hope.

Contents

List of Acronyms

Advanced Driving System	ADS
Angle of Arrival	AoA
Batch Normalization	BN
Bidirectional Gated Recurrent Unit	bi-GRU
Convolutional Neural Network	CNN
Constant False Alarm Rate	CFAR
Deep Learning	DL
Exponential Moving Average	EMA
Field of View	FoV
Frame Per Second	FPS
Frequency-Modulated Continuous-Wave	FMCW
Global Positioning System	GPS
Ground Truth	GT
Heterogeneous Sensor Fusion	HTSF
Homogeneous Sensor Fusion	HMSF
Inertial Measurement Unit	IMU
Intersection over Union	IoU
Iterative Closest Point	ICP
Kalman Filter	KF
Leave-One-Out	L1O
Light Detection And Ranging	LiDAR
Line-Of-Sight	LoS
Mean Absolute Error	MAE
Mean Squared Error	MSE
Median Absolute Error	MedAE
Millimeter-wave	mmWave
Multiple-Input Multiple-Output	MIMO
Neural Network	NN
Normal Distribution Transform	NDT
Normalized Cross-Correlation	NCC
Non-Line-Of-Sight	NLOS
Orthogonal Distance Regression	ODR
Radar Cross-Section	RCS
Radar Sensor Network	RSN
Random Sample Consensus	RANSAC
Rectified Linear Unit	ReLU
Recurrent Neural Network	RNN
Relative Trajectory Error	RTE
Root Mean Square Propagation	RMSProp

Saturated Root Mean Square Error	S-RMSE
Shared Multi-Layer Perceptron	shared-MLP
Simultaneous Localization And Mapping	SLAM
Simple Moving Average	SMA
State-Of-The-Art	SOTA
Synthetic Aperture Sonar	SAS
Weighted Least Squares	w-LSQ
Weighted Moving Average	WMA

Summary

Autonomous driving relies on the ability of vehicles to perceive and interpret their surroundings under all conditions. Among the various sensing modalities in modern perception systems, automotive radar holds a unique and indispensable position. It operates reliably in adverse weather and low-visibility conditions, directly measures radial velocities through the Doppler effect, and can detect objects hidden from the line of sight. These advantages make radar an indispensable component of any future autonomous driving stack. However, radar data are inherently sparse and noisy, and affected by artifacts such as multipath reflections and sidelobes, leading to weak geometric representations of detected objects and a high rate of false alarms. Consequently, well-established perception techniques developed for vision and lidar cannot be directly applied to radar without significant adaptation. This dissertation therefore aims to advance robust radar perception for autonomous vehicles through deep learning methods for motion estimation, radar calibration, and scene segmentation, all tailored to the unique characteristics of radar data.

This research is guided by a central question: *Can radar be elevated from a supporting sensor to a primary perception modality capable of providing robust and accurate information for automotive applications?* To address this central question, five sub-questions structure the work: (1) which perception tasks are best suited to automotive radar, (2) how deep learning can be effectively integrated with radar data and task-specific characteristics, (3) how multiple radars' data can be fused effectively, (4) how misalignment in radar extrinsic parameters can be corrected under realistic driving conditions, and (5) whether common information requirements across perception tasks can be fulfilled through a unified processing framework. These questions are addressed through four main contributions, presented in Chapters 3–6, each validated on large-scale real-world radar data.

The first contribution, in Chapter 3, is *DeepEgo*, a deep learning-based framework for vehicle ego-motion estimation using radar data only. Unlike previous approaches, *DeepEgo* operates on a single radar frame, eliminating the need for multi-frame data association and high spatial resolution. It combines a deep learning frontend with a weighted least squares backend, forming a hybrid architecture that preserves the interpretability of model-based methods while improving robustness to false positives and moving objects. Trained in a self-supervised manner using odometry as ground truth, *DeepEgo* automatically learns to assign weights to radar detections, removing the need for manual annotation. Experiments on the public *RadarScenes* dataset demonstrate approximately 50% higher accuracy and 129× faster runtime than state-of-the-art baselines, highlighting the potential of radar-only odometry as a low-cost and weather-resilient alternative to camera- or lidar-based systems.

Building on this foundation, Chapter 4 extends ego-motion estimation to multi-sensor setups through *DeepEgo+*, a framework that performs decentralized late fusion across multiple unsynchronized radars. Each radar independently produces an initial motion estimate, which is then fused by a neural network-based Kalman filter. This architecture enhances estimation accuracy while increasing robustness to sensor failures, synchronization errors,

and strong vehicle accelerations. Experimental results on the *RadarScenes* dataset show that *DeepEgo+* achieves approximately twice the accuracy of existing methods, with root mean square errors as low as $5.3\,\mathrm{cm/s}$ in translational velocity and $0.44\,\mathrm{deg/s}$ in yaw rate. The method thus establishes a scalable and synchronization-free solution for ego-motion estimation using multiple automotive radars under realistic driving conditions.

The third contribution, detailed in Chapter 5, focuses on radar extrinsic calibration under operational conditions. Accurate knowledge of parameters such as the radar's mounting angle and position is essential for reliable perception, yet these can drift over time due to vibration or minor collisions. The proposed method combines radar and inertial measurement unit (IMU) data to estimate the radar's mounting angle online, without calibration targets or predefined routes. A neural network processes radar point clouds to reject moving objects and estimate radar motion, while a measurement model compensates for IMU bias and scale errors. By exploiting the equality of lateral velocities on a rigid body, the radar's mounting angle is derived from the estimated motion and vehicle yaw rate. Experiments on the *RadarScenes* dataset demonstrate convergence within $25\,\mathrm{s}$ and a mean absolute error of about $0.01°$, surpassing existing RANSAC- and Kabsch-based approaches and confirming that accurate radar calibration can be achieved during normal driving.

Finally, Chapter 6 presents a unified perception framework that jointly performs static–moving object segmentation and ego-motion estimation using radar data only. Designed with a deep understanding of radar characteristics and task requirements, the network extracts essential spatial and temporal features from radar point clouds without additional steps such as aggregation or motion compensation. This design reduces system latency and eliminates the need for external sensors. The framework functions independently, and its outputs can support various downstream perception tasks. Experiments demonstrate state-of-the-art segmentation accuracy, with an intersection-over-union of 0.86 and an F1 score of 0.92, together with accurate odometry yielding a relative trajectory error of $1.8\,\mathrm{m}$. With only 0.15 million trainable parameters, the proposed approach shows that high-level scene understanding and motion estimation can be achieved jointly in a compact and efficient radar-only system.

Taken together, these contributions advance radar perception from methods constrained by controlled environments and impractical assumptions toward robust operation in more realistic scenarios. The developed approaches in this thesis leverage the potential of deep learning, identify radar tasks best suited for autonomous driving, and design tailored neural architectures to address their key challenges. The findings of this dissertation demonstrate that radar can serve not merely as a supporting sensor but as a primary perception modality, providing robust and accurate information essential for the future of autonomous vehicles.

Samenvatting

Autonoom rijden is afhankelijk van het vermogen van voertuigen om hun omgeving onder alle omstandigheden waar te nemen en te interpreteren. Van de verschillende sensormodaliteiten in moderne perceptiesystemen neemt automobielradar een unieke positie in. Deze sensor werkt betrouwbaar bij slecht weer en in omstandigheden met weinig zicht, meet radiale snelheden direct via het Dopplereffect en kan objecten detecteren die gedeeltelijk buiten het gezichtsveld liggen. Deze voordelen maken radar tot een onmisbaar onderdeel van toekomstige autonome voertuigen. Radardata zijn echter van nature schaars en ruisgevoelig, en worden beïnvloed door artefacten zoals meerpadreflecties en zijlobben. Dit leidt tot zwakke geometrische representaties van objecten en een hoog aantal foutieve detecties. Daardoor kunnen perceptietechnieken ontwikkeld voor camera's en lidar niet direct op radar worden toegepast zonder aanzienlijke aanpassingen. Dit proefschrift heeft daarom tot doel de radarperceptie voor autonome voertuigen te versterken door middel van deep learning-methoden voor bewegingsschatting, radarkalibratie en scènesegmentatie, afgestemd op de specifieke eigenschappen van radar data.

Het onderzoek wordt geleid door één centrale vraag: *Kan radar worden verheven van een ondersteunende sensor tot een primaire perceptiemodaliteit die robuuste en nauwkeurige informatie kan leveren voor toepassingen in de automobielsector?* Om deze vraag te beantwoorden, wordt het werk gestructureerd rond vijf deelvragen: (1) welke perceptietaken het best aansluiten bij de eigenschappen van automobielradar, (2) hoe deep learning effectief kan worden geïntegreerd met radardata, (3) hoe meerdere radars efficiënt kunnen worden gefuseerd, (4) hoe verkeerde uitlijning in radarextrinsieke parameters kan worden gecorrigeerd onder realistische rijomstandigheden, en (5) of gemeenschappelijke informatie-eisen over verschillende perceptietaken kunnen worden ingevuld via één uniform verwerkingskader. Deze vragen worden beantwoord in vier hoofdbijdragen, gepresenteerd in Hoofdstukken 3–6, gevalideerd met grootschalige radardata uit de echte wereld.

De eerste bijdrage, beschreven in Hoofdstuk 3, is *DeepEgo*, een deep learning-raamwerk voor voertuigsnelheids- en bewegingsschatting met uitsluitend radardata. In tegenstelling tot eerdere methoden werkt *DeepEgo* met één enkel radarbeeld, waardoor de noodzaak voor data-associatie tussen meerdere frames wordt geëlimineerd. Het combineert een deep learning-frontend met een gewogen kleinste-kwadraten-backend, wat resulteert in een hybride architectuur die de interpreteerbaarheid van modelgebaseerde methoden behoudt en de robuustheid tegen foutieve detecties en bewegende objecten vergroot. Het model wordt zelflerend getraind met odometrie als referentie en leert automatisch gewichten toe te wijzen aan radardetecties. Experimenten met de publieke *RadarScenes*-dataset tonen ongeveer 50% hogere nauwkeurigheid en 129× snellere verwerking dan bestaande methoden, waarmee het potentieel van radar-alleen-odometrie als een goedkope en weersbestendige oplossing wordt aangetoond.

Voortbouwend op deze basis breidt Hoofdstuk 4 de bewegingsschatting uit naar multisensorsystemen via *DeepEgo+*, een raamwerk dat gedecentraliseerde late fusie uitvoert tus-

sen meerdere niet-gesynchroniseerde radars. Elke radar genereert onafhankelijk een eerste schatting, die vervolgens wordt gecombineerd door een neurale-netwerk-Kalmanfilter. Deze architectuur verhoogt de nauwkeurigheid en robuustheid tegen sensorstoringen, synchronisatiefouten en voertuigversnellingen. Resultaten laten zien dat *DeepEgo+* ongeveer tweemaal zo nauwkeurig presteert als bestaande methoden, met fouten van slechts $5{,}3\,\mathrm{cm/s}$ in translatiesnelheid en $0{,}44\,\mathrm{deg/s}$ in giersnelheid. De methode biedt daarmee een schaalbare en synchronisatievrije oplossing voor bewegingsschatting met meerdere radars onder realistische rijomstandigheden.

De derde bijdrage, uitgewerkt in Hoofdstuk 5, richt zich op radarkalibratie onder operationele omstandigheden. Nauwkeurige kennis van parameters zoals de montagehoek en positie van de radar is essentieel voor betrouwbare perceptie, maar deze kunnen verschuiven door trillingen of kleine botsingen. De voorgestelde methode combineert radar- en traagheidsmeeteenheid (IMU)-gegevens om de montagehoek van de radar online te schatten, zonder gebruik van kalibratiedoelen of vaste routes. Een neuraal netwerk verwerkt radarpuntenwolken om bewegende objecten te onderdrukken, terwijl een meetmodel corrigeert voor IMU-bias en schaalfouten. Door gebruik te maken van de gelijkheid van laterale snelheden op een star lichaam wordt de oriëntatie van de radar afgeleid uit de geschatte beweging en de giersnelheid van het voertuig. Experimenten tonen convergentie binnen $25\,\mathrm{s}$ en een gemiddelde fout van ongeveer $0{,}01°$, beter dan bestaande RANSAC- en Kabsch-benaderingen.

Ten slotte presenteert Hoofdstuk 6 een verenigd raamwerk dat gelijktijdig statisch bewegende objectsegmentatie en voertuigsnelheidsschatting uitvoert met uitsluitend radardata. Het netwerk extraheert ruimtelijke en temporele kenmerken uit radarpuntenwolken zonder extra stappen zoals aggregatie of bewegingscompensatie. Hierdoor wordt de systeemvertraging verminderd en is geen externe sensorinformatie nodig. Het raamwerk functioneert zelfstandig, en de uitvoer kan worden gebruikt voor verschillende vervolgmodules in de perceptieketen. Experimenten tonen segmentatieprestaties op topniveau, met een intersection-over-union van $0{,}86$, een F1-score van $0{,}92$ en een relatieve trajectfout van $1{,}8\,\mathrm{m}$. Met slechts $0{,}15$ miljoen trainbare parameters laat deze aanpak zien dat scènebegrip en bewegingsschatting gezamenlijk kunnen worden bereikt in een compact en efficiënt radar-gebaseerd systeem.

Gezamenlijk brengen deze bijdragen radarperceptie van methoden die beperkt zijn tot gecontroleerde omgevingen en onpraktische aannames naar robuuste werking in realistische scenario's. De ontwikkelde benaderingen benutten het potentieel van deep learning, identificeren de radartaken die het meest geschikt zijn voor autonoom rijden en ontwerpen specifieke neurale architecturen om hun belangrijkste uitdagingen aan te pakken. De resultaten tonen aan dat radar niet slechts een ondersteunende sensor hoeft te zijn, maar kan dienen als een primaire perceptiemodaliteit die robuuste en nauwkeurige informatie levert voor de autonome voertuigen van de toekomst.

1

Introduction

I envision that future autonomous vehicles will integrate multiple complementary sensing modalities, among which automotive radar will play a critical and irreplaceable role in ensuring safety and reliability. This chapter elaborates on the motivation behind this vision by discussing the unique advantages of radar sensing within multimodal perception systems. At the same time, it acknowledges the inherent limitations of radar and identifies the key challenges that motivate the research presented in this dissertation. The chapter concludes by formulating the research questions that guide the subsequent investigations, bridging the vision of robust radar perception with solutions that are practical, reliable, and tailored to the distinct characteristics of radar data.

1

1.1. Background and Motivation

Over the past few decades, automated driving systems (ADSs) have attracted significant attention from both academia and industry [1, 2]. These technologies aim not only to improve driving comfort and efficiency, but also to enhance road safety by reducing accidents caused by human error, which remains the dominant factor in traffic incidents [3]. At a high level, an ADS can be conceptually divided into two major modules: a perception frontend and a decision-making backend. The backend is responsible for tasks such as behavior prediction, obstacle avoidance, trajectory planning, and motion control, based on the information provided by the frontend. The frontend focuses on sensing the driving environment and interpreting it from multimodal sensor data [4]. Consequently, reliable decision-making depends on robust and accurate perception, which is widely recognized as the cornerstone of autonomous driving safety.

To achieve such robust perception, vehicles typically employ multiple complementary sensors, including cameras, lidars, and radars [5, 6]. Each modality offers unique strengths but also exhibits critical limitations, especially under challenging real-world conditions. For example, cameras and lidars can provide high spatial resolution, rich semantic information, and fine geometric details of illuminated objects. However, they also suffer from notable shortcomings: cameras are highly sensitive to illumination changes, while lidar performance can degrade in adverse weather such as rain, snow, and fog [7]. These challenges raise concerns about the robustness of vision- and lidar-based perception systems in safety-critical scenarios. By contrast, millimeter-wave (mmWave) radar offers a distinct set of advantages. Radar perception is largely unaffected by poor lighting or adverse weather [8, 9], and it can detect objects that are partially occluded or even outside the direct line of sight [10]. With the development of multiple-input multiple-output (MIMO) technology, modern radars achieve improved spatial resolution with compact and cost-effective hardware compared to traditional non-MIMO systems. Moreover, radars can directly provide a variety of object features, including range, azimuth, elevation, radial velocity, and radar cross section (RCS) [11]. Compared with lidar, automotive radars are generally smaller, consume less power, and are more affordable, making them suitable for large-scale deployment in consumer vehicles [12]. In addition, the ability to directly measure radial velocity through the Doppler effect enables accurate estimation of ego-vehicle speed [13] as well as the motion of surrounding objects [14].

These strong advantages of automotive radar have motivated both industry and academia to elevate radar from an auxiliary sensing modality to a major perception source, enabling coverage of critical driving situations that other sensors may fail to handle. However, exploiting and integrating radar data is not straightforward, as radar also has several inherent limitations. For example, radar data are often sparse and have lower angular resolution (in both elevation and azimuth), making it difficult to distinguish closely spaced objects or capture fine object contours. Radar signals are also subject to multipath reflections and mutual interference, which can lead to ghost targets. In addition, radar data typically exhibit higher noise levels, which may result in missed detections of low-RCS objects such as pedestrians. These challenges limit the applicability of well-established vision- and lidar-based perception methods to radar data. Consequently, there is a strong need not only for advanced signal processing methods tailored to radar's unique characteristics, but also for carefully selected perception tasks that are less affected by radar's weaknesses while leveraging its distinctive

physical properties to improve the reliability and robustness of automated driving systems.

Motivated by these factors, this dissertation focuses on developing radar-based perception solutions for automotive applications. More specifically, it aims to move beyond traditional model-based radar signal processing methods, which often struggle to handle complex and dynamic driving scenarios without extensive manual modeling, by incorporating data-driven approaches such as deep learning into the radar processing pipeline. Recent advances in deep learning have demonstrated remarkable success in perception tasks for vision and lidar, particularly in automatically extracting robust features from high-dimensional and noisy sensor data. These developments motivate the integration of deep learning into radar perception, where learning-based methods can exploit real-world radar data to model complex scene characteristics and mitigate the negative effects caused by radar's inherent limitations.

1.2. Research Questions

The overarching goal of this dissertation is to enhance the robustness, accuracy, and practicality of radar-based perception solutions for automotive applications by leveraging deep learning techniques. To this end, the research is guided by the following questions:

- **Q1: Among the various radar-based perception tasks in automotive applications, which are best suited to the characteristics of radar data?** Over the past decade, advances in radar sensing technology have enabled the exploration of diverse perception tasks for automotive applications, demonstrating the feasibility of radar-based solutions. However, due to inherent radar limitations and the complexity of driving scenarios, many studies have been conducted in controlled environments or unrealistic conditions. Consequently, the natural next step is to first identify the perception tasks that are best aligned with radar characteristics and then address the associated limitations that hinder real-world implementation.

- **Q2: What advantages does deep learning bring to the perception pipeline? At which stages should it be integrated, and in what manner?** Although deep learning techniques have been developed over several decades and have demonstrated remarkable performance gains in camera- and lidar-based perception tasks, their application to radar perception has only recently begun. Early research primarily focused on directly adapting well-established approaches from other sensing modalities to radar for comparable tasks in automotive applications. This was followed by efforts to employ increasingly complex feature extraction backbones to maximize performance, often without considering computational costs, fundamental bottlenecks, or the unique characteristics of radar data. Consequently, the next step is to investigate how radar-specific knowledge can be effectively integrated with deep learning techniques that exploit radar's strengths, mitigate its limitations, and remain practical for real-world deployment.

- **Q3: Does the use of multiple radars improve perception performance? What are the requirements, and how can effective radar sensor fusion be achieved?** Modern autonomous vehicles are often equipped with multiple radar sensors, raising the question of how much benefit can be gained from additional sensing and which fusion strategies are most effective. While prior studies have explored radar fusion to some

extent, it is equally important to consider practical challenges in real-world deployments. For example, radar sensors may have non-overlapping fields of view, operate with unsynchronized sampling, or produce outputs containing errors. Addressing these issues is essential for establishing radar as a primary perception modality in future autonomous driving systems.

- **Q4: To what extent does misalignment in radar extrinsic parameters impact perception performance, and can such misalignment be calibrated under operational driving conditions?** In practice, automotive radars are mounted on moving vehicles with fixed positions and pointing directions. The parameters that describe the relative position and orientation between the radar coordinate system and the vehicle coordinate system are referred to as extrinsic parameters. These are typically measured during vehicle assembly when the radar is first installed. However, due to vibration, material aging, or collisions, these parameters can drift over time. Previous studies have investigated this issue and proposed various remedies, yet existing methods often rely on controlled environments, predefined driving routes, specialized radar targets, or restrictions on sensor placement. When these conditions are not satisfied, such methods experience significant performance degradation. Therefore, it is important to develop a more practical approach for extrinsic calibration that operates under fewer constraints while providing higher robustness and accuracy.

- **Q5: Across radar perception tasks, what are the common requirements and assumptions? Can these be addressed through a unified processing step?** Radar has been applied to a variety of automotive perception tasks, including mapping, tracking, localization, classification, and segmentation. When focusing on a specific task, however, assumptions or prior knowledge are often required. Satisfying these prerequisites is not always straightforward and may depend on external sensors or complex preprocessing of radar data. To preserve the integrity and independence of the radar perception chain, it is important to identify the common information required across tasks and to investigate whether a unified solution can fulfill these needs.

By addressing these research questions, this dissertation identifies the radar perception tasks most suitable for automotive applications and systematically overcomes key limitations of existing radar-based methods by integrating the advantages of deep learning techniques. Furthermore, it demonstrates the potential of radar as a central sensing modality for robust, accurate, and reliable autonomous driving.

1.3. Main Contributions

The main contributions of this PhD research can be briefly summarized as follows:

- (**Addresses Q1 and Q2**) A novel radar-only method, named *DeepEgo*, is proposed for instantaneous vehicle ego-motion estimation from a single radar frame. Its key innovation lies in three aspects. First, *DeepEgo* adopts a hybrid architecture that combines deep learning with weighted least squares, leveraging the strengths of both data-driven and model-based approaches. Second, instead of relying on conventional

scan-matching techniques that require multiple radar frames and high spatial resolution, *DeepEgo* exploits the inherent relationship between observed radial velocities and the underlying sensor motion. This design simplifies feature extraction, avoids overly complex network structures, and enables ego-motion estimation using only a single radar frame. Third, unlike prior works that depend on majority-inlier assumptions, *DeepEgo* employs neural networks to extract spatial sinusoidal patterns from radar data, ensuring robustness even under high outlier ratios. Experiments on the RadarScenes dataset [15] demonstrate approximately 50% higher accuracy and about 129× faster runtime compared to state-of-the-art baselines. The novelty and practical value of this contribution have also been recognized through a granted patent.

- (**Addresses Q2 and Q3**) The second contribution extends *DeepEgo* to multi-radar systems for robust vehicle ego-motion estimation. The proposed method, named *DeepEgo+*, adopts a decentralized late-fusion framework in which each radar independently generates an initial motion estimate, which is subsequently fused by a neural network–based Kalman filter. This design enables operation with unsynchronized radar networks, ensures resilience to partial sensor failures, and significantly mitigates the effects of vehicle acceleration. Experiments on the RadarScenes dataset demonstrate a further 50% improvement in accuracy over state-of-the-art baselines (including *DeepEgo*), achieving a translational velocity RMSE of 5.3 cm/s and a rotational velocity RMSE of 0.44 deg/s. The novelty and practical value of this contribution have also been recognized through a granted patent.

- (**Addresses Q2 and Q4**) The third contribution introduces a novel signal processing pipeline that combines radar and inertial measurement unit (IMU) data to achieve accurate and reliable radar mounting angle estimation under realistic driving scenarios. Unlike previous studies, the method employs neural networks to process sparse and noisy radar measurements, reject detections from moving objects, and estimate radar motion. A dedicated measurement model is further proposed to correct IMU bias and scale factor errors. Leveraging vehicle kinematics, the radar mounting angle is then inferred from the estimated radar motion and the vehicle's yaw rate. Evaluation on the RadarScenes dataset shows convergence within 25 seconds and a mean absolute error below 0.02 degrees, significantly outperforming state-of-the-art baselines. This contribution demonstrates that automotive radar mounting angles can be accurately estimated in complex real-world traffic conditions, without the need for controlled environments, calibration targets, or specially designed driving routes.

- (**Addresses Q1, Q2, and Q5**) The final contribution is a unified framework that jointly performs static–moving object segmentation and vehicle ego-motion estimation using radar data only. This method is motivated by the observation that most radar perception tasks require knowledge of either the vehicle ego-velocity, the location of static measurements, or the location of moving objects. Consequently, it addresses an important gap in the radar perception chain. Furthermore, the success of jointly performing segmentation and odometry relies on a careful analysis of radar data and effective neural network design. As a result, the proposed framework operates independently, accurately, and robustly with a lightweight, compact, yet powerful architecture. Experiments on the RadarScenes dataset demonstrate state-of-the-art perfor-

mance, with an IoU of 0.86 and F1-score of 0.92 for segmentation, and an RTE_50 of 1.8 m for odometry. This marks the first radar-only solution to solve both tasks simultaneously, advancing effective and efficient radar perception for autonomous driving.

Taken together, these contributions address the research questions in a systematic manner and highlight the potential of radar to serve as a central and dependable sensing modality for future autonomous driving systems.

1.4. Thesis Outline

The remaining part of this thesis is organized as follows:

Chapter 2 – Preliminaries: Radar Fundamentals and Dataset. This chapter establishes the technical foundation for the remainder of the dissertation. It first introduces the fundamental steps in automotive radar signal preprocessing, describing the processing chain that transforms raw intermediate-frequency data into radar point clouds through range–Doppler processing, angle-of-arrival (AoA) estimation, and CFAR-based detection. It then provides an overview of the public radar dataset that serves as the primary source of experimental data throughout this work. Finally, it presents the evaluation metrics employed to assess perception performance in subsequent chapters.

Chapter 3 – *DeepEgo*: Radar-Based Vehicle Ego-Motion Estimation. This chapter presents the first core contribution of the dissertation. It addresses research questions Q1 and Q2 by demonstrating how radar data can be exploited for instantaneous vehicle motion estimation and how deep learning techniques can be effectively integrated into the estimation pipeline. The results of this chapter have been published in the following works:

- S. Zhu, F. Fioranelli, and A. Yarovoy, "Radar-only Instantaneous Ego-motion Estimation Using Neural Networks," 2023 20th European Radar Conference (EuRAD), Berlin, Germany, 2023, pp. 201-204.
- S. Zhu, A. Yarovoy, and F. Fioranelli, "DeepEgo: Deep Instantaneous Ego-Motion Estimation Using Automotive Radar," in IEEE Transactions on Radar Systems, vol. 1, pp. 166-180, 2023.
- S. Zhu, A. Yarovoy, F. Fioranelli, S. Ravindran, "An apparatus for determining ego-motion" in US Patent Application, patent filed in 2023, WO2024183926A1.
- S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "DeepEgo+: Unsynchronized Radar Sensor Fusion for Robust Vehicle Ego-Motion Estimation," in IEEE Transactions on Radar Systems, vol. 3, pp. 483-497, 2025.

Chapter 4 – *DeepEgo+*: Multi-Radar Fusion for Robust Motion Estimation. Building on Chapter 3, this chapter extends vehicle ego-motion estimation to multi-radar setups, thereby addressing research questions Q2 and Q3. The proposed method, *DeepEgo+*, enhances system robustness against sensor failures, high outlier ratios, and vehicle acceleration, while providing accurate estimation performance and scaling naturally to larger radar networks. The results of this chapter have been published in the following work:

- S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "DeepEgo+: Unsynchronized Radar Sensor Fusion for Robust Vehicle Ego-Motion Estimation," in IEEE Transactions on Radar Systems, vol. 3, pp. 483-497, 2025.

- S. Zhu, A. Yarovoy, F. Fioranelli, S. Ravindran, L. Chen, "Methods and devices for multi-radar, multi-frame ego-motion estimation" – application number: PCT/ US2024/ 031392, filed in 2024, accessible as WO2025250125A1.

Chapter 5 – Radar Mounting Angle Estimation under Operational Driving Conditions. This chapter proposes a method for online radar mounting angle estimation by combining radar and inertial measurement unit (IMU) data. It addresses research questions Q2 and Q4 and represents the first demonstration of practical radar extrinsic calibration in unconstrained urban driving scenarios. This work has been submitted to:

- S. Zhu, S. Ravindran, C. Lihui, A. Yarovoy, and F. Fioranelli, "Radar Mounting Angle Estimation in Operational Driving Conditions," in IEEE Transactions on Radar Systems. (under review)

Chapter 6 – Toward Holistic Radar Perception: Simultaneous Segmentation and Odometry. This chapter presents a novel multi-task framework that jointly performs point cloud segmentation and vehicle ego-motion estimation. By coupling these tasks within a lightweight neural network, the method delivers both scene understanding and vehicle odometry simultaneously. It addresses research questions Q1, Q2, and Q5, demonstrating the feasibility of radar-only holistic perception and highlighting future directions for radar-based perception systems. This work has been submitted to:

- S. Zhu, S. Ravindran, A. Yarovoy, and F. Fioranelli, "Redefining Radar Segmentation: Simultaneous Static-Moving Segmentation and Ego-Motion Estimation using Radar Point Clouds," in IEEE Transactions on Radar Systems. (under review)

Chapter 7 – Conclusions. The final chapter summarizes the main findings and contributions of this dissertation. It consolidates the results from Chapters 3 to 6, demonstrating how radar-only solutions can enable accurate and robust ego-motion estimation, multi-radar fusion, extrinsic parameter calibration, and point cloud segmentation. The chapter also revisits the research questions introduced in Chapter 1, outlining how each has been addressed, and discusses the broader implications of the findings for the role of automotive radar in autonomous driving. Finally, recommendations are provided for future research directions.

2

Preliminaries: Radar Fundamentals and Dataset

This chapter provides the technical background and definitions necessary to support the research presented in the subsequent chapters. It first introduces the fundamental principles of automotive radar signal processing, outlining the key steps that transform raw intermediate-frequency data into radar point clouds. Although the public dataset used in this dissertation provides only processed point clouds, a concise overview of the preprocessing pipeline is included to present a clearer understanding of how such data are generated. The chapter then describes the public radar dataset that serves as the primary experimental foundation for this work, followed by the definition of the evaluation metrics used to assess the proposed perception solutions.

2.1. Radar Signal Preprocessing Pipeline

Since this dissertation builds upon a public automotive radar dataset that provides only radar point clouds, it is important to understand how such data are generated. To this end, this section presents the signal model and processing steps of a common automotive radar system, outlining the key stages from raw signal acquisition to target detection. For illustration, the discussion focuses on a conventional Time-Division Multiplexing (TDM) Multiple-Input Multiple-Output (MIMO) radar employing Frequency-Modulated Continuous Wave (FMCW) modulation. The material presented in this section is not novel but serves to provide a complete overview of the radar signal chain preceding point cloud generation. The descriptions and formulations summarized here are based primarily on the four research articles [16–19]. Nevertheless, similar signal processing fundamentals are also widely available in radar textbooks, academic lectures, and doctoral theses.

2.1.1. FMCW Radar Signal Model

A FMCW radar transmits a continuous chirp signal whose frequency increases linearly over time. Let f_0 denote the starting frequency, B the swept bandwidth, and T_c the chirp duration. The chirp slope is then given by $S = B/T_c$. The transmitted signal, expressed in complex (analytic) passband representation, can be written as:

$$s_{\text{tx}}(t) = \exp\left\{j2\pi\left(f_0\, t + \frac{1}{2} S\, t^2\right)\right\}, \quad 0 \le t < T_c, \tag{2.1}$$

where t denotes the fast time within one chirp duration. Assuming that the transmitted waveform reflects from a relatively stationary point target and returns to the receiver after a round-trip propagation delay $\tau = 2r/c$ for the target at range r (with c denoting the speed of light), the received signal can be modeled as a time-delayed copy of the transmitted signal:

$$s_{\text{rx}}(t) = \alpha \exp\left\{j2\pi\left[f_0(t-\tau) + \frac{1}{2} S(t-\tau)^2\right]\right\}, \tag{2.2}$$

where α is a complex amplitude proportional to the target's radar cross-section (RCS) and inversely proportional to the target's range. To avoid sampling or processing signals directly at the carrier frequency, the received signal is mixed (multiplied) with the conjugate of the transmitted signal to generate a much lower intermediate frequency (IF) or beat signal. After low-pass filtering, the resulting baseband signal can be expressed as:

$$s_{\text{IF}}(t) = A \exp\{j(-2\pi f_b t + \phi)\}, \tag{2.3}$$

where $f_b = S\tau$ is the beat frequency proportional to the target range, A is the signal amplitude, and ϕ is a constant phase term (including propagation and hardware offsets).

2.1.2. Range Processing

Having established the radar signal model, the next step is to extract range information from the IF signal $s_{\text{IF}}(t)$. This is possible because the beat frequency f_b is directly proportional to the target range. Therefore, by identifying the dominant frequency component of the IF signal, the distance of the target from the radar can be estimated. However, since the IF signal is still an analog signal, it must be converted into discrete samples before any digital

processing can be performed. This is achieved by sampling the signal using an analog-to-digital converter (ADC) at a sampling rate of f_s that satisfies the Nyquist criterion. The resulting discrete-time (DT) signal over the chirp duration T_c contains $N_s = f_s T_c$ samples and can be expressed as:

$$x[n] = s_{\text{IF}}\left(\frac{n}{f_s}\right) = A\exp\{j(-2\pi f_b n/f_s + \phi)\}, \quad 0 \le n < N_s. \tag{2.4}$$

Equation 2.4 describes a single-target scenario. In a practical environment, multiple reflecting objects are typically present, each producing a distinct beat frequency. Therefore, the sampled signal is the superposition of multiple components, each corresponding to a different target:

$$x[n] = \sum_{i=1}^{N_t} A_i \exp\left\{j\left(-2\pi f_{b,i} n/f_s + \phi_i\right)\right\}, \tag{2.5}$$

where N_t is the number of targets, and A_i, $f_{b,i}$, and ϕ_i denote the amplitude, beat frequency, and constant phase of the i-th target, respectively. The discrete Fourier transform (DFT) of $x[n]$ is then computed as:

$$X[k_r] = \sum_{n=0}^{N_s-1} x[n]\, e^{-j2\pi k_r n/N_s}, \quad 0 \le k_r < N_s, \tag{2.6}$$

where k_r is the discrete frequency index. The magnitude spectrum $|X[k_r]|$ reveals peaks at frequency bins corresponding to target beat frequencies, producing a one-dimensional range profile for each transmitted chirp. If the beat frequency f_b is associated with the frequency bin k_r, i.e., $f_b = k_r f_s/N_s$, the corresponding target range of bin k_r can be computed as:

$$r_k = \frac{c\tau}{2} = \frac{cf_b}{2S} = \frac{ck_r f_s}{2SN_s} = \frac{c}{2B}k_r. \tag{2.7}$$

This expression also indicates that the range resolution is determined by the chirp bandwidth B, given by $\Delta r = c/(2B)$. A larger sweep bandwidth therefore yields finer range discrimination. For example, a bandwidth of $B = 1$ GHz corresponds to a theoretical range resolution of approximately 0.15 m.

2.1.3. Doppler Processing

Sections 2.1.1 and 2.1.2 described the process of range estimation under the assumption that all point targets are stationary relative to the radar line-of-sight, i.e., the relative radial velocity is zero. In real-world driving scenarios, however, not only may surrounding objects be in motion, but the radar-equipped vehicle itself is typically moving. As a result, the relative radial velocity between the radar and each target is generally nonzero. Knowledge of the relative radial velocity is of great importance: in automotive applications, it enables moving object tracking, road-user classification, and vehicle ego-motion estimation. One of the key advantages of radar over conventional optical sensors is precisely this ability

to measure radial velocity directly through the Doppler effect. This process, referred to as Doppler processing, involves coherent processing of multiple transmitted and received chirps. Starting from the m-th received chirp, and considering a point target moving away from the radar with a relative radial velocity v, the received signal can be expressed as:

$$s_{\text{rx}}(t,m) = \alpha \exp\left\{j2\pi\left[f_0(t-\tau(t,m)) + \frac{1}{2}S(t-\tau(t,m))^2\right]\right\},\ m = 0,\dots,M-1, \quad (2.8)$$

where M is the total number of chirps. The time-varying round-trip delay $\tau(t,m)$ can be expressed as:

$$\tau(t,m) = \frac{2r_0}{c} + \frac{2vT_c}{c}m + \frac{2v}{c}t \triangleq \tau_m + bt, \quad (2.9)$$

with

$$\tau_m = \frac{2}{c}(r_0 + vT_c m), \qquad b = \frac{2v}{c} \ll 1. \quad (2.10)$$

Here, τ_m denotes the delay due to the initial target range r_0 and the range change accumulated over the previous m chirps, while bt denotes the additional round-trip delay within the current chirp. After mixing with the transmitted signal, the IF signal can be expressed as:

$$s_{\text{IF}}(t,m) = s_{\text{rx}}(t,m)\, s_{\text{tx}}^*(t) = \alpha \exp[j2\pi\, \Phi(t,m)], \quad (2.11)$$

with

$$\Phi(t,m) \triangleq -f_0\tau(t,m) - St\tau(t,m) + \frac{1}{2}S\,\tau(t,m)^2. \quad (2.12)$$

Based on Equation 2.9, $\Phi(t,m)$ can be divided into a constant, a linear, and a quadratic term in t:

$$\begin{aligned} \text{Constant: } & -f_0\tau_m + \frac{1}{2}S\tau_m^2, \\ \text{Linear: } & (-f_0 b - S\tau_m + S\tau_m b)\, t, \\ \text{Quadratic: } & (-Sb + \frac{1}{2}Sb^2)\, t^2. \end{aligned} \quad (2.13)$$

Here, the constant term sets a per-chirp phase, the linear term sets the instantaneous frequency within the chirp, and the quadratic term captures intra-chirp range–Doppler coupling. Since b is extremely small, the quadratic term and the $S\tau_m b$ term in the linear coefficient can be neglected. The IF signal then simplifies to:

$$s_{\text{IF}}(t,m) \approx \alpha \exp\left\{j2\pi\left[(-f_D - S\tau_m)t - f_0\tau_m + \frac{1}{2}S\tau_m^2\right]\right\}, \quad (2.14)$$

with f_D denoting the Doppler frequency:

$$f_D \triangleq \frac{2v}{\lambda} = f_c\frac{2v}{c} = f_c b \approx f_0 b. \quad (2.15)$$

where f_c is the center/carrier frequency and the approximation follows from $B \ll f_c$ (hence $f_0 \approx f_c$). Under the no range migration assumption, it is normally assumed that during the coherent processing interval, a moving target remains in the same range bin:

$$vT_cM \ll \Delta r = \frac{c}{2B}. \tag{2.16}$$

With this assumption, the IF signal can be further simplified as:

$$s_{\mathrm{IF}}(t,m) \approx \alpha \exp\{-j2\pi(f_b + f_D)t - j2\pi f_D T_c m + \phi\}, \tag{2.17}$$

where $f_b = 2Sr_0/c = S\tau_0$. Similarly, when considering N_t point targets, the IF signal can be written as:

$$s_{\mathrm{IF}}(t,m) \approx \sum_{i=1}^{N_t} \alpha_i \exp\{-j2\pi(f_{b,i} + f_{D,i})t - j2\pi f_{D,i} T_c m + \phi_i\}, \tag{2.18}$$

From Equation 2.18, it can be seen that the instantaneous frequency of the IF signal is slightly shifted by the Doppler frequency, which can be interpreted as a range–Doppler coupling. In principle, the Doppler frequency could be estimated directly from this frequency shift, as it slightly offsets the beat frequency within each chirp. However, in practical automotive systems, f_D is much smaller than f_b and the range-FFT frequency resolution is usually insufficient to resolve the offset reliably. Therefore, instead of estimating f_D directly from the beat frequency, Doppler estimation is conventionally performed across multiple chirps through coherent processing in the slow-time domain. This approach exploits the fact that the Doppler effect introduces a progressive phase change between consecutive chirps. However, since each target lies at a different range bin, the Doppler estimation becomes range-dependent. Hence, Doppler estimation is performed after first applying ADC sampling and the DFT along the fast-time dimension (as discussed in Section 2.1.2), followed by an M-point DFT along the slow-time dimension (across chirps) to obtain the Doppler spectrum:

$$X_D[k_r, \ell] = \sum_{m=0}^{M-1} X_m[k_r]\, e^{-j2\pi \ell m/M}, \tag{2.19}$$

where $X_m[k_r]$ is the complex amplitude at range bin k_r from the m-th chirp, and ℓ denotes the Doppler frequency bin index. The corresponding Doppler frequency and radial velocity at bin ℓ can then be computed as:

$$f_D = \frac{\ell}{MT_c}, \qquad v = \frac{\lambda\, \ell}{2MT_c}. \tag{2.20}$$

The maximum unambiguous Doppler frequency and corresponding radial velocity are determined by the chirp repetition interval T_c (assumes no idle time) as:

$$f_{D,\max} = \frac{1}{2T_c}, \qquad v_{\max} = \frac{\lambda}{4T_c}. \tag{2.21}$$

Increasing the number of chirps M improves the velocity resolution, while decreasing the chirp repetition interval T_c extends the unambiguous velocity range. In summary, this section has shown that coherent processing of multiple chirps provides an accurate estimate of target velocity through Doppler analysis, complementing the range information extracted from individual chirps.

2.1.4. Angle of Arrival Estimation

In addition to range and radial velocity, automotive radars can also provide angular information of detected targets, obtained through the process known as angle-of-arrival (AoA) estimation. This section first introduces the basic principles of AoA estimation using a single-transmit, multi-receive antenna configuration, showing how the angular information of the reflected signal is encoded in the phase progression across spatially separated receive antennas. It then discusses the limitations of this configuration in terms of angular resolution and explains how TDM-MIMO can be used to synthesize a larger virtual aperture, thereby mitigating these limitations. Consider a radar system with one transmit antenna and N_{rx} receive antennas arranged linearly with uniform inter-element spacing d_{rx}. The array aperture size D can be calculated as:

$$D = (N_{\text{rx}} - 1)d_{\text{rx}}. \tag{2.22}$$

For a point target located at angle θ and distance r, it is often assumed that the target satisfies the far-field (plane-wave) condition, which requires:

$$r \gg \frac{2D^2}{\lambda}. \tag{2.23}$$

This assumption is typically valid for automotive applications, since array apertures are only a few centimeters wide while targets are located several meters to hundreds of meters away. Under this assumption, the reflected wavefront can be approximated as planar across the receive array. As a result, each antenna receives a delayed version of the same signal, and the time delay between adjacent antennas is constant. Because of the impinging angle θ, the total propagation distance of the reflected wave differs linearly across the array. Consequently, the propagation delay between the reference (first) receive antenna and the p-th antenna is:

$$\tau_p = \frac{d_{\text{rx}} \sin\theta}{c} p, \qquad p = 0, 1, \dots, N_{\text{rx}} - 1. \tag{2.24}$$

For example, if the signal arriving at the first antenna is:

$$s_{\text{ref}}(t) = e^{j2\pi f_c t}, \tag{2.25}$$

then the signal received at the p-th antenna, delayed by τ_p, is:

$$s_p(t) = e^{j2\pi f_c (t-\tau_p)} = e^{j2\pi f_c t} e^{-j2\pi f_c \tau_p} = e^{j2\pi f_c t} e^{-j2\pi \frac{d_{\text{rx}} \sin\theta}{\lambda} p}. \tag{2.26}$$

Hence, each receive antenna observes the same waveform but with a phase shift proportional to its position along the array and to the sine of the target's azimuth angle. Following

the baseband signal model introduced earlier, after range and Doppler processing, the received complex baseband signal at the p-th antenna for a given range–Doppler bin (k_r, ℓ) can be expressed as:

$$x_p[k_r, \ell] = A_{k_r,\ell} \exp\left\{-j2\pi \frac{d_{\text{rx}} \sin\theta}{\lambda} p\right\}, \tag{2.27}$$

where $A_{k_r,\ell}$ denotes the complex amplitude of the reflection at (k_r, ℓ). The exponential term represents a spatial phase progression across the array elements, which encodes the AoA information. Similar to before, a simple and efficient method to estimate the AoA is to apply DFT across the antenna elements:

$$X_A[k_r, \ell, q] = \sum_{p=0}^{N_{\text{rx}}-1} x_p[k_r, \ell]\, e^{-j2\pi qp/N_{\text{rx}}}, \tag{2.28}$$

where q is the angular bin index. The corresponding azimuth angle of the q-th bin is obtained by matching the spatial frequency of the arriving wave to the discrete spatial frequency of the DFT bins, i.e.,

$$\frac{\sin\theta_q}{\lambda} = \frac{q}{N_{\text{rx}} d_{\text{rx}}} \implies \sin\theta_q = \frac{\lambda q}{N_{\text{rx}} d_{\text{rx}}}, \tag{2.29}$$

where $1/d_{\text{rx}}$ represents the spatial sampling frequency determined by the array sampling interval d_{rx}. Therefore, to avoid aliasing, the following Nyquist criterion must be satisfied:

$$-\frac{1}{2d_{\text{rx}}} < \frac{\sin\theta_q}{\lambda} < \frac{1}{2d_{\text{rx}}} \implies d_{\text{rx}} \le \frac{\lambda}{2}. \tag{2.30}$$

As for the angular resolution, it is derived from the first-null beamwidth of the array factor (AF) of a uniform linear array (ULA), which is approximately:

$$\Delta\theta \approx \frac{2\lambda}{(N_{\text{rx}} - 1)\, d_{\text{rx}} \cos\theta} \tag{2.31}$$

As can be seen, increasing the number of receive antennas N_{rx} improves the angular resolution, but doing so directly increases hardware cost and physical array size. To overcome this limitation and enhance spatial resolution without significantly increasing hardware complexity, MIMO architectures are widely employed in automotive radars. Unlike the previous single-transmitter configuration, a MIMO radar employs multiple transmit and receive antennas. For illustration purposes, this section assumes that the MIMO array is arranged linearly and that each transmitter emits chirps sequentially, following a time-division multiplexing (TDM) scheme. Under this assumption, while the spacing between the N_{rx} receive antennas (d_{rx}) follows the same criterion as Equation 2.30 to avoid aliasing, the spacing between the N_{tx} transmit antennas (d_{tx}) is chosen as:

$$d_{\text{tx}} = N_{\text{rx}} d_{\text{rx}}. \tag{2.32}$$

Under this configuration, when each transmitter is activated sequentially and the returned signal is captured by all receive antennas, a total of $N_{\text{tx}} N_{\text{rx}}$ received signals are

collected, each corresponding to a unique transmit–receive antenna pair. These signals preserve the same spatial phase progression described earlier in Equation 2.27. In other words, this is equivalent to forming a virtual linear array with one transmit antenna and $N_{\text{tx}}N_{\text{rx}}$ receive antennas, which effectively expands the aperture and improves angular resolution. More importantly, under this configuration, the same DFT-based estimation can be directly applied to the virtual array to compute the angular information of the detected object.

2.1.5. CFAR-based Detection

After the range, Doppler, and angle estimation stages, the received radar signals from multiple chirps and receive antennas are typically represented as a 3D range–Doppler–angle data cube, in which each cell corresponds to the reflected signal power at a specific range bin, Doppler (or radial velocity) bin, and azimuth angle bin. To determine which of these cells correspond to true targets rather than noise or artifacts (e.g., sidelobes resulting from limited DFT), a detection stage is required. In automotive radars, this is most commonly achieved using the constant false alarm rate (CFAR) detection algorithm [16], which also constitutes the final step before generating radar point clouds. CFAR detectors operate on the magnitude or power of the complex-valued radar data cube[1], and adaptively determine a detection threshold for each cell under test (CUT) based on the local noise level. This adaptive approach maintains a constant probability of false alarm across the entire cube, even when the background power varies with range, Doppler, or angle. Accordingly, the basic principle of CFAR is to compare the power value P_{CUT} of the target cell against a threshold computed from its neighboring cells. Since the range–Doppler–angle data cube after the spatial DFT (Equation 2.28) provides the complex amplitude $X_A[k_r, \ell, q]$ at each range, Doppler, and angle bin, the corresponding power of the CUT is obtained as:

$$P_{\text{CUT}} = \left|X_A[k_r, \ell, q]\right|^2 . \tag{2.33}$$

The general CFAR decision rule can be expressed as:

$$P_{\text{CUT}} \gtrless_{H_0}^{H_1} T_{\text{CFAR}} = \gamma\, \hat{P}_{\text{noise}}, \tag{2.34}$$

where H_0 and H_1 denote the hypotheses of noise-only and target-present, respectively, $\hat{P}_{\text{noise}}$ is the estimated average noise power surrounding the CUT, and γ is a scaling factor (threshold multiplier) chosen to maintain the desired false alarm probability P_{FA}. To estimate the local noise level, a typical CFAR algorithm employs a sliding window (or cube) consisting of three regions: a single CUT at the center, a surrounding set of guard cells to prevent signal leakage from the target into the noise estimate, and a larger set of training cells used to estimate the background noise power. Once the training cells are selected, one widely used approach to estimate the background noise level is the cell-averaging method (CA-CFAR) [20]:

$$\hat{P}_{\text{noise}} = \frac{1}{N}\sum_{i=1}^{N} P_i, \tag{2.35}$$

[1] For conventional energy-based detection, CFAR operates on magnitude/power and the phase information is typically not exploited in the decision process.

where P_i denotes the power of the i-th training cell and N is the number of training cells. In general, the threshold multiplier γ depends on the assumed noise statistics and the desired false alarm rate P_{FA}. For CA-CFAR, assuming exponentially distributed noise, γ is given by:

$$\gamma = N\left(P_{\mathrm{FA}}^{-1/N} - 1\right). \tag{2.36}$$

After applying the CFAR test (Equation 2.34) to all range–Doppler–angle bins, a binary 3D detection map is generated, where H_1 indicates target presence and H_0 indicates noise. The radar point cloud is then obtained by extracting all detections corresponding to H_1. Each point in the cloud carries the estimated object attributes such as range, radial velocity, azimuth angle, and, if a planar 2D array is used instead of a simpler 1D one, elevation angle as well.

In summary, this section introduced the basic principles of CFAR-based detection and demonstrated how the previously estimated range–Doppler–angle cube can be converted into radar point clouds. It should be noted that, beyond the classical CFAR approach discussed here, numerous advanced CFAR algorithms exist in the literature such as the ordered-statistic CFAR (OS-CFAR) [21], neural network-based CFAR [22], and other modern variants [23]. A detailed review of these methods lies beyond the scope of this dissertation and is left to the interested reader.

2.2. Radar Dataset

To ensure reproducibility and enable fair benchmarking across the studies presented in this dissertation, it is essential to employ a consistent dataset. For this purpose, all experiments in this dissertation were conducted using the publicly available *RadarScenes* dataset [15]. RadarScenes is a large-scale real-world automotive radar point cloud dataset collected with four 77 GHz radar sensors mounted on a passenger vehicle. The radar mounting scheme, coordinate frames, and sign conventions are illustrated in Figure 2.1. The dataset covers more than 100 km of driving in diverse environments (urban streets, country roads, and highways) and under varying traffic and weather conditions. It comprises 158 sequences with over 7500 manually annotated road users from eleven object classes, including cars, trucks, buses, pedestrians, and cyclists. Each radar detection point contains the following features: range, angle of arrival (AoA), radial (Doppler) velocity[2], and radar cross section (RCS). These features make RadarScenes particularly valuable for radar-centric perception tasks and fundamentally different from datasets collected with optical sensors, which typically lack Doppler and RCS information.

Although the RadarScenes dataset comprises 158 sequences in total, this dissertation employs a subset of 64 sequences (approximately 79 km of driving). The selection was made to ensure that the data are both representative and challenging for the tasks investigated here. In many of the available sequences, the ego-vehicle is largely stationary while observing moving road users nearby. Such scenes are valuable for object detection, tracking, or classification, but are not well suited for ego-motion estimation, calibration, or joint segmentation and ego-motion estimation, where vehicle dynamics are essential. To avoid

[2] Following the RadarScenes convention, the sign of radial velocity is defined as positive when the target is moving away from the radar.

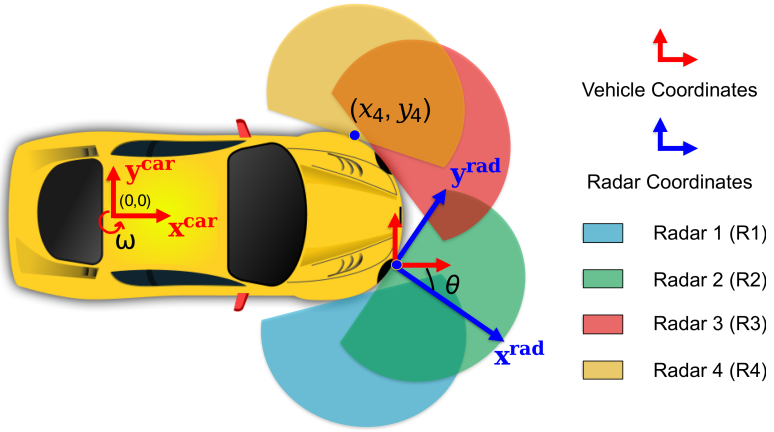


Figure 2.1: Illustration of the vehicle and the four automotive radars used for collecting the RadarScenes dataset [15], together with the coordinate frames and sign conventions adopted throughout this thesis. The vehicle coordinate frame is defined with x^{car} pointing forward and y^{car} pointing to the left, and a positive yaw rate ω corresponding to counter-clockwise rotation. A representative radar coordinate frame is shown, with x^{rad} aligned with the radar boresight and the radar mounting angle θ defined as the signed rotation from x^{car} to x^{rad}, where positive θ corresponds to a counter-clockwise rotation (right-hand rule). The vehicle-frame origin $(0, 0)$ is located at the rear-axle center, with the z-axis pointing upward, and the mounting position of the i-th radar is denoted by (x_i, y_i) and expressed in the vehicle frame. Each radar operates in near-range mode with a maximum detection range of 100 m, a field of view (FoV) of approximately $\pm 60°$, a range resolution of 0.15 m, an angular resolution of $0.5°$ at boresight, and a radial velocity resolution of 0.1 km/h. With respect to their pointing directions, Radar 1 and Radar 4 are side-looking and tilted $85°$ outwards, while Radar 2 and Radar 3 are front-facing and tilted $25°$ outwards. Unless stated otherwise, all chapters follow the coordinate definitions and sign conventions illustrated in this figure.

trivial cases, only sequences with an ego-vehicle driving distance greater than 500 m were therefore selected.

In addition to the radar measurements, the dataset also provides metadata and ground truth (GT) information. The four radars operate independently and are not hardware synchronized; each radar point cloud (or radar frame) is stored with its own timestamp referenced to a common logging system. This enables approximate temporal alignment across radars, camera images, and GT vehicle motion, although small offsets remain, which are relevant for multi-radar fusion studies. The GT trajectory of the ego-vehicle is obtained by fusing high-precision differential GPS (DGPS) with vehicle odometry data (wheel encoders and inertial sensors). In the remaining chapters, these reference data are used both as supervision for neural network training and as GT for performance evaluation. For more details on the dataset, the reader is referred to [15].

2.3. Evaluation Metrics

In addition to employing a consistent dataset, fair and comprehensive evaluation also requires well-defined performance metrics. To this end, two sets of task-specific metrics are introduced to assess both the proposed methods and relevant approaches from the literature. These metrics are designed to capture the distinctive characteristics of each perception

task and to ensure consistent and objective comparisons across experiments. The first set of metrics is used for vehicle ego-motion estimation, as detailed below:

- **Root Mean Square Error (RMSE)** computes the square root of the mean of squared differences between GT vehicle motion and estimated vehicle motion. Due to the squaring operation, it places greater weight on larger deviations, which makes it particularly effective for capturing overall accuracy and penalizing models that occasionally produce very poor predictions. However, the main limitation is its high sensitivity to outliers: in radar datasets, occasional imperfect frames or inaccurate GT may cause disproportionately large RMSE values, which can bias comparisons across methods.
- **Saturated Root Mean Square Error (S-RMSE)** is a truncated version of RMSE that reduces sensitivity to outliers. It is defined as:

$$S\_RMSE(\mathbf{X}_{car}, \hat{\mathbf{X}}_{car}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} d_t^2} \tag{2.37}$$

where:

$$d_t = \begin{cases} x^t - \hat{x}^t, & |x^t - \hat{x}^t| \leq c_{err} \\ \text{sign}(x^t - \hat{x}^t) \cdot c_{err}, & |x^t - \hat{x}^t| > c_{err} \end{cases} \tag{2.38}$$

where x^t and $\hat{x}^t$ denote the GT and estimated ego-motion component (either translational velocity $\hat{v}_x^{car}$ or yaw rate $\hat{\omega}$) at timestamp t. The sequences are denoted as $\mathbf{X}_{car} = \{x^1, x^2, \ldots, x^T\}$ for the GT and $\hat{\mathbf{X}}_{car} = \{\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^T\}$ for the estimates, where T is the total number of timestamps. The parameter c_{err} specifies the maximum tolerated error magnitude. In this thesis, c_{err} is set to 50 cm/s for translational velocity $\hat{v}_x^{car}$ and 2.86 deg/s for yaw rate $\hat{\omega}$, respectively, based on an empirical analysis of the selected RadarScenes sequences.

- **Median Absolute Error (MedAE)** is the median of absolute errors. Because it relies on the median operator, it is robust to extreme outliers and reflects the central tendency of the error distribution. Therefore, MedAE can indicate the "typical" error magnitude without being overly influenced by rare, larger deviations.
- **Mean Absolute Error (MAE)** is the mean of absolute errors. It directly represents the average magnitude of the estimation error and retains the same units as the underlying variable. Compared to RMSE, MAE is less sensitive to outliers because all errors are equally weighted, making it an intuitive and interpretable metric for overall accuracy.
- **Relative Trajectory Error (RTE)** evaluates the drift of the estimated trajectory with respect to the GT over fixed travel distances. For a given segment length L, the estimated and GT trajectories are partitioned into consecutive segments of length L (e.g., $L = 50$ m). For each segment, the starting points are aligned in the 2D plane, and the Euclidean distance between the estimated and GT end points is computed. The RTE for that segment length, denoted as RTE_L, is then obtained by averaging the endpoint deviations across all segments. In summary, RTE can reflect the long-term stability of an estimator, as accumulated errors along the trajectory are explicitly captured.

RTE is reported in meters, and all other four metrics (RMSE, S-RMSE, MedAE, and MAE) retain the same units as the underlying variable, namely centimeters per second (cm/s) for translational velocity and degrees per second (deg/s) for yaw rate. The metrics described above are used in subsequent chapters to evaluate vehicle ego-motion estimation and trajectory accuracy.

In addition to ego-motion, Chapter 5 investigates moving object segmentation, which requires a different evaluation scheme. Unlike ego-motion estimation, where errors are computed directly on continuous motion variables, this dissertation proposes a novel evaluation approach for radar-based moving object segmentation. The motivation arises from the limitations of point-level annotation in radar datasets: even in manually annotated data such as RadarScenes, points belonging to the same physical object are not always consistently labeled, and some may even be marked as background. Relying on point-to-point comparisons would therefore introduce spurious errors. Moreover, in downstream perception tasks such as object tracking or classification, exact point-level correspondence is less critical than the reliable detection of entire moving objects. For this reason, the evaluation is conducted at the object level, using associations between predicted and GT moving instances.

To this end, the following procedure is applied before calculating the metrics. Inspired by the convention in radar-based object tracking [24], the moving objects predicted by the proposed method and those identified in the GT labels are first clustered into moving instances. The density-based spatial clustering of applications with noise (DBSCAN) algorithm [25] is used for clustering. The grouped moving objects are then converted into point targets by averaging the positions of all points within each cluster. Afterwards, a cost matrix is calculated based on the Euclidean distance between objects in the GT and prediction lists. The Jonker–Volgenant algorithm [26] is subsequently applied to solve the data association problem. Based on its output, three quantities are determined for each radar frame: the number of correctly detected moving objects (TP), the number of false detections (FP), and the number of missed detections (FN). Finally, the TP, FP, and FN counts from all radar frames are accumulated, and the following evaluation metrics are calculated:

1. **False Discovery Rate (FDR)** shows the proportion of false detections among all detected moving instances. In other words, it reflects the frequency of false detections. FDR is computed as:

$$FDR = \frac{FP}{FP + TP} \tag{2.39}$$

2. **Missed Detection Rate (MDR)** measures how often true moving instances are misclassified as non-moving. It is computed as:

$$MDR = \frac{FN}{FN + TP} \tag{2.40}$$

3. **F1 Score (F1)** is the harmonic mean of Precision and Recall. Therefore, the F1 Score will be high only when both Precision and Recall are high. This property makes it well-suited for summarizing detection performance, especially in the case of class imbalance. It is computed as:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2.41}$$

4. **Intersection over Union (IoU)** is a commonly used evaluation metric for computer vision tasks such as detection and segmentation. Traditionally, it is computed geometrically based on the overlap between the predicted region (for example, the bounding box) and the actual region. However, due to the characteristics of radar sensors, the shape of detected objects changes with distance and angle, and they have fewer geometric features due to low azimuth resolution. In addition, the actual area may also be erroneous and incomplete due to errors in the GT label. Therefore, in order to adapt to the radar characteristics, this work defines the IoU metric as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.42}$$

Here, TP represents the correct overlap, FP represents the extra predicted moving instances, and FN represents the missed GT instances.

Together, the metrics described in this section establish a comprehensive evaluation framework for the dissertation. For ego-motion estimation tasks, RMSE, S-RMSE, MedAE, and MAE quantify frame-wise accuracy and robustness, while RTE captures long-term trajectory stability. For moving object segmentation, the proposed object-level evaluation scheme, combined with FDR, MDR, F1, and IoU, quantifies false alarms, missed detections, and the balance between precision and recall, while also assessing the overall overlap quality between predicted and GT moving instances. In summary, the unified evaluation framework ensures consistency across chapters and enables rigorous comparison of the developed methods against existing approaches.

3

DeepEgo: Radar-Based Vehicle Ego-Motion Estimation

The original plan of this PhD project was to investigate radar-based simultaneous localization and mapping (SLAM) with the aid of deep learning techniques. However, a detailed literature study revealed that successful radar SLAM depends on many intermediate steps, making a comprehensive study beyond the feasible scope and timeframe of this PhD. The focus was therefore narrowed to radar-based vehicle ego-motion estimation, which emerged as the most promising starting point. At the time, traditional model-based methods had been extensively studied, but their performance in real driving conditions remained limited. Deep learning methods were also being explored, yet they often treated the problem as a black box, forcing networks to directly regress ego-motion from radar data. Building on these observations, I set out to develop a method that combines the interpretability of model-based approaches with the flexibility of deep learning. This effort led to the creation of DeepEgo, the hybrid framework introduced in this chapter.

Parts of this chapter have been published in:

S. Zhu, F. Fioranelli, and A. Yarovoy, "Radar-only Instantaneous Ego-motion Estimation Using Neural Networks," 2023 20th European Radar Conference (EuRAD), Berlin, Germany, 2023, pp. 201-204.

S. Zhu, A. Yarovoy, and F. Fioranelli, "DeepEgo: Deep Instantaneous Ego-Motion Estimation Using Automotive Radar," in IEEE Transactions on Radar Systems, vol. 1, pp. 166-180, 2023.

S. Zhu, A. Yarovoy, F. Fioranelli, S. Ravindran, "An apparatus for determining ego-motion," in US Patent Application, patent filed in 2023, WO2024183926A1.

S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "DeepEgo+: Unsynchronized Radar Sensor Fusion for Robust Vehicle Ego-Motion Estimation," in IEEE Transactions on Radar Systems, vol. 3, pp. 483-497, 2025.

3.1. Introduction

Over the past few decades, advanced driving systems (ADSs) have attracted considerable attention in both industry and academia. Modern self-driving cars employ ADSs to assist, or in some cases replace, the human driver in handling complex driving situations [1, 2]. ADS technology is widely regarded as a means to reduce energy consumption and, more importantly, to improve traffic safety by mitigating accidents caused by human error [3]. At the system level, an ADS can be decomposed into a perception frontend and a decision-making backend. The backend is responsible for tasks such as path planning and motion control, while the frontend handles driving scene understanding [4]. Among the many signal-processing steps in the perception frontend, ego-motion estimation (i.e., self-localization) is one of the first components that directly processes raw sensor data [27]. A robust and accurate ego-motion estimator is therefore crucial, as errors at this stage can propagate and degrade the performance of downstream modules such as multi-object tracking and environment mapping. Traditionally, the motion of the ego-vehicle is estimated using odometry sensors such as inertial measurement units (IMUs), wheel encoders, and the global positioning system (GPS). However, each of these modalities has well-known shortcomings: IMUs suffer from drift over time [28], wheel encoders are vulnerable to wheel slippage under acceleration and braking [29], and GPS is prone to non-line-of-sight (NLOS) propagation errors and multipath reflections [30]. As a result, none of these sensors alone is sufficient for reliable ego-motion estimation under all driving conditions.

To address the limitations of conventional odometry sensors, ego-motion estimation has also been explored using alternative sensing technologies such as cameras [31], LiDAR (Light Detection and Ranging) [32], synthetic aperture sonar (SAS) [33], scanning radar [34], and automotive radar[1] [35]. Among these, automotive radar offers several unique advantages. Unlike cameras, radar operates reliably under all weather and illumination conditions [36]. Compared with LiDAR, it is less susceptible to line-of-sight (LoS) occlusions [37]. Furthermore, automotive radars are lightweight, low-cost, compact owing to advances in millimeter-wave (mmWave) technology [38], and can be mounted discreetly behind the bumper of a vehicle [39]. Finally, automotive radar can directly measure the radial velocity of detected objects [18], a feature that has proven highly useful for applications such as moving-object tracking [40] and instantaneous ego-motion estimation [13].

Despite these advantages, ego-motion estimation using automotive radar remains challenging. Automotive radar provides fewer geometric features about detected objects due to its relatively low resolution in range and angle-of-arrival (AoA) [41]. In addition, radar data are prone to false positives, missed detections, multipath reflections, and mutual interference [16]. These factors limit the direct transfer of many well-established ego-motion estimation methods developed for optical sensors to the radar domain. To address this gap, the objective of this chapter is to develop a robust and accurate ego-motion estimation algorithm tailored specifically for automotive radar. In particular, we propose an end-to-end solution for instantaneous ego-motion estimation using neural networks. Unlike conventional scan-matching approaches, the proposed method requires only a single radar frame, thereby avoiding the difficult problem of data association. Moreover, instead of directly

[1] In this dissertation, the term *scanning radar* refers to mechanical-scanning FMCW radar systems that generate high-resolution 360° scans. In contrast, the term *automotive radar* refers to fixed-beam FMCW MIMO radar sensors commonly used in vehicles.

regressing ego-motion from raw inputs [42], the method adopts a hybrid design: a neural network predicts point-wise weights for the radar point cloud, and these weights are then used in a weighted least squares (w-LSQ) backend to compute the final motion estimate.

In addition to these advantages, the proposed method employs neural networks to directly process multi-dimensional radar point clouds, with only minimal preprocessing such as normalization and resampling. The experimental results further demonstrate that incorporating additional object-level features, such as range and return power, improves translational and rotational velocity estimation by **64.3%** and **20.9%**, respectively. Compared with our preliminary study in [43], the method presented here differs substantially in both network structure and training strategy, leading to significant performance improvements. These differences can be summarized as follows. Firstly, in addition to the conventional motion loss, which penalizes the difference between estimated and ground-truth motion, this work introduces a novel Doppler loss. This loss function enables the network to automatically identify detections from static objects (inliers) in radar point clouds without manual annotation, a capability that is critical for robust w-LSQ. Moreover, the Doppler loss guides the network to predict the likelihood that a detection originates from a static object. These weights are not only essential for ego-motion estimation but also hold potential for downstream applications such as map denoising and static object detection, though these lie beyond the scope of this chapter.

Secondly, inspired by [44], a new network architecture is proposed that estimates and applies point-wise offsets to the w-LSQ method. The purpose of these offsets is to bring distant outliers closer to the regression line, thereby mitigating their adverse impact. Prior work has shown that point-wise offsets can stabilize network training in surface normal estimation tasks. In this study, they are shown to improve both training stability and ego-motion estimation accuracy, as confirmed by the ablation studies. Thirdly, the proposed method is comprehensively evaluated and compared with six state-of-the-art approaches using the challenging real-world radar dataset *RadarScenes* [15]. The evaluation covers 64 radar sequences, corresponding to over **79 km** of driving. Results show that the proposed method achieves more than a **50%** improvement in estimation accuracy relative to the second-best baseline, along with consistent gains across all evaluation metrics. In addition, the method is straightforward and requires no iterative optimization or random sampling. Quantitatively, it runs approximately **129.6** times faster than the Normal Distribution Transform (NDT)-based baseline.

The remainder of this chapter is organized as follows. Section 3.2 reviews related research on radar-based ego-motion estimation. Section 3.3 presents the design details of the proposed method. Section 3.4 reports a comprehensive experimental evaluation and compares the proposed method with existing approaches on a challenging real-world radar dataset. Finally, Section 3.5 summarizes the key findings and outlines directions for future work.

3.2. Related Work

This section provides an overview of the prior work in the area of (A) vehicle ego-motion with automotive radar, and (B) deep learning on point cloud processing from which the proposed method of this paper is inspired.

3.2.1. Vehicle Ego-Motion Estimation with Automotive Radar

In general, radar-based vehicle ego-motion estimation methods can be grouped into two main categories: scan-matching methods [12, 45–51] and instantaneous methods [13, 52–57]. The fundamental difference between these two approaches lies in the nature of the radar data they exploit for motion estimation.

Scan-matching Methods The fundamental idea behind scan-matching is to estimate the relative Euclidean transformation between consecutive radar point clouds [45]. This principle is similar to approaches widely used in camera- and LiDAR-based localization [58]. While scan-matching methods have shown promising performance on high-resolution scanning radars [59], their direct application to automotive radars is challenging due to low angular resolution and radar cross-section (RCS) fluctuations. To address these issues, several adaptations have been proposed. Hard point-to-point associations have been replaced by soft associations using the Normal Distributions Transform (NDT) [60], which represents each radar point or cluster as a Gaussian distribution [45, 46]. Additional radar features such as measured radial velocity and return power have been incorporated to improve estimation accuracy and mitigate the effect of false positives [12, 47]. More recently, complex noise models have been integrated into the optimization objective [49], in order to reduce local maxima caused by summation approximations [47]. Despite these efforts, scan-matching methods for automotive radar still face several limitations. First, they are inherently optimization-based, requiring good initialization and iterative refinement. Second, many approaches rely on heuristic design choices and manually tuned parameters, which may not adapt well to diverse driving scenarios. Finally, scan-matching can suffer from so-called "tunneling effects" when the radar is not forward-facing. In such cases, large and dense point clouds from nearby static structures (e.g., curbs or vegetation) dominate the scans but provide little motion information.

Instantaneous Methods Instantaneous methods exploit the fact that, for a given azimuth, the measured radial velocity of a static object can be expressed as a linear transformation of the vehicle's own motion. The main objective is therefore to identify detections originating from static objects (inliers). Once these inliers are located, vehicle ego-motion can be estimated using linear regression. Compared with scan-matching methods, instantaneous methods require only a single radar frame (hence the term "instantaneous"), avoid the need for data association in sparse point clouds, and are less affected by RCS fluctuations. However, real-world radar data typically contain a large number of outliers (moving objects and false positives), making the reliable identification of inliers a non-trivial challenge. To address this, the Random Sample Consensus (RANSAC) algorithm [61] has been widely adopted to reject outliers prior to regression [13, 52]. Powered by RANSAC, instantaneous methods have been the standard for radar-based ego-motion estimation in various applications for nearly a decade [48, 55, 62, 63]. However, despite their success, RANSAC-based instantaneous methods face several limitations. First, they rely exclusively on azimuth and radial velocity measurements to identify outliers, while prior work [12, 57] has shown that additional features such as range and return power can provide valuable cues. Second, they involve iterative procedures and hyperparameters that must be tuned to the current outlier ratio, complicating real-time deployment. Third, their robustness depends on the majority assumption: static objects must outnumber moving objects and false detections, otherwise performance degrades significantly. Finally, inliers are determined by binary thresholding,

which makes these methods vulnerable to slow-moving objects that may be misclassified as static.

3.2.2. Deep Learning on Point Cloud Processing

Driven by breakthroughs in deep learning (DL) techniques, a large body of recent research has focused on applying DL frameworks to vehicle perception [64]. However, most studies have concentrated on data from LiDAR [65–67] and scanning radar [68–70], while automotive radar has received comparatively little attention. As noted earlier, this is largely because automotive radar data are typically sparser, noisier, and more expensive to collect than those of other modalities, making it difficult to train neural networks (NNs) effectively. Despite these challenges, several pioneering works have demonstrated the feasibility of ego-motion estimation using DL tools. Nevertheless, these approaches either rely on auxiliary odometry sensors to improve performance [42] or are constrained to idealized scenarios with an almost static environment [71]. Furthermore, most existing methods employ image-based feature extraction backbones, such as convolutional neural networks (CNNs), to process radar data [72]. However, modern commercial automotive radars typically output multi-dimensional radar point clouds [15], which CNNs cannot be directly applied to. Therefore, before developing the proposed methodology, it is necessary to review the available feature extraction backbones and their respective advantages and limitations when applied to point clouds. These can be broadly grouped into the following categories:

Image-based methods project multi-dimensional point clouds into two-dimensional (2D) image-like representations, after which conventional 2D CNN backbones can be applied. This approach has the advantage of leveraging mature 2D CNN architectures and training pipelines, while also offering relatively low computational and memory costs. In addition, the projection step converts irregular point clouds into dense image-like grids, which are well-suited to hardware optimized for 2D convolution. However, image-based methods inevitably discard geometric information by collapsing the multi-dimensional structure into a 2D plane. Depending on the chosen projection, different types of information may be lost (e.g., elevation in range–azimuth projection, range in elevation–azimuth projection), and multiple points may collapse into the same pixel [73, 74]. As a result, fine-grained geometric details cannot be preserved, and performance is sensitive to the chosen projection plane, and projection may also lead to occlusions. These limitations restrict the ability of 2D projection methods to generalize across different perception tasks.

Voxel-based methods [75] discretize point clouds into 3D volumetric cells, or voxels, which are then processed using 3D CNNs. Compared with 2D projections, voxelization preserves more geometric structure and allows networks to exploit spatial reasoning across all three dimensions. This makes voxel-based methods attractive for capturing local context and modeling object geometry in greater detail. On the other hand, voxel-based methods are computationally and memory intensive [76], since 3D convolutions scale poorly with resolution. To keep the computation tractable, relatively coarse voxel grids are often used, which introduces quantization errors and may result in the loss of fine-grained features. These trade-offs can degrade ego-motion estimation performance and limit the practical deployment of voxel-based methods in real-time automotive systems.

Point-based methods, such as PointNet [77] and PointNet++ [78], directly operate on raw, unordered point clouds without requiring projection or voxelization. This design pre-

serves the full geometric fidelity of the data and naturally provides permutation invariance to the input point order. Furthermore, hierarchical point-based networks [78, 79] can capture both global and local structures, enabling fine-grained feature learning that is particularly well suited for sparse radar data. Despite these advantages, point-based methods also face challenges. They are computationally more expensive per point than image-based methods, and scaling them to very large point clouds often requires aggressive sampling strategies. Moreover, compared to CNN-based approaches, the training pipelines for point-based networks are less mature and may require careful engineering to achieve robust performance.

In summary, image-based, voxel-based, and point-based methods each offer complementary strengths and weaknesses when applied to point clouds. Image-based methods are computationally efficient but discard geometric detail; voxel-based methods preserve more structure but incur high memory costs and quantization errors; and point-based methods maintain data fidelity but are computationally intensive and less mature in terms of standardized training practices. These considerations motivate the design of the proposed method in this chapter. Specifically, to achieve accurate ego-motion estimation while maintaining computational efficiency, the proposed framework adopts a point-based backbone coupled with a hybrid architecture that reduces complexity by incorporating model-based priors. The following section presents a detailed description of the proposed method.

3.3. Proposed Method

This section details the design of the proposed method for radar-based ego-motion estimation. The remainder of the section is organized as follows. Section 3.3.1 formally defines the ego-motion estimation problem. Section 3.3.2 presents the proposed architecture in detail. Section 3.3.3 introduces the novel loss function used to guide network training. Finally, Section 3.3.4 provides the implementation details.

3.3.1. Problem Formulation

The proposed method addresses the problem of estimating the instantaneous 2D ego-motion of a vehicle using data from a linear-array automotive radar within a single measurement cycle. Specifically, after the implementation of detection algorithms, the radar provides a multi-dimensional point cloud with J detection points, each associated with M features. For example, each detection may include features such as range, AoA, radial velocity, and return power, depending on the radar hardware and signal-processing pipeline. For the ego-vehicle coordinate system, the origin is placed at the rear-center of the vehicle. The x-axis aligns with the vehicle's longitudinal (down-range) direction, and the y-axis aligns with the lateral (cross-range) direction. Under this convention, the vehicle's 2D motion state is denoted as $\left[v_x^{car}, v_y^{car}, \omega\right]$, where v_x^{car} is the down-range velocity, v_y^{car} is the cross-range velocity, and ω is the yaw rate (rotational velocity).

Consider an automotive radar mounted at position $\{x, y, \theta\}$, with x and y denoting its displacement relative to the vehicle coordinate system, and θ the mounting angle relative to the vehicle x-axis. In the radar's local coordinate system, the x-axis is aligned with the radar boresight, while the y-axis is perpendicular to it. The 2D motion state of the radar is expressed as $[v_x^{rad}, v_y^{rad}, \omega]$, where v_x^{rad} and v_y^{rad} are the down-range and cross-range velocities, respectively, and ω is the yaw rate. Since all points on a rigid body share the

same angular velocity, ω is common to both the radar and the vehicle. For instantaneous ego-motion estimation, the radar must provide at least the radial velocity[2] d_j (m/s) and AoA α_j (radians) of each detection j ($M \geq 2$). For detections originating from static objects (e.g., buildings), the measured radial velocity equals the projection of the radar ego-motion onto the detection's line-of-sight (LoS) direction, leading to the following equality:

$$\begin{bmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_J \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ \cos(\alpha_2) & \sin(\alpha_2) \\ \vdots & \vdots \\ \cos(\alpha_J) & \sin(\alpha_J) \end{bmatrix} \begin{bmatrix} v_x^{rad} \\ v_y^{rad} \end{bmatrix} \tag{3.1}$$

For compactness, let $\mathbf{D}$ denote the stacked vector of $-d_j$, $\mathbf{A}$ the projection matrix constructed from the AoA values, and $\mathbf{V}$ the vector of unknown radar velocities (excluding ω). With these definitions, Equation 3.1 can be expressed in matrix form as:

$$\mathbf{D} = \mathbf{A} \cdot \mathbf{V} \tag{3.2}$$

Given at least two independent detection points ($J \geq 2$ and $\mathbf{A}$ has full column rank), $\mathbf{V}$ can be estimated using ordinary least squares:

$$\mathbf{V}^{est} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{D} \tag{3.3}$$

Since the radar is rigidly mounted on the vehicle, the following coordinate transformation relates the radar motion state to that of the vehicle:

$$\begin{bmatrix} v_x^{rad} \\ v_y^{rad} \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x^{car} \\ v_y^{car} \\ \omega \end{bmatrix} \tag{3.4}$$

The above equation transforms the vehicle's velocity state into the radar's coordinate frame, where the leftmost matrix represents a rotation by angle θ, and the middle matrix accounts for the displacement of the radar from the vehicle origin. However, estimating only v_x^{rad} and v_y^{rad} is insufficient to fully recover the vehicle motion state, which contains three unknowns, unless the angular velocity ω is provided by an IMU or multiple synchronized radars are available [52]. To resolve this issue, this chapter assumes that the ego-vehicle does not experience significant lateral motion, i.e., $v_y^{car} = 0$. It is worth noting that this assumption is reasonable in automotive applications and has been widely adopted in prior works [12, 13, 15, 47, 48]. Moreover, datasets that include measurements of the ego-vehicle's lateral speed (e.g., [80]) show that it is typically very small and remains zero for most of the time. With this zero-lateral velocity assumption, the relationship between the radar and vehicle motion states simplifies to:

$$\begin{aligned} \omega &= \frac{v_y^{rad}\cos(\theta) + v_x^{rad}\sin(\theta)}{x} \\ v_x^{car} &= v_x^{rad}\cos(\theta) - v_y^{rad}\sin(\theta) + \omega \cdot y \end{aligned} \tag{3.5}$$

[2]Negative towards radar

This formulation enables recovery of the vehicle's forward velocity v_x^{car} and yaw rate ω directly from radar measurements under the zero-lateral motion assumption. The remaining challenge is to accurately identify detections originating from static objects ("inliers"). Real-world driving scenes inevitably contain many non-static objects ("outliers"), such as moving vehicles, which do not satisfy Equation (3.1). These outliers can severely bias the solution of Equation (3.3), since ordinary least squares is highly sensitive to outliers. As discussed in Section 3.2, traditional instantaneous methods mitigate this issue using random sampling strategies such as RANSAC [13] and MSAC [81]. While these approaches can yield good results under favorable conditions, they suffer from several drawbacks. First, they are iterative in nature, which impacts runtime performance. Second, they rely only on AoA and radial velocity measurements, leaving richer object features provided by automotive radars (e.g., range, return power) unused. Third, their fixed parameters require adaptive tuning to handle varying conditions. Finally, and most importantly, their success depends on the majority assumption: inliers must outnumber outliers, otherwise performance can degrade dramatically. To overcome these limitations, this chapter introduces *DeepEgo*, a hybrid framework that leverages neural networks to exploit the full set of radar features and robustly distinguish inliers from outliers, while retaining interpretability through a model-based backend. The design details of *DeepEgo* are presented in the next section.

3.3.2. Network Architecture

The overall architecture of the proposed ego-motion estimator, *DeepEgo*, is illustrated in Figure 3.1. The design follows a hybrid paradigm: a neural network frontend extracts spatial features from radar point clouds and predicts point-wise weights and offsets, while a weighted least squares (w-LSQ) backend computes the final ego-motion estimate. This combination leverages the representation power of neural networks while retaining interpretability through a model-based estimation step. The input to *DeepEgo* is a radar point cloud, consisting of J detections with M associated features. *DeepEgo* is not tied to a specific radar specification. It only requires that the radar provides at least radial velocity and AoA measurements (i.e., $M \geq 2$). The primary challenge is to identify inlier detections, since these are essential for robust ego-motion estimation. To address this, the first stage employs a shared multilayer perceptron (shared-MLP) [77] that encodes the raw input features into a high-dimensional feature space. The shared-MLP applies linear and nonlinear transformations independently to each detection point, thereby producing point-wise features. Importantly, this operation is permutation invariant with respect to the order of input points, a desirable property since radar point clouds are inherently unordered.

Following the feature encoding stage, the output of the shared-MLP is passed through an average pooling layer to perform global feature extraction. The pooling operation, being a symmetric function, aggregates information across all points to produce a compact global feature vector that characterizes the entire radar point cloud. To enrich the representation, this global feature vector is then duplicated J times and concatenated with both the original input features and the intermediate point-wise features from the encoder. As shown in Figure 3.1, this results in a $J \times F$ feature matrix that jointly encodes local and global information. The combination of local and global features is particularly important, as it enables the network to reason about individual detections in the context of the overall scene, thereby improving the reliability of inlier identification. This feature matrix is subsequently processed

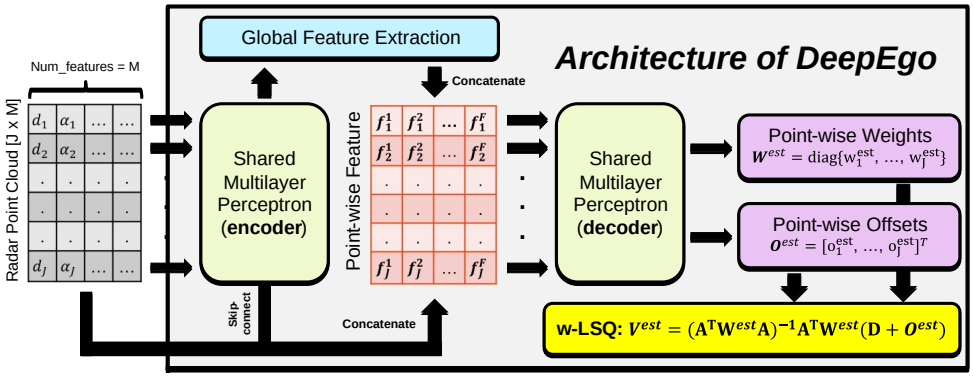


Figure 3.1: The architecture of the proposed method (*DeepEgo*). The input data is the radar point cloud with J points and M features. Note that $M \geq 2$ is required in this chapter because radial velocity and AoA measurements are essential object features for instantaneous ego-motion estimation. The outputs of *DeepEgo* are the motion predictions $\mathbf{V}^{est}$ and pointwise weights $\mathbf{W}^{est}$.

by a second shared-MLP, which decodes the fused representation into a low-dimensional embedding. Finally, the decoder output is passed to two prediction heads: one dedicated to estimating point-wise weights, and the other to predicting point-wise offsets. Together, these outputs provide the necessary inputs for the weighted least squares backend.

Since *DeepEgo* employs w-LSQ for ego-motion estimation, the point-wise weight prediction forms the most critical output of the network, as it directly attenuates the influence of outliers by downscaling their contribution to the regression. This prediction is realized through a fully connected layer with a single output neuron per point, followed by a sigmoid activation to constrain the predicted weights to the interval $[0, 1]$. In addition to the weights, *DeepEgo* introduces point-wise offset prediction, inspired by [44]. Before the w-LSQ stage, the measured radial velocities are adjusted by the learned offsets, thereby reducing the impact of "distant" outliers, detections whose radial velocities deviate significantly from the expected values. This mechanism not only mitigates the effect of hard-to-fit points but also improves the robustness of the system by reducing its sensitivity to their associated weights. Offset prediction is implemented using a single fully connected neuron per point, without an activation function to allow unrestricted adjustment. Together, the predicted weights and offsets provide refined inputs to the w-LSQ backend, enabling robust and accurate estimation of the radar ego-motion. Finally, the vehicle's translational and rotational velocities can be efficiently recovered according to Equation 3.5.

3.3.3. Loss Function

This section introduces the novel loss function designed for training *DeepEgo*, which integrates three key components: motion loss, Doppler loss, and sample weighting. The first component, the motion loss, employs the mean squared error (MSE) to compute the deviation between the estimated radar motion $\mathbf{V}^{est}$ and the corresponding ground truth $\mathbf{V}^{gt}$. Formally, it is defined as:

$$\ell_{\text{Motion}}^{(b)} = MSE(\mathbf{V}_{(b)}^{gt}, \mathbf{V}_{(b)}^{est}) = \frac{1}{2} \cdot [(v_{x,(b)}^{rad,gt} - v_{x,(b)}^{rad,est})^2 + (v_{y,(b)}^{rad,gt} - v_{y,(b)}^{rad,est})^2] \quad (3.6)$$

where b indexes the training samples within a mini-batch. Although conceptually simple, the motion loss plays a crucial role as it directly encodes the primary objective of this study: producing accurate ego-motion estimates. It ensures that the network is explicitly optimized to minimize the discrepancy between predictions and the true radar/vehicle motion, thereby anchoring the overall training objective.

While the motion loss encourages predictions to align with the ground truth, it does not explicitly guide the network in distinguishing between inliers and outliers, nor in assigning appropriate weights during the w-LSQ process. As shown later in Section 3.4.5, relying solely on motion loss causes the model to overfit a small number of points while neglecting many valid inliers. Consequently, the w-LSQ step becomes overly dependent on the correct identification of a few keypoints and degrades significantly when outliers are mistakenly classified as inliers. To mitigate this issue, a novel Doppler loss is introduced. The central idea is to leverage the ground-truth ego-motion to compute the discrepancy between the expected and the measured radial velocity, and then use this discrepancy to guide the learning of the pointwise weights $\mathbf{W}^{est}$. Specifically, based on Equation (3.2), the expected radial velocity measurement $\mathbf{D}^{exp}$ can be expressed as:

$$\mathbf{D}^{exp} = \mathbf{A} \cdot \mathbf{V}^{gt} \quad (3.7)$$

where $\mathbf{A}$ is the projection matrix constructed from the AoA, and $\mathbf{V}^{gt}$ is the ground-truth ego-motion vector. The Doppler error $\mathbf{D}^{err}$, defined as the difference between expected and measured radial velocities, is given by:

$$\mathbf{D}^{err} = \mathbf{D}^{exp} - \mathbf{D} \quad (3.8)$$

In an ideal case, $\mathbf{D}^{err}$ would be a zero vector if all J points originated from stationary objects on the ground plane and both measurements and ground-truth motion were noise-free. In practice, however, radar data are contaminated by outliers (e.g., moving vehicles, false detections) and measurement noise. To distinguish inliers from outliers, it is common to model the Doppler error as a Gaussian distribution with zero mean [47]. Accordingly, the pointwise weight $\mathbf{W}^{gt}$ can be expressed as:

$$\mathbf{W}^{gt} = \exp\left(-\frac{(\mathbf{D}^{err})^2}{2\sigma^2}\right) \quad (3.9)$$

where σ denotes the standard deviation of the assumed Gaussian distribution. In this work, σ is treated as a tunable empirical parameter, although it may also be approximated from radar system characteristics such as angular and Doppler noise variances [45, 47]. Finally, given the estimated weights $\mathbf{W}^{est}$ and the Gaussian-derived ground-truth weights $\mathbf{W}^{gt}$, the Doppler loss is defined as follows:

$$\ell_{\text{Doppler}}^{(b)} = MSE(\mathbf{W}_{(b)}^{gt}, \mathbf{W}_{(b)}^{est}) = \frac{1}{J} \sum_{j=1}^{J} (w_{j,(b)}^{gt} - w_{j,(b)}^{est})^2 \quad (3.10)$$

To further enhance robustness against poor training examples, a sample weighting mechanism is introduced. The sample weight s is defined for each training example as the sum of the pointwise weights, i.e.,

$$s^{(b)} = \sum_{j=1}^{J} w_{j,(b)}^{gt} \tag{3.11}$$

Intuitively, s reflects the amount of reliable information present in a radar point cloud. Samples with more detections consistent with the expected radial velocity receive higher weights, while samples dominated by outliers, inaccurate ground truth, or non-ideal conditions (e.g., non-zero lateral velocity or elevated objects) are down-weighted. This mechanism significantly reduces the negative influence of unreliable training data. Finally, bringing all components together, the overall training loss for *DeepEgo* is formulated as:

$$\mathcal{L}_{all} = \frac{\sum_{b=1}^{B} s^{(b)} \ell_{\text{Motion}}^{(b)}}{\sum_{b=1}^{B} s^{(b)}} + \mu \cdot \frac{\sum_{b=1}^{B} s^{(b)} \ell_{\text{Doppler}}^{(b)}}{\sum_{b=1}^{B} s^{(b)}} \tag{3.12}$$

where B is the batch size, and μ is a weighting factor that balances the contribution of the Doppler loss relative to the motion loss and is determined empirically.

3.3.4. Implementation Details

To reproduce the proposed method, the following implementation details are provided:

1. **Shared-MLP**: Two shared multilayer perceptrons (MLPs) are used for point cloud feature encoding and decoding. Each shared-MLP consists of three fully connected layers with hidden sizes of (128, 256, 512) for the encoder, and (512, 256, 128) for the decoder. Each fully connected layer is followed by a Batch Normalization (BN) [82] layer and a non-linear activation function. Unless otherwise specified, the rectified linear unit (ReLU) [83] is used as the default activation.

2. **Data Resampling**: To ensure a fixed input size, each radar point cloud is randomly up- or down-sampled to 256 points. In addition, point clouds with fewer than 30 detections are discarded prior to resampling, as they are considered too sparse for meaningful learning.

3. **Data Normalization**: In this work, radial velocity and AoA are used directly in their physical units (m/s and radians, respectively), without normalization[3]. Other auxiliary features, such as range and return power, are rescaled using min-max normalization to ensure stable training.

4. **Network Training**: The network is trained using mini-batch gradient descent with a batch size of $B = 512$. The learning rate is initialized at 1×10^{-3} and optimized using

[3] This is because preserving the physical meaning of AoA and radial velocity is important for computing the loss function and the w-LSQ regression. While input normalization is generally recommended for neural network training, these features are physically bounded, and training stability is further ensured through the use of BN layers, thereby reducing the associated risk.

root mean square propagation (RMSProp). For validation, 20% of the training set is held out, and early stopping is applied when the validation loss does not improve for 50 consecutive epochs. Finally, the proposed network is designed for the general case of a moving vehicle equipped with a single radar. If multiple radars are available, a separate instance of the network needs to be trained for each sensor.

3.4. Results and Discussion

This section presents the evaluation results of the proposed ego-motion estimation method, *DeepEgo*. Section 3.4.1 introduces the baseline methods selected for comparison, followed by a performance comparison in Section 3.4.3. Section 3.4.4 reports results of testing *DeepEgo* on the complete RadarScenes dataset, and Section 3.4.5 presents an ablation study to analyze the contributions of individual design choices. The radar dataset and evaluation metrics used for performance measurement were previously introduced in Section 2.2.

3.4.1. Selected Methods for Comparison

For performance comparison, the following state-of-the-art (SOTA) methods, spanning a wide range of methodologies, are selected from the literature:

1. **Biber's Method [60]**: The first NDT-based approach, which models measurement uncertainty using local normal distributions without requiring explicit scan-to-scan associations. It is an iterative method, where the piecewise continuous and differentiable score function can be optimized using Newton's method.

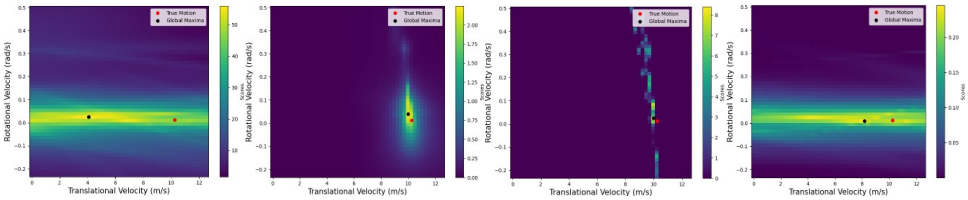
2. **Kellner's Method [13]**: An instantaneous method that estimates ego-motion from a single scan. The approach remains iterative due to its reliance on the RANSAC algorithm [61] for outlier rejection. Unlike NDT-based approaches, it formulates ego-motion as a parametric curve-fitting problem using AoA and radial velocity measurements.

3. **Rapp's Method [47]**: An improved clustering-based NDT method that integrates spatial registration and a likelihood model for radial velocity. It extends earlier works [45, 46] and outperforms them with a single radar, although its accuracy remains lower than Kellner's method.

4. **Lu's Method [42]**: The first deep learning–based method combining automotive radar with IMU. It uses a CNN-based subnet to process mmWave radar images and extract ego-motion features, outperforming conventional ICP-based methods [84, 85].

5. **Heller's Method [86]**: An extension of 2D-NDT to 3D polar coordinates, incorporating radial velocity and return power into the distribution modeling and score function. This method achieves higher accuracy than conventional NDT approaches.

6. **Kung's Method [12]**: A 2D-NDT pipeline applicable to both scanning and automotive radar. It constructs probabilistic radar submaps from multiple scans and applies returned power for thresholding. Reported performance surpasses Kellner's [52], Rapp's [47], and ICP-based methods [35, 81].



(a) Biber's Method (b) Rapp's Method (c) Heller's Method (d) Kung's Method

Figure 3.2: Results of the score functions of the four selected NDT-based methods. The scores are calculated using the exhaustive grid search in the translational velocity and rotational velocity space. In the above plots, the location of the ground-truth pose is represented by a red dot, and the global maximum of the score function is denoted by a black dot.

Among the six selected methods, there is one instantaneous approach based on AoA–velocity profile analysis (Kellner's), one scan-matching method leveraging deep neural networks (Lu's), and four scan-matching methods based on NDT (Biber's, Rapp's, Heller's, and Kung's). For Lu's method, the official implementation of the mmWave sub-network provided by the authors is used. The only modification relates to the input format: since RadarScenes point clouds do not contain elevation information, radar data are converted into 2D bird's-eye-view images rather than the 2D depth images used in the original work. For Kellner's method, no official implementation is publicly available. The method is therefore re-implemented based on the procedures described in the original paper. The remaining four NDT-based scan-matching methods are also re-implemented due to the lack of publicly available code, following the descriptions in the corresponding publications. For all re-implemented methods, hyperparameters are tuned using a small validation subset of the RadarScenes dataset. All baselines are evaluated using the same dataset selection and evaluation protocol as the proposed method to ensure a fair comparison.

Since the NDT-based methods share the same principle but differ mainly in their score function design, it is reasonable to first compare them and then select the most competitive one for further evaluation against the proposed method and other SOTA baselines. Figure 3.2 shows the score surfaces of the four NDT methods obtained via exhaustive grid search in the translational and rotational velocity space. The ground-truth pose is marked by a red dot, while the global maximum of the score function is denoted by a black dot. It can be observed that the global maximum in Kung's method (d) lies closer to the ground truth than in Biber's method (a). However, neither method can reliably resolve the translational velocity of the ego-vehicle. By contrast, both Rapp's (b) and Heller's (c) methods achieve more accurate estimates of translational and rotational velocities, owing to their explicit use of radial velocity measurements. Moreover, Rapp's method is more likely to converge to the global maximum under proper initialization, whereas Heller's method shows less stable behavior. Based on these observations, Rapp's method is selected as the representative NDT baseline for further comparison with the proposed approach. To the best of the author's knowledge, this is the first study to systematically compare all four NDT-based methods using the same dataset. While a detailed failure analysis is beyond the scope of this dissertation, the parameter settings of the compared methods are provided in [57], and readers are encouraged to consult the original papers for additional details.

Table 3.1: The five selected scenes used for performance evaluation with SOTA methods. Each scene has data sequences from four automotive radars. For deep learning-based methods, these data sequences are not used for model training and validation.

Index	Surface	Weather	Traffic	Length	Road Types
Sequence 89	Bumpy	Rainy	No	≈752m	Collector Road
Sequence 67	Smooth	Clear	Little	≈668m	Local Street
Sequence 108	Smooth	Clear	No	≈2281m	Arterial Road
Sequence 10	Smooth	Cloudy	High	≈1443m	Collector Road
Sequence 138	Smooth	Clear	Medium	≈700m	Collector Road

3.4.2. Dataset and Evaluation Protocol

All experiments in this chapter are conducted using the RadarScenes dataset [15]. To ensure meaningful ego-motion estimation, only scenes in which the ego-vehicle travels at least 500 m are selected, resulting in 64 scenes out of the original 158. Each scene contains four approximately two-minute-long radar recordings acquired independently by four on-vehicle automotive radars (Radar 1–4). The selected subset comprises approximately 10.5k radar frames from Radar 1, 11.2k from Radar 2, 11.2k from Radar 3, and 10.9k from Radar 4. Performance evaluation is conducted at the radar level, meaning that recordings from different radars are treated independently and results are reported separately.

Different sections of this chapter adopt slightly different evaluation scopes. Section 3.4.4 evaluates all 64 scenes using data from all four radars, whereas Section 3.4.3 evaluates five selected scenes using data from all four radars. All remaining experiments use five selected scenes from Radar 3 only. For learning-based methods, a leave-one-out (L1O) protocol is adopted at the sequence (recording) level. In each fold, one recording is held out for testing, while the remaining recordings are split into training and validation sets. This procedure is repeated until every recording has been used once as test data, and the final performance for each radar is obtained by averaging the results across all L1O folds.

3.4.3. Results of Comparison with SOTA Methods

Table 3.1 summarizes the five scenes selected for comparison with SOTA methods. These scenes were chosen to cover diverse conditions in terms of weather, traffic density, and road type. Each scene includes data sequences from the four automotive radars mounted on the test vehicle, yielding a total of 20 sequences (five scenes × four radars). For deep learning–based methods, these scenes are used exclusively for testing following a leave-one-out (L1O) protocol. Under this setup, the test scenes remain entirely unseen during training and validation, which is essential for evaluating the generalization capability of the models.

Table 3.2 presents the root mean square error (RMSE) of translational velocity estimation. In this case, Lu's method scores the worst; a potential reason is that Lu's method does not specifically mitigate the influence of dynamic objects in the scene. For Rapp's method, when the Radar 2 or Radar 3 is used, the error in translational velocity estimation is small. This is because these two radars are front-facing, hence, they are more sensitive to vehicle speed than the Radar 1 or Radar 4, which are side-looking. Clearly, Kellner's method has the lowest RMSE in translational velocity estimation among the three methods chosen for

Table 3.2: The root mean square error (RMSE) of translational velocity estimation. The unit is meters per second. For each radar, the reported result is averaged over the five test data sequences.

Methods	Radar 1	Radar 2	Radar 3	Radar 4	Mean
Kellner's	0.135	0.0904	0.145	0.390	0.190
Rapp's	1.268	0.474	0.692	1.296	0.933
Lu's	3.207	2.508	2.807	2.918	2.860
Proposed	**0.0955**	**0.0877**	**0.0876**	**0.203**	**0.118**

Table 3.3: The RMSE of rotational velocity estimation. The unit is degrees per second. For each radar, the reported result is averaged over the five test data sequences.

Methods	Radar 1	Radar 2	Radar 3	Radar 4	Mean
Kellner's	0.785	1.014	2.315	0.894	1.255
Rapp's	7.563	13.866	14.668	8.079	11.058
Lu's	5.031	5.546	4.973	4.641	5.048
Proposed	**0.642**	**0.859**	**0.911**	**0.653**	**0.768**

comparison. However, the proposed approach still outperforms Kellner's method, regardless of the position of the radar on the vehicle. On average, the proposed solution is about **37.9%** better than Kellner's method, while also maintaining consistently small estimation errors across all radars (mean RMSE of only 0.118 m/s). This demonstrates not only its relative advantage over existing methods but also its robustness and reliability in real-world driving conditions.

Table 3.3 shows the RMSE of rotational velocity estimation. First, it can be observed that all tested methods perform relatively poorly on the front-facing radars, since side-looking radars are more sensitive to the angular motion of the ego-vehicle. Nevertheless, the proposed method achieves the lowest RMSE compared to the SOTA methods. On average, the estimation accuracy is improved by **38.8%** compared to the second-best method. These results demonstrate the accuracy of the proposed method in both translational and rotational velocity estimation. However, the RMSE metric cannot quantitatively capture the effect of drift. Therefore, an additional evaluation metric, the relative trajectory error (RTE), is introduced. Figure 3.3 shows the RTE for the four tested methods. To better evaluate the long-term stability of the methods, three driving distances were selected for the RTE metric, namely 10 m, 20 m, and 30 m. Based on the figure, it is evident that the RTE is more affected by errors in rotational velocity estimation than in translational velocity estimation, although it should be seen as a combined effect. Still, the proposed method achieves the best performance among all tested methods, regardless of radar position or scene conditions. Finally, it is worth mentioning that although the proposed method outperforms Kellner's method on the RMSE metric by a large margin, Kellner's method remains competitive on the RTE metric.

For real-time applications such as autonomous driving, ego-motion estimation methods must demonstrate strong runtime performance. Figure 3.4 shows the update rates of the tested methods. Among them, Rapp's method is the slowest, achieving only five frames per second (FPS). Although 17 FPS was reported in [47] with a more powerful CPU, the

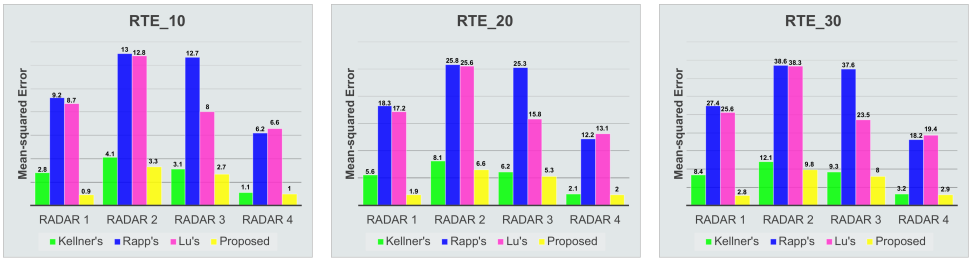


Figure 3.3: The relative trajectory error (RTE) of the methods under comparison. In this experiment, three driving distances (10 m, 20 m, and 30 m) were selected for the RTE metric. The unit is in meters. For each radar, the reported result is the average over the five test data sequences.

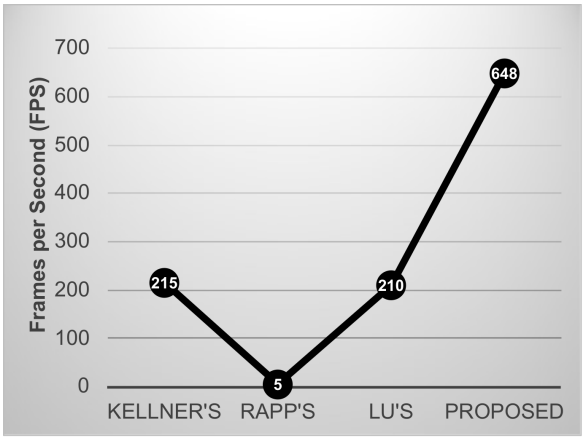
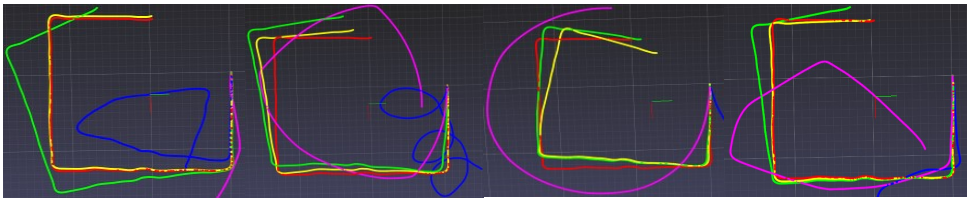


Figure 3.4: Update rate of the methods under test. All methods are tested under the same system environment using the Delft High Performance Computing Center (DHPC) *DelftBlue* [87]. In this experiment, neither GPU nor parallel computing is applied. Instead, a single Intel Xeon compute node is used.

relatively low speed is expected since the method requires an iterative optimization process for every measurement scan. In contrast, Kellner's method and Lu's method achieve similar runtime performance, with 215 FPS and 210 FPS, respectively. For Kellner's method, there is no optimization process, but multiple iterations are still needed due to the random sampling mechanism [61]. Lu's method performs direct, non-iterative estimation, but its large network size leads to significant computational cost. By comparison, the proposed method achieves the fastest runtime. It is entirely non-iterative and supported by a lightweight neural network with about 800K trainable parameters, in contrast to nearly 15M parameters in Lu's method. Finally, it should be noted that this experiment assumes pre-collected data; in real-world deployments, runtime performance may additionally be constrained by factors such as the radar update rate.

Finally, to visualize the long-term stability of the tested methods, Figure 3.5 shows the trajectories obtained by integrating the estimated ego-motion over time, aligned to the same global reference point. The ground-truth trajectory is also plotted for comparison. As shown in the figure, Lu's method deviates significantly from the true path, mainly due to



(a) Radar #1 (b) Radar #2 (c) Radar #3 (d) Radar #4

Figure 3.5: The estimated vehicle trajectory from Sequence 67. The trajectory is calculated by integrating the estimated ego-motion given the timestamps. In the plot, the red line represents the **ground-truth** trajectory, the yellow line represents our **proposed method**, the green line represents **Kellner's Method**, the pink line represents **Lu's Method**, and the blue line represents **Rapp's Method**.

the discretization of radar point clouds, which is unavoidable when using CNNs but destroys positional accuracy. Rapp's method performs slightly better, yet the drift remains pronounced, likely due to the "tunnel" effect of road curbs, which biases the spatial likelihood [47] toward zero motion. Kellner's method demonstrates better long-term stability than other SOTA methods, as its random sampling mechanism [61] helps suppress the influence of non-static objects. However, the success of random sampling techniques relies on the majority inlier assumption. Therefore, under adverse conditions, large errors may occur and accumulate when non-stationary detection points are included in the estimation. In contrast, the proposed method shows visibly less drift than all baselines, particularly when using side-looking radars, confirming its robustness in long-term trajectory estimation.

3.4.4. Further Results on All Data

The previous sections evaluated seven ego-motion estimation methods using five selected sequences collected by four automotive radars. The results demonstrated that the proposed method achieves the best estimation accuracy, long-term stability, and runtime performance. To provide a more comprehensive analysis, this section conducts further comparisons using all 64 radar sequences, corresponding to an equivalent driving distance of over 79 km. Since Kellner's method proved to be the most competitive baseline and achieved the second-best performance among the compared approaches, it is selected for this full-scale evaluation. Table 3.4 summarizes the results using the RMSE and the RTE_50 metrics, averaged over 64 testing sequences for each radar. For the proposed method, the L1O strategy is applied during training to ensure that each testing sequence remains unseen by the trained model. The results clearly show that the proposed method significantly outperforms Kellner's method, with average improvements of **51.2%** on translational velocity estimation, **49.8%** on rotational velocity estimation, and **23.2%** on RTE_50.

While the previous evaluation metrics provide a quantitative indication of performance, they do not reveal where the errors originate. To address this, Figure 3.6 presents 2D histograms of estimation errors accumulated over 64 sequences for each radar. The first row shows the results of Kellner's method, while the second row corresponds to the proposed method. At first glance, most errors occur when the rotational velocity is close to zero. This is mainly due to dataset imbalance, since the ego-vehicle spends more time driving straight than turning. Nevertheless, compared to Kellner's method, the proposed approach exhibits

Table 3.4: A full-scale performance comparison between the proposed method and Kellner's method. For each radar, results are averaged over 64 radar sequences. For the proposed method, the leave-one-out approach is used during model training, in order to measure the estimation error for the testing sequence that is unseen by the trained model.

RMSE in Translational Velocity Estimation (m/s)					
Methods	Radar 1	Radar 2	Radar 3	Radar 4	Mean
Kellner's	0.193	0.132	0.253	0.363	0.235
Proposed	0.108	0.0976	0.112	0.142	0.115
Improvement	**44.0%**	**26.1%**	**55.7%**	**60.9%**	**51.2%**
RMSE in Rotational Velocity Estimation (deg/s)					
Methods	Radar 1	Radar 2	Radar 3	Radar 4	Mean
Kellner's	0.802	2.172	2.934	0.882	1.697
Proposed	0.607	1.071	1.049	0.682	0.852
Improvement	**24.3%**	**50.7%**	**64.3%**	**22.7%**	**49.8%**
RTE_50 (m)					
Methods	Radar 1	Radar 2	Radar 3	Radar 4	Mean
Kellner's	11.7	18.8	21.2	6.1	14.5
Proposed	6.5	15.1	17.4	5.5	11.1
Improvement	**44.7%**	**19.7%**	**18.0%**	**10.4%**	**23.2%**

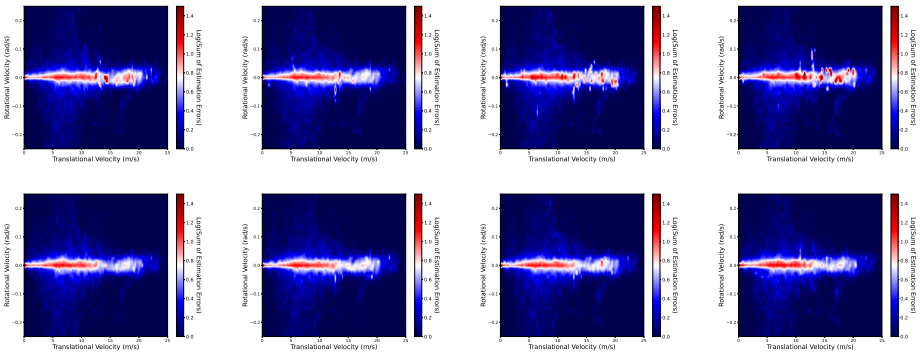


Figure 3.6: The histogram of the estimation errors accumulated over 64 data sequences (an equivalent drive length of over 79 km). The first row shows the results of Kellner's method, while the second row shows the results of the proposed method. Due to the imbalanced ego-motion distribution in the dataset, the natural logarithm is applied to the sum of the estimation errors for better visualization. For the proposed method, the leave-one-out approach is used during training to measure the estimation error for sequences unseen by the trained model.

substantially smaller errors when the vehicle is traveling at high speeds or cornering sharply. From a road safety perspective, reducing errors in high-speed regions is particularly crucial and meaningful.

Table 3.5: Ablation study on input features. Results show the impact of excluding range, returned power, or sample weights compared to the proposed method that uses all features. RMSE on the validation data is reported.

Conditions	Train_loss	Val_loss	RMSE v_x^{car} (m/s)	RMSE ω (deg/s)
No range No power	3.31E-3	7.56E-3	0.118	0.557
No returned power	1.04E-3	5.44E-3	0.101	0.513
No range	1.05E-3	3.71E-3	0.0817	0.479
No sample weights	9.02E-4	2.14E-3	0.0551	0.533
Proposed	**6.18E-4**	**1.19E-3**	**0.0421**	**0.441**

3.4.5. Ablation Study on Input Features

Conventionally, most scan-matching methods use range and AoA measurements, while instantaneous methods rely solely on AoA and radial velocity measurements. However, radar point clouds often contain multi-dimensional features of detected objects. For example, the RadarScenes dataset [15] provides four-dimensional point clouds that include AoA, radial velocity, range, and returned power measurements. Although the proposed method naturally supports multi-dimensional radar point clouds, it is important to examine the contribution of different features for ego-motion estimation. Table 3.5 presents the ablation study on input features. Specifically, the proposed model is trained multiple times with certain features removed, while AoA and radial velocity are retained since they are indispensable. The results show that both range and returned power measurements contribute to improving estimation accuracy. Furthermore, the best performance is achieved when all available features are used, as shown by the results labeled "Proposed."

Nevertheless, the quality of input features can be affected by many factors. As discussed in Section 3.3, the measured radial velocity of a detected object can be influenced not only by its motion relative to the ego-vehicle, but also by measurement errors, object height, and lateral velocity. As a result, the expected radial velocity, computed from the ground-truth ego-motion, may not perfectly match the measured radial velocity. This discrepancy can confuse the training process, as the neural network attempts to learn features invariant to different inputs. To mitigate this issue, an extra weighting term is applied to the proposed loss function to weight the training data. As shown in Table 3.5, the proposed model achieves more accurate pose estimation and yields lower training and validation losses compared with training without sample weighting.

3.4.6. Effect of Pointwise Offset

In contrast to previous works that rely on simulated radar data or data collected in controlled environments, this dissertation uses a challenging real-world radar dataset containing diverse driving scenarios and road conditions. Consequently, the dataset includes not only stationary objects but also a large number of moving objects, false positives, and ghost objects. To address these real-world non-idealities, the proposed neural network learns to regress pointwise weights and pointwise offsets. In the weighted least squares procedure, pointwise weights scale down errors caused by outliers, while pointwise offsets shift radial velocity measurements that deviate significantly from expected values. In other words, the pointwise offset helps further reduce large errors caused by "distant" outliers, such as measurements from fast-approaching vehicles. Table 3.6 presents the impact of including

Table 3.6: The effect of pointwise offset. This experiment is conducted after model training and evaluated using the five test sequences. Specifically, the 'wo_Doppler_offset' re-runs the test sequences and uses only the predicted weights for ego-motion estimation, while the 'Proposed' uses the predicted weights and pointwise offset.

Condition	RMSE in v_x^{car} (m/s)	RMSE in ω (deg/s)	RTE_50 (m)
wo_pointwise_offset	0.0903	0.991	12.589
Proposed	0.0884	0.928	10.118
Improvement (%)	**2.1%**	**6.4%**	**19.6%**

pointwise offsets. In this experiment, the proposed method is tested on the five selected data sequences: first with the predicted pointwise offset, and then without it. The results show that the pointwise offset not only improves estimation accuracy but also enhances long-term stability. However, the improvement is less significant than in the previous ablation study (Table 3.5), since the pointwise weight remains the most influential factor in the linear regression process.

3.4.7. Effect of Doppler Loss

The main objective of this chapter is to estimate the motion of the ego-vehicle. To achieve this, a hybrid model is proposed. Specifically, a neural network is used to exploit the multi-dimensional radar point cloud and assign pointwise weights to each measurement, while the weighted least squares approach estimates the ego-motion. Thus, the key capability the model must learn during training is to identify which measurements are more important than others. A simple way to guide the model is through the motion loss described in Section 3.3. However, without explicit instruction, the model tends to overfit a few key points. As shown in the first row of Figure 3.7, although many measurements should be valid and assigned high weights, most of them receive very small weights if the proposed Doppler loss is not used. In contrast, the second row of Figure 3.7 shows the results with Doppler loss, where the model assigns high weights to measurements close to the ground-truth curve (denoted by the black dashed line). Compared to relying on a few points, incorporating more measurements improves the robustness of the estimator. This improvement is also reflected in the RTE metric, which increases by about **19.5%** when Doppler loss is included. Moreover, with Doppler loss, the proposed method not only estimates the instantaneous ego-motion of the vehicle but also directly indicates the likelihood that measurements originate from stationary objects. Importantly, the predicted weights, as a byproduct of ego-motion estimation, can support other downstream tasks such as environment mapping, multiple object tracking, and instance segmentation.

3.5. Conclusion

A novel deep learning-based method, *DeepEgo*, for instantaneous ego-motion estimation using automotive radar is proposed. Unlike previous works, *DeepEgo* directly processes multi-dimensional radar point clouds with minimal preprocessing. Specifically, it adopts a hybrid architecture that employs neural networks (NNs) for feature extraction and weighted least squares (w-LSQ) for ego-motion estimation. *DeepEgo* is fully differentiable and can be trained end-to-end without manual annotation. Furthermore, it produces pointwise weights

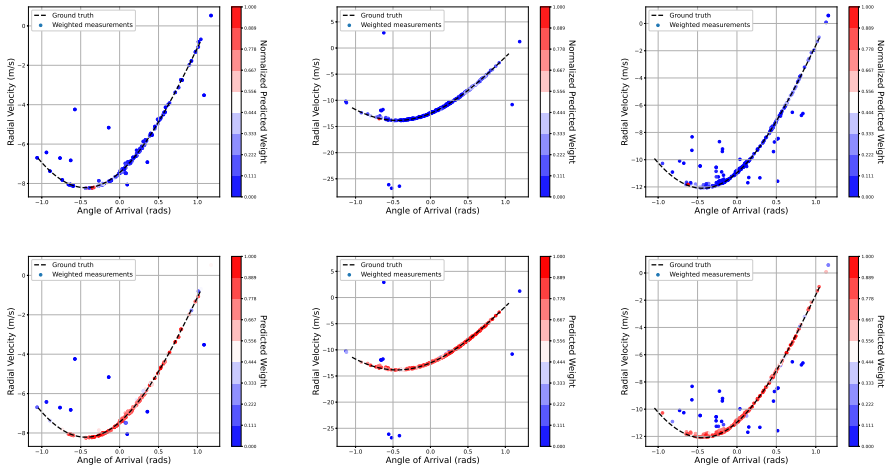


Figure 3.7: The effect of the proposed Doppler loss. For illustration purposes, three frames of radar data are randomly selected and presented as AoA-radial velocity plots. The figures in the first row are the results of the model trained *without* the Doppler loss, while the results in the second row are from a model trained *with* the Doppler loss. The black dashed line represents the expected radial velocity, given the ground-truth ego-motion and the AoA of the detected object. Measured radial velocities are represented by dots, and colors indicate the magnitude of predicted weights. The weights in the first row are normalized by the maximum weight value, while the weights in the second row are direct outputs of the proposed method.

for each radar detection. With the proposed Doppler loss function, these predicted weights are linked to the likelihood that a detection originates from a stationary object. *DeepEgo* was evaluated on a challenging real-world radar dataset, where it demonstrated superior performance in estimation accuracy, long-term stability, and runtime performance compared to existing methods. For future research, it should be noted that this chapter focuses solely on ego-motion estimation with a single sensor type. Although radar sensor fusion is an active research area [88–90], the full potential of automotive radar for ego-motion estimation remains underexplored. Consequently, various fusion approaches (e.g., homogeneous and heterogeneous sensor fusion) are not addressed here and are left for future investigation. Other promising directions include exploring alternative regression techniques such as orthogonal distance regression (ODR) [91], or enhancing the robustness of the proposed method to different radar installation configurations.

4

DeepEgo+: Multi-Radar Fusion for Robust Motion Estimation

The previous chapter presented DeepEgo, a radar-only solution for vehicle ego-motion estimation. While effective, DeepEgo is limited to single-radar setups, even though modern vehicles are often equipped with multiple radars. A natural extension, therefore, is to adapt it for multi-radar sensor fusion. This is especially important for ego-motion estimation, a safety-critical application where redundancy and robustness are essential. Sensor fusion provides a means to exploit overlapping information and improve system reliability. Designing such a fusion approach, however, is not straightforward. Most existing studies assume tightly synchronized radar networks, which is a demanding and often unrealistic requirement. Moreover, it remains unclear how data from multiple radars should be combined: at the raw input stage, at the feature level, or only after independent estimation. This chapter introduces DeepEgo+, an extension of the original framework designed to address these challenges and enable robust ego-motion estimation with multiple radars.

Parts of this chapter have been published in:
S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "DeepEgo+: Unsynchronized Radar Sensor Fusion for Robust Vehicle Ego-Motion Estimation," in IEEE Transactions on Radar Systems, vol. 3, pp. 483-497, 2025.
S. Zhu, A. Yarovoy, F. Fioranelli, S. Ravindran, L. Chen, "Methods and devices for multi-radar, multi-frame ego-motion estimation", application number: PCT/US2024/031392, filed in 2024, accessible as WO2025250125A1.

4.1. Introduction

Automotive radar is a stronger candidate for vehicle ego-motion estimation. Radar-based methods usually estimate vehicle motion using the measured range, azimuth, and radial velocity of detected stationary objects (inliers). However, the performance of such methods is hampered by several shortcomings of radar sensors. For example, due to their low spatial resolution, radar point clouds are typically sparse and contain less geometric information about the sensed environment [92]. Therefore, directly matching consecutive radar point clouds (such as for localization methods developed for optical sensors [32]) rarely achieves good results, and customization and heuristics are often required [46]. Furthermore, the quality of radar data used for ego-motion estimation is susceptible to phenomena such as false alarms, multipath reflections, RCS fluctuations, and detection of moving objects in the scene [16]. These effects can produce a large number of detections that are not from stationary objects (i.e., outliers), and thus confuse motion estimation methods. To address this issue, recent studies [93–95] use random sampling techniques such as the random sample consensus (RANSAC) [61] or its variants [96–98]. However, these techniques are often iterative, inefficient, susceptible to high outlier ratios, and unable to distinguish slow-moving objects due to binary thresholding.

As introduced in Chapter 3, a neural network (NN)-based method, named *DeepEgo* [57], was proposed to deal with the above limitations. Additionally, unlike other NN-based methods [42, 99, 100], *DeepEgo* can directly process multi-dimensional radar point clouds without introducing quantization errors. To benchmark, *DeepEgo* was evaluated using a challenging real-world radar dataset, RadarScenes [15], and the result demonstrates its superior performance compared to previous studies in terms of estimation accuracy and robustness. Even so, there are still some aspects of *DeepEgo* that can be further improved. Firstly, *DeepEgo* is designed for single-radar scenarios, while today's cars are usually equipped with multiple temporally misaligned radar sensors (i.e., forming a radar sensor network). It is therefore worthwhile to investigate what benefits sensor fusion can bring to ego-motion estimation, and how fusion with temporally misaligned radars can be achieved while still maintaining the advantages of *DeepEgo*. Secondly, even though *DeepEgo* proposed a novel loss function to supervise model training, it is not robust to low-quality training examples (e.g., radar point clouds associated with incorrect true vehicle motion, or radar point clouds with almost no inliers). Last but not least, *DeepEgo* relies on radial velocity measurements of inliers to compute vehicle motion, but the impact of vehicle acceleration on radial velocity estimation has not been identified, let alone the relevant remedies for this.

To fill the above gaps, this study proposes a novel neural network, named *DeepEgo+*, for vehicle ego-motion estimation using a temporally misaligned radar sensor network. To the best of the authors' knowledge, compared with previous studies such as [12, 13, 57, 97], the proposed method provides the following advantages:

1. **Lack of synchronization:** To work with temporally misaligned radars, *DeepEgo+* uses a decentralized signal processing architecture: the output of each radar node is first processed independently before implementing sensor fusion. To the best of the authors' knowledge, this is the first work in the field of radar-based ego-motion estimation that works with temporally misaligned [1] radar sensor networks.

[1] The term "temporally misaligned" means that the radar sensors in the network are not synchronized, i.e., the time

2. **Robustness:** Several modifications to the original loss function of *DeepEgo* [57] are proposed to improve its robustness to low-quality training examples. In addition, the designed decentralized architecture with the late fusion approach [101] further enhances its robustness against situations such as sensor failure[2] or erroneous radar node output. Furthermore, *DeepEgo+* shows significant improvement over previous studies in terms of robustness to data with high outlier ratios.

3. **Acceleration:** Although this is not one of the primary goals, *DeepEgo+* demonstrates for the first time in the literature its ability to offset the impact of the vehicle's non-zero acceleration on ego-motion estimates after point cloud generation. Specifically, the normalized cross-correlation between estimation error and vehicle acceleration becomes 3.5 times smaller than the prior art, due to the proposed temporal NNs.

In addition to the above contributions, *DeepEgo+* also inherits all the advantages of the previous *DeepEgo* model [57]. Furthermore, *DeepEgo+* is tested using the challenging RadarScenes dataset [15], which is the same dataset used to measure the performance of *DeepEgo* and six other works [12, 13, 42, 47, 60, 86] selected from the literature[3]. Last but not least, a set of novel evaluation metrics and visualization tools are introduced to gain a comprehensive understanding of the performance of *DeepEgo+*. Compared with previous works, *DeepEgo+* shows a significant improvement in terms of estimation accuracy, long-term stability, and robustness against high outlier ratios and sensor failures.

The rest of the chapter follows this organization. Section 4.2 presents an overview of existing research on the topic. Section 4.3 presents the detailed design of *DeepEgo+*. In Section 4.4, the performance of *DeepEgo+* is measured using different evaluation metrics, and other approaches found in the literature are also tested as a comparison. Finally, Section 4.5 offers conclusions and outlines directions for future studies.

4.2. Related Work

This section provides an overview of the prior work in the area of (A) vehicle localization with automotive radar, and (B) sensor fusion with automotive radar. After that, a summary of the literature study is provided.

4.2.1. Vehicle Localization with Automotive Radar (Continued)

In general, radar-based vehicle localization approaches can be categorized into two main groups: scan-matching methods [12, 45–51] and instantaneous methods [13, 52–57]. Since most of the related work has already been reviewed in Chapter 3, Section 3.2, this section focuses exclusively on summarizing *DeepEgo* to avoid redundancy.

As introduced earlier, *DeepEgo* is a neural network (NN)-based instantaneous method proposed to overcome the limitations of conventional RANSAC-based approaches. Instead of relying on RANSAC, *DeepEgo* directly processes multi-dimensional radar point clouds

intervals between individual radar outputs are uneven, and the radar transmitting & receiving operations are not ordered.

[2] Sensor failure is denoted as a condition in which the radar sensor stops functioning properly, or temporarily fails to provide accurate data.

[3] Chapter 3 presents in more detail the performance of these six previous studies on the RadarScenes dataset.

using NNs to extract complex spatial features and estimate point weights, which are subsequently applied in a weighted regression. Although *DeepEgo* effectively addresses several shortcomings of instantaneous methods, there remain important aspects that can be further improved. Firstly, modern vehicles are typically equipped with multiple radars that capture complementary perspectives of the environment and can therefore enhance estimation accuracy and robustness [39]. However, *DeepEgo* is designed for a single radar, and it remains unclear how to fuse data from multiple radars, particularly when the sensors are temporally misaligned. Secondly, as mentioned earlier, the performance of instantaneous methods depends on accurate measurements of the inliers' radial velocity. Yet, non-zero vehicle acceleration can introduce inconsistent Doppler phase shifts, resulting in an expanded Doppler frequency range and, consequently, inaccurate radial velocity estimation [102]. Finally, *DeepEgo* introduces a novel loss function that supervises the network in learning how to assign weights to inliers and outliers. However, this loss function is not robust to imperfections in real-world training data. For instance, a training sample may be associated with incorrect ground truth labels, or it may be unusable in practice if no inliers are detected by the radar.

4.2.2. Sensor Fusion with Automotive Radars

Perception sensors are essential for achieving full automation in self-driving cars. As discussed in Chapter 1, different sensor types offer distinct advantages and disadvantages. Moreover, even within the same sensor type, variations in configuration and mounting position can lead to differences in performance. The main objective of sensor fusion is therefore to combine sensor data in order to overcome the limitations of individual sensors [103]. Achieving effective sensor fusion typically requires two key design choices: the selection of a fusion architecture and the choice of a fusion technique.

Fusion Architecture

According to the sensor type, fusion architectures can be divided into heterogeneous sensor fusion (HTSF) [89, 90, 104–108] and homogeneous sensor fusion (HMSF) [52, 109, 110]. HMSF focuses on fusing the same type of sensors, while HTSF considers fusing radar with other sensor modalities such as IMU [42, 89, 104, 106, 108], camera [105], or LiDAR [107]. Judging from the amount of literature, HTSF appears to have received more attention than HMSF. This is because HTSF can combine the advantages of different types of sensors and overcome their respective weaknesses. However, HMSF offers better scalability than HTSF, which makes it relatively straightforward to add more sensors to the system. Additionally, by fusing multiple radars, HMSF can improve the overall signal-to-noise ratio and use redundancy to increase reliability. Furthermore, future autonomous vehicles are likely to have greater sensor redundancy, meaning there may be multiple sensors of the same and different types [111]. Therefore, it is also necessary to study HMSF and realize that future fusion architectures can be hybrid, where HMSF can first simplify the complexity of data integration and reduce computing power, and then HTSF combines the advantages of different modalities.

Fusion Technique

Fusion techniques determine at what point in a data processing chain the sensor data are fused. These techniques can generally be divided into early fusion [52, 110], halfway fusion

[42, 105], and late fusion. Early fusion combines sensor data at a very early stage, such as in [52], where multiple radar point clouds are fused to estimate the vehicle ego-motion. Halfway fusion performs fusion at an intermediate stage, for example, [42] uses different NNs to extract features from radar and odometry sensors separately and then fuses them. Late fusion combines the final output of each sensor after processing the data from each sensor separately. Therefore, it is clear that the later the fusion occurs, the less information is preserved. However, this also means that early and halfway fusion can be computationally expensive. Moreover, in order to fuse data at an early stage, early and halfway fusion usually require good sensor synchronization [52] or restrictions on radar configuration and mounting location [109]. In contrast, late fusion is more robust to sensor failures than other fusion techniques because each sensor is processed independently, making it a preferred choice for safety-critical applications. Furthermore, the computational cost of late fusion techniques increases linearly with the number of sensors.

4.2.3. Summary

Based on the literature review, the following conclusions can be drawn. Firstly, instantaneous methods demonstrate greater potential than scan-matching methods, with the recently proposed *DeepEgo* [57] standing out in particular. Nevertheless, to accommodate future sensor networks and to handle complex driving scenarios with challenging radar data, *DeepEgo+* incorporates updated loss functions, a decentralized architecture, and enhanced robustness to address the identified gaps. Secondly, although HTSF has received more attention than HMSF in the literature, future fusion architectures are likely to require both approaches, which highlights the relevance of the present study. Finally, for driving safety and in the context of temporally misaligned radar sensor networks, late fusion is preferable to other fusion techniques.

4.3. Proposed Method

This section presents the design of the proposed method *DeepEgo+* for vehicle ego-motion estimation using multiple temporally misaligned automotive radars.

4.3.1. Problem Formulation

This chapter focuses on the problem of estimating vehicle 2D ego-motion using temporally misaligned radar sensors in real traffic scenarios. Given a temporally misaligned radar sensor network mounted on a moving vehicle, where each radar has a linear array and the vehicle is not sliding sideways, after signal preprocessing [17], each radar outputs a multi-dimensional radar point cloud with J points and each is associated with M features. Assuming the point cloud contains points only originating from detected static objects, and each point is associated with at least a radial velocity d_j and azimuth angle α_j measurements (i.e., $M \geq 2$), the following equation holds:

$$\begin{bmatrix} -d_1 \\ \dots \\ -d_J \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ \dots & \dots \\ \cos(\alpha_J) & \sin(\alpha_J) \end{bmatrix} \cdot \begin{bmatrix} v_x^{rad} \\ v_y^{rad} \end{bmatrix} \tag{4.1}$$

where v_x^{rad} and v_y^{rad} are the 2D motion components of the radar in radar coordinates.

Equation (4.1) can be simplified as:

$$\mathbf{D} = \mathbf{A} \cdot \mathbf{V} \tag{4.2}$$

Given at least two independent detection points ($J \geq 2$ and $\mathbf{A}$ has full column rank), the vector $\mathbf{V}$ can be estimated using ordinary least squares:

$$\mathbf{V}^{est} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{D} \tag{4.3}$$

However, real-world radar data often contains a large number of outliers that negatively impact the least-squares solution. Conventional instantaneous methods employ RANSAC to address this issue, whereas *DeepEgo* [57], as introduced in Chapter 3, proposes the use of weighted least squares (w-LSQ):

$$\mathbf{V}^{est} = (\mathbf{A}^T\mathbf{W}^{est}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^{est}\mathbf{D}, \quad with\ \mathbf{W}^{est} = \mathrm{diag}(w_1, \dots, w_J) \tag{4.4}$$

$\mathbf{W}^{est}$ is a diagonal matrix with J positive weights estimated by *DeepEgo*. Given the radar motion, under the zero-lateral-velocity assumption, the vehicle ego-motion can be computed by:

$$\mathbf{e}^{est} = \begin{bmatrix} v_x^{car} \\ \omega \end{bmatrix} = \begin{bmatrix} v_x^{rad}\cos(\theta) - v_y^{rad}\sin(\theta) + y\omega \\ \frac{1}{x}(v_y^{rad}\cos(\theta) + v_x^{rad}\sin(\theta)) \end{bmatrix} \tag{4.5}$$

where v_x^{car} is the vehicle translational velocity, and ω is the rotational speed. x, y are the mounting position and θ is the mounting angle of the radar with respect to the vehicle coordinates. Based on the radar sensor network, given T consecutive timestamps, Equation (4.5) can deliver T vehicle velocity estimates arranged in chronological order across I radar sensors:

$$\left\{\mathbf{e}_{i,t}^{est}\right\}_{i=1\dots I, t=1\dots T} \tag{4.6}$$

However, it is worth repeating that the above velocity estimates can be inaccurate due to the issues mentioned in Section 4.2.1. To further improve this, as concluded in Section 4.2.3, homogeneous sensor fusion with the late fusion technique can be applied. Since the vehicle motion has temporal correlation across frames, the Kalman filter (KF) [112] can be used as one of the alternative solutions to fuse these motion estimates. Assuming a linear motion model and additive Gaussian perturbation, the process model and measurement model can be expressed as follows:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{q}_t,\ \mathbf{q}_t \sim N(\mathbf{0}, \mathbf{Q}_t) \\ \mathbf{y}_t &= \mathbf{G}_t\mathbf{x}_t + \mathbf{p}_t,\ \mathbf{p}_t \sim N(\mathbf{0}, \mathbf{P}_t) \end{aligned} \tag{4.7}$$

Here, $\mathbf{x}_t$ is the vehicle kinematic state, $\mathbf{y}_t$ is the initial velocity estimate $\mathbf{e}_t^{est}$ from Equation (4.6), $\mathbf{F}_t$ is the transition matrix, and $\mathbf{G}_t$ is the measurement matrix. $\mathbf{q}_t$ and $\mathbf{p}_t$ represent the process and measurement noise, respectively. Additionally, $\mathbf{q}_t$ and $\mathbf{p}_t$ are assumed to be white, independent, and Gaussian-distributed with covariance matrix $\mathbf{Q}_t$ and $\mathbf{P}_t$. Therefore, according to [113], the vehicle motion can be computed by:

$$
\begin{aligned}
\hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} \\
\mathbf{H}_{t,t-1} &= \mathbf{F}_t \mathbf{H}_{t-1,t-1} \mathbf{F}_t^T + \mathbf{Q}_t \\
\mathbf{K}_t &= \mathbf{H}_{t,t-1} \mathbf{G}_t^T (\mathbf{G}_t \mathbf{H}_{t,t-1} \mathbf{G}_t^T + \mathbf{P}_t)^{-1} \\
\hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{y}_t - \mathbf{G}_t \hat{\mathbf{x}}_{t|t-1}) \\
\mathbf{H}_{t,t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{G}_t) \mathbf{H}_{t,t-1}
\end{aligned}
\tag{4.8}
$$

where $\hat{\mathbf{x}}_{t|t-1}$ is the predicted state estimate, $\mathbf{H}_{t,t-1}$ is the predicted estimate covariance, $\mathbf{K}_t$ is the Kalman gain, $\hat{\mathbf{x}}_{t|t}$ is the updated state estimate, and $\mathbf{H}_{t,t}$ is the updated estimate covariance. In order to compute $\hat{\mathbf{x}}_{t|t}$, it is essential to know $\mathbf{F}_t$, $\mathbf{G}_t$, $\mathbf{Q}_t$, $\mathbf{P}_t$, and $\mathbf{y}_t$. While $\mathbf{F}_t$ and $\mathbf{G}_t$ can be determined based on the assumed motion model and timestamps, $\mathbf{y}_t$ is related to the point cloud weights $\mathbf{W}^{est}$ as in Equation (4.4), and $\mathbf{Q}_t$ and $\mathbf{P}_t$ are unknown and often require adaptive adjustment.

4.3.2. Proposed Architecture Overview

Figure 4.1 presents the architecture of *DeepEgo+*. It is a decentralized signal processing architecture with two NN-based modules, Module A for motion estimation and Module B for fusion. Module A is implemented at each radar node and, similar to *DeepEgo* [57], can directly process multi-dimensional radar point clouds and output initial motion estimates. However, with the proposed modifications, Module A addresses several limitations of *DeepEgo*. For example, with the new loss function, Module A is more robust to low-quality training examples in the dataset. Also, Module A employs additional NN layers to reduce the complexity of input features so that it can process data from all radars and share the same parameters among different nodes. This parameter sharing significantly reduces the number of training parameters and keeps the network size constant even if the number of radar nodes increases. After collecting T initial estimates from Module A, Module B fuses them using the proposed NN-based KF. The NN-based KF mimics the procedure of the conventional KF to exploit the temporal correlation in vehicle motion and provide smoother estimates. The difference is that it avoids laborious parameter tuning, and instead uses temporal NNs to adaptively estimate the Kalman gain. Additionally, as shown in Section 4.4.7, the temporal NNs can learn to offset the effects of non-zero vehicle acceleration. Moreover, Module B can be used as a smoothing function that provides T velocity estimates, or as a filter that updates the velocity based on the latest timestamp.

In summary, *DeepEgo+* provides an end-to-end solution for ego-motion estimation using temporally misaligned radar sensor networks. Through a special network design, it addresses the previous limitations mentioned in Section 4.2. Furthermore, the proposed decentralized processing architecture helps reduce the computational overhead and improve the runtime performance by distributing the processing tasks across multiple radar nodes. The remainder of this section presents its design details. Specifically, the modifications that upgrade *DeepEgo* [57], which works for single-radar and single-frame, to *DeepEgo+*, which works for multi-radar and multi-frame, will be presented.

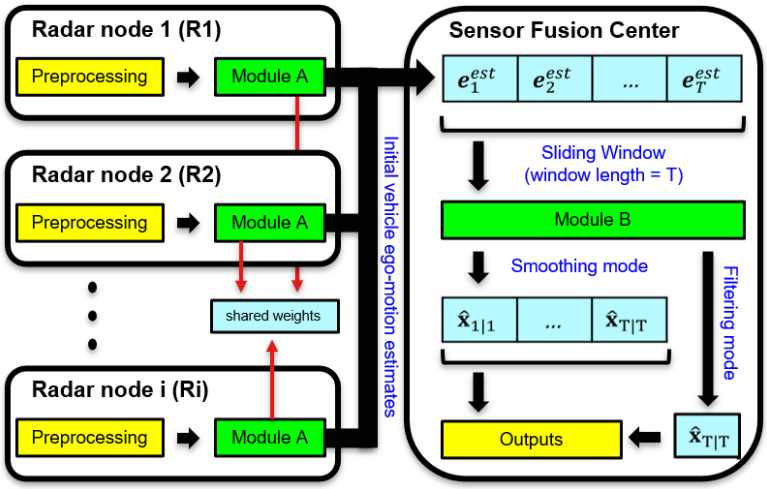


Figure 4.1: The architecture of *DeepEgo+*. It is a decentralized processing architecture with two NN-based components, Module A and Module B. Each radar has a Module A that processes its output and calculates an initial velocity estimate. After collecting T initial estimates, with late fusion, Module B fuses them and outputs the updated estimates for multiple timestamps (smoothing mode), or for the latest timestamp (filtering mode).

4.3.3. Improved Loss Function

This section introduces the proposed modifications to the loss function of *DeepEgo*. The original loss function comprises three components: motion loss, Doppler loss, and sample weighting. The detailed formulations and roles of these components were presented in Section 3.3.3. Here, the focus is placed on their limitations and the corresponding solutions. The main limitations can be summarized as follows. First, the motion loss employs the mean squared error between the estimation and ground truth, which may cause the model to be dominated by large residuals arising from rare or extreme cases. Second, the sample weighting mechanism sums the weights of all points, thereby favoring training examples with low outlier ratios. However, examples with moderately high outlier ratios are also important, since traditional instantaneous methods tend to fail under such conditions. Third, the Doppler loss minimizes the squared distance between estimated and ground-truth point weights. As a result, the model is also forced to match points with small weights caused by slow-moving objects. This can bias the final w-LSQ solution and increase the risk of overfitting. Finally, the overall loss of *DeepEgo* can penalize cases in which the Doppler loss is large while the motion loss is small. Yet, due to vehicle acceleration (see Section 4.4.7), the Doppler loss can remain large even after motion filtering and acceleration compensation, while the motion loss becomes smaller. To address these issues, the following modifications are proposed:

1. The new motion loss uses the Huber loss [114] instead of MSE, which is more robust to outlier scenarios.

$$\ell_{\text{Motion}}^{(b)} = \mathcal{L}_\delta(c_x) + \mathcal{L}_\delta(c_y), \tag{4.9}$$

where $c_x = v_{x,(b)}^{rad,gt} - v_{x,(b)}^{rad,est}$, $c_y = v_{y,(b)}^{rad,gt} - v_{y,(b)}^{rad,est}$, b is the batch index, δ is a threshold parameter, and

$$\mathcal{L}_\delta(c) = \begin{cases} \frac{1}{2}c^2 & \text{if } |c| \leq \delta, \\ \delta\left(|c| - \frac{1}{2}\delta\right) & \text{otherwise.} \end{cases} \tag{4.10}$$

2. The new sample weighting calculates the mean of all inlier weights to represent the quality of inliers. In addition, to completely eliminate the influence of low-quality training examples (as shown in Figure 4.2), the new sample weighting is set to zero when the mean weight or the number of inliers is below a given threshold.

$$\begin{aligned} \mathcal{J}_{(b)} &= \left\{ j \in \{1, \ldots, J\} \ : \ w_{j,(b)}^{gt} \geq 0.01 \right\}, \\ n_{(b)} &= |\mathcal{J}_{(b)}|, \\ m_{(b)} &= \begin{cases} \frac{1}{n_{(b)}} \sum_{j \in \mathcal{J}_{(b)}} w_{j,(b)}^{gt}, & \text{if } n_{(b)} > 0, \\ 0, & \text{if } n_{(b)} = 0, \end{cases} \\ s^{(b)} &= \begin{cases} m_{(b)}, & \text{if } m_{(b)} \geq 0.4 \text{ and } n_{(b)} \geq 40, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.11}$$

3. The new Doppler loss measures the mean squared error between the K points with the largest weights in $\mathbf{W}_{(b)}^{est}$ and their actual weights in $\mathbf{W}_{(b)}^{gt}$. In addition, the w-LSQ solution, as shown in Equation (4.4), is computed using the selected K points instead of all J points.

$$\ell_{\text{Doppler}}^{(b)} = \frac{1}{K} \sum_{j \in \mathcal{Z}_{(b)}} \left(w_{j,(b)}^{gt} - w_{j,(b)}^{est} \right)^2, \quad \mathcal{Z}_{(b)} = \text{TopK}\left(\{ w_{j,(b)}^{est} \}_{j=1}^{J} \right), \tag{4.12}$$

4. The final loss function of *DeepEgo+* is a multiplication of the motion loss, Doppler loss, and sample weighting.

$$\mathcal{L}_{all} = \frac{1}{B} \sum_{b=1}^{B} \left(\ell_{\text{Motion}}^{(b)} \cdot \ell_{\text{Doppler}}^{(b)} \cdot s^{(b)} \right) \tag{4.13}$$

For simplicity, in the rest of this chapter, *DeepEgo* with the above proposed modifications to its loss function is named *Model V1*. Similar to *DeepEgo*, *Model V1* estimates ego-motion based on input from the same single radar. However, a radar sensor network consists of multiple radar sensors installed at different locations with different viewing angles. As

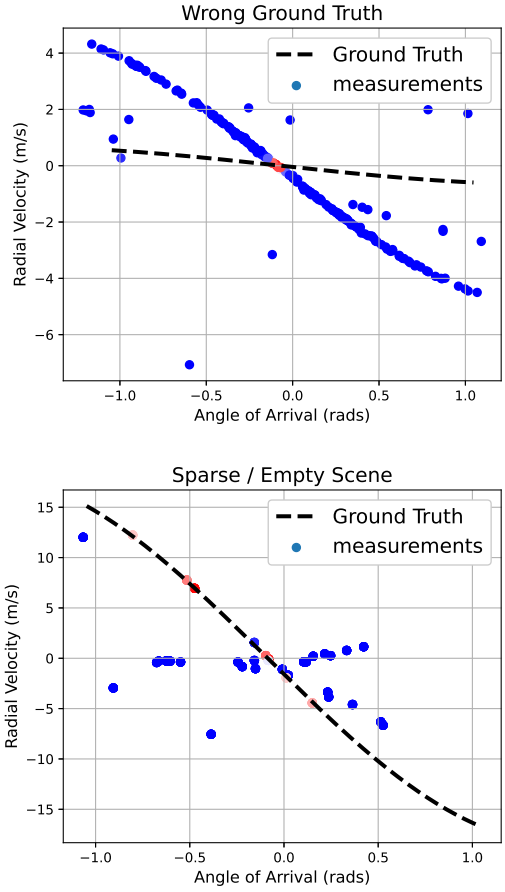


Figure 4.2: Examples of imperfect real-world radar data from the RadarScenes dataset [15]. For both sub-figures, radar measurements are denoted as colored dots (red inliers, blue outliers), and the black dashed line is computed based on the current ground truth vehicle motion and AoA measurements. The upper figure shows what happens when the ground truth velocity is incorrect, and the lower figure provides an example of a sparse/empty radar point cloud.

illustrated in Figure 4.1, to handle the complex input data features caused by different radar nodes, the next section proposes several modifications to the architecture and input structure of *Model V1*, beyond the aforementioned changes to the loss function.

4.3.4. Module A: Processing Multi-radar Nodes

In our previous work, *DeepEgo* [57] is trained and tested separately and independently for each radar. Experimental results show that its performance degrades when using different radars for training and testing. In general, the more training data, the better performance

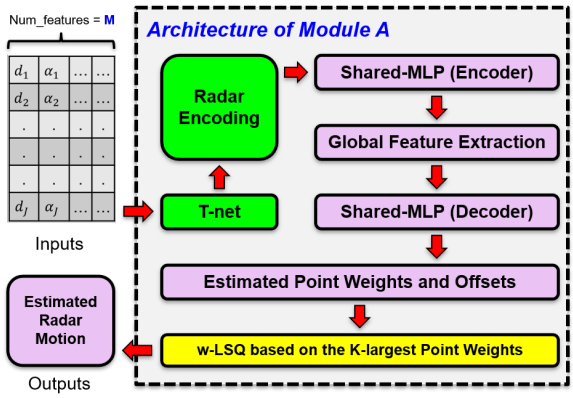


Figure 4.3: The architecture of Module A. Purple blocks are included in *DeepEgo* [57]. As explained in Section 4.3.3, *Model V1* computes the w-LSQ solution based on the K largest point weights, with the relevant modification shown in yellow. Based on *Model V1*, Module A adds a T-net [77] and a radar encoding step (denoted in green) to handle arbitrary radar inputs.

a deep learning model should produce. However, further experiments indicate that when *DeepEgo* is trained and tested using data from all radars of the sensor network, its performance decreases compared to the independently trained per-radar models. The reason behind this phenomenon is that as the training data increases, the input feature distribution becomes complex and multi-modal, due to different radar mounting positions, orientations, and FoVs, which makes it more challenging to share parameters across radar nodes and to generalize.

To address this challenge, T-net [77] and a radar encoding step are added to *Model V1*, and the final neural network is named Module A. Figure 4.3 shows the architecture of Module A and explains the differences in the architecture between *DeepEgo*, *Model V1*, and Module A. T-net is a neural network that takes a multi-dimensional radar point cloud as input, outputs a matrix, and uses the matrix to perform an affine transformation on the input point cloud. This step can reduce the complexity of input features by aligning point clouds captured by different radars into a canonical space. Afterwards, a radar encoding step is implemented to concatenate a predefined number (e.g., with two radars the numbers would be ”0” and ”1”) with the output of the affine transformation. This radar encoding step can further help Module A distinguish among different feature patterns caused by different radars. As shown in the results in Section 4.4.3, Module A can improve performance even when the model is trained with more complex data. However, it is important to acknowledge that Module A can only work with known radar sensor networks. For testing using data from “unseen” radar nodes, further developments are required.

4.3.5. Module B: Processing Multi-radar Frames

As illustrated in Figure 4.1, the outputs of Module A are the vehicle motion estimates, arranged in time order. Although these motion estimates come from different radar nodes, they are all sampled from the continuous motion of the vehicle. Therefore, to exploit the advantage of multiple radars and the prior knowledge of the vehicle motion model, this work

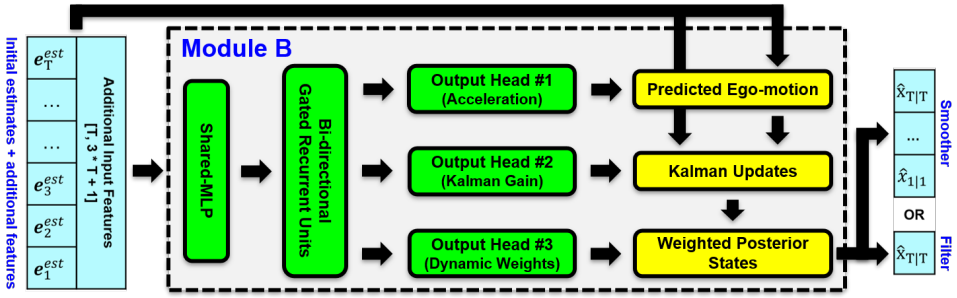


Figure 4.4: The architecture of the proposed neural networks for sensor fusion (named Module B in this work). Within Module B, only the green blocks are built by neural networks. Module B takes the initial vehicle motion estimates provided by Module A as its input. Then, it uses bidirectional gated recurrent units (bi-GRUs) to extract temporal features from motion estimates. These features are further processed by three output heads, each of which estimates the matrices used to compute the Kalman prediction, Kalman update, and weighted output. Finally, Module B can also be used as a smoother to output T motion estimates, or as a filter to update the most recent estimate.

proposes to use a Kalman filter (KF) to fuse these initial estimates. Following Equation (4.8) for the prediction and update steps of the KF, although most variables can be determined, the covariance matrices $\mathbf{Q}_t$ and $\mathbf{P}_t$ are unknown and usually need to be adaptively adjusted based on the certainty of the model predictions and measurements. Instead of manually tuning the covariance matrix, inspired by [115], this work proposes using NNs to estimate the Kalman gains. In this way, the explicit knowledge of the underlying noise statistics is no longer required. Additionally, based on the input, the estimated Kalman gain is able to automatically determine how much weight the filter gives to predictions (from the model) versus measurements (from the radar) when updating the state estimate.

Figure 4.4 presents the architecture of the proposed neural networks (collectively named Module B). Module B consists of three NN-based functional components: (1) a shared multi-layer perceptron (shared-MLP), (2) the bidirectional gated recurrent units (bi-GRUs), and (3) three output heads. The shared-MLP takes T initial vehicle motion estimates, T radar encoding labels, and three $T \times T$ feature matrices as its input. The initial vehicle motion estimates are provided by Module A and the radar encoding label is the same as described in Section 4.3.4. Regarding the feature matrices, the first two are generated through the outer subtraction[4] for the vector of initial motion estimates and recorded timestamps, respectively. The third feature matrix is given by the division of the first two feature matrices and represents the estimated vehicle acceleration. The shared-MLP is used to preprocess the raw input data and transform it into a representation that is more suitable for the subsequent bi-GRUs. The bi-GRUs are recurrent neural networks (RNNs) that can extract temporal features from continuous vehicle motions. Unlike traditional unidirectional RNNs, the bidirectional architecture allows leveraging information from previous motion states and upcoming motion states to make predictions about the current state. With this feature, Module B with sliding window input can be used not only as a filter but also as a

[4]The outer subtraction is an operation that subtracts every element of a vector from every other element, resulting in a matrix with zeros on the diagonal.

smoother. The output of the bi-GRUs is further processed by three output heads, each consisting of 2 layers of the shared-MLPs. The first output head estimates a $T \times T$ acceleration matrix τ. The acceleration matrix has the same structure as the previously-mentioned third feature matrix and contains the same type of information; it can be expressed as follows:

$$\tau = \begin{bmatrix} \tau_{1\leftarrow 1} & \tau_{1\leftarrow 2} & \cdots & \tau_{1\leftarrow T} \\ \tau_{2\leftarrow 1} & \tau_{2\leftarrow 2} & \cdots & \tau_{2\leftarrow T} \\ \cdots & \cdots & \cdots & \cdots \\ \tau_{T\leftarrow 1} & \tau_{T\leftarrow 2} & \cdots & \tau_{T\leftarrow T} \end{bmatrix} \tag{4.14}$$

Each element of τ represents an estimated acceleration value, and the index notation (for example '$\kappa \leftarrow \lambda$') indicates that this element is used to make a prediction/update at timestamp κ using previous/future states from timestamp λ. With the same structure, the second output head estimates a $T \times T$ Kalman gain matrix, and the third outputs a $T \times T$ dynamic weight matrix. Therefore, following the yellow blocks in Figure 4.4, the prediction step is achieved by multiplying the second feature matrix representing the time difference with the acceleration matrix τ, and then added to the initial motion estimates. Afterwards, following the terminology used in KF theory, the innovation matrix is computed using the predicted motion states and the initial motion estimates as measurements. The innovation matrix is then multiplied with the Kalman gain matrix, outputting a correction matrix which is then added to the predicted motion state forming an updated (posterior) state matrix. The updated state matrix has the same structure and formulation as the matrix in Equation (4.14), where each row represents T potential updates for the timestamp κ. However, only one update is needed for each timestamp. To solve the data association problem, the dynamic weight matrix is multiplied with the updated state matrix. The weight matrix, whose rows sum to 1, acts as a soft data association, combining all potential updates via a weighted average. Finally, the output of Module B is a vector of T updated states, and the output can be used as a smoother or a filter.

4.3.6. Implementation Details

As explained in previous sections, *DeepEgo+* consists of two key components: Module A and Module B. Module A is used to process multiple radar nodes and compute initial motion estimates, and Module B fuses these initial estimates using an NN-based KF. Their relevant implementation details are as follows:

Module A Firstly, the new Doppler loss and the new w-LSQ scheme are computed using the K largest point weights. In this study, K is set to 224 based on empirical evaluation on the RadarScenes dataset [15]. Secondly, the T-net [77] used in Module A has three shared-MLPs as an encoder, one max-pooling layer, and another three shared-MLPs as a decoder for the output. Thirdly, for a sensor network with X radars, the radar encoding mentioned in Section 4.3.4 and Section 4.3.5 is a natural number ranging from 0 to $X - 1$. Fourthly, the parameters of all purple blocks of Module A in Figure 4.3 are presented in Chapter 3. Lastly, as mentioned in Section 4.3.3, the sample weight is set to zero unless the mean weight $m_{(b)}$ and the number of inliers $n_{(b)}$ is greater than 0.4 and 40, respectively[5].

[5]Note that these two hyperparameters are determined empirically and they may change slightly if the model is trained with a different radar dataset.

Module B. Firstly, all shared-MLPs in this work have the same structure, which consists of a 1D convolutional layer, a batch normalization layer, and a ReLU layer for non-linearity. Secondly, to reduce overfitting, a dropout layer with a dropout rate of 0.2 was attached to the bi-GRUs. Thirdly, for the three output heads in Module B, the top layer of the first head (acceleration) has no activation function (linear output), the top layer of the second (Kalman gain) uses the hyperbolic tangent activation function, and the third head (dynamic weights) applies the softmax activation function.

4.4. Results and Discussion

This section presents the evaluation results of the proposed method *DeepEgo+*. Moreover, methods from the literature are evaluated and compared.

4.4.1. Dataset and Evaluation Protocol

This chapter uses the same RadarScenes subset and baseline evaluation protocol described in Chapter 3.4.2. Specifically, scenes with an ego-vehicle driving distance greater than 500 m are selected, performance evaluation is conducted at the radar level, and training and testing follow a leave-one-out protocol at the sequence level unless otherwise stated.

Section 4.4.3 also evaluates five selected scenes using data from all four radars, but adopts a multi-radar training strategy. During training, data from all radars and all scenes except the test scene are jointly used for model training and validation. Performance is then evaluated separately for each radar and averaged over the selected test scenes.

Unlike the previous two settings, Section 4.4.4 performs testing at the scene level, as the model explicitly exploits temporal information across multiple frames. For training, data from all radars and all scenes except the test scene are temporally ordered based on the provided timestamps and used jointly for model training. During testing, data from the four radars of the test scene are likewise temporally ordered before evaluation. Final results are obtained by averaging the performance across the five test scenes. Sections 4.4.5 and 4.4.7 further extend this evaluation protocol to all 64 selected scenes instead of five.

4.4.2. Single-Radar and Single-Frame

As detailed in Section 4.3, from the previous *DeepEgo* [57] to the current *DeepEgo+*, there are three intermediate steps. The first step is to modify its loss function and the w-LSQ scheme (referred to as *Model V1*). Then, its network structure is modified in order to process data from multiple radars (using the previously defined *Module A*). Lastly, to fuse information from multiple radar nodes at multiple timestamps, a neural network-based Kalman filter/smoother is proposed (the previously defined *Module B*). Although *DeepEgo+* is the combination of Module A and Module B, it is helpful to understand how model performance changes with these intermediate steps. Therefore, as an ablation study, the evaluation results of these steps are presented in the current and the following subsections.

Table 4.1 presents the performance comparison between *DeepEgo* and Model V1. With the proposed modifications, Model V1 outperforms *DeepEgo* in almost all evaluation metrics, except the RMSE metric in translational velocity estimation. However, this outcome is expected since the proposed loss function (i.e., the Huber loss and the new sample weight scheme) makes the model less affected by large errors caused by bad training data. Addi-

Table 4.1: Performance comparison between *DeepEgo* [57] and the proposed Model V1 for single-radar single-frame scenarios. Their performance is measured using the radar sequences from the five testing scenes also selected in [57]. With four automotive radars and five scenes, the metrics of each method are computed based on the average of 20 radar sequences.

	Translation Velocity (cm/s)				**Rotational Velocity (deg/s)**				**RTE (m)**
Metrics	RMSE	S-RMSE	MedAE	MAE	RMSE	S-RMSE	MedAE	MAE	RTE_50
DeepEgo [57]	11.3	9.3	4.6	6.8	0.77	0.74	0.36	0.53	9.5
Model V1	11.6	9.1	4.4	6.6	0.72	0.67	0.32	0.48	8.5
Improvement(%)	-1.9%	+2.1%	+5.4%	+3.0%	+5.9%	+8.6%	+11.1%	+10%	+9.8%

Table 4.2: Performance comparison between *DeepEgo* [57] and Module A. For *DeepEgo*, the model is trained and tested separately for each radar, while Module A is trained and tested using data from all four radars. The term "R1" denotes "Radar 1". The metric S-RMSE is used to measure errors in translational velocity estimation (v_x^{car} in cm/s) and rotational velocity estimation (ω in deg/s). The final result is averaged over five test radar sequences.

Metrics: S-RMSE		R1 Tested		R2 Tested		R3 Tested		R4 Tested	
		v_x^{car}	ω	v_x^{car}	ω	v_x^{car}	ω	v_x^{car}	ω
DeepEgo	R1 Trained	**9.5**	**0.64**	30.4	2.37	43.1	2.62	43.4	2.04
	R2 Trained	41.4	2.27	**9.2**	**0.88**	37.0	2.50	42.6	2.32
	R3 Trained	44.3	2.27	36.5	2.57	**8.7**	**0.81**	37.9	2.12
	R4 Trained	44.3	1.99	43.1	2.60	33.1	2.41	**9.9**	**0.63**
Module A	All Radars	**9.2**	**0.62**	**8.9**	**0.77**	**8.5**	**0.71**	**9.3**	**0.59**

tionally, the proposed S-RMSE metric reflects how these rare faulty data samples in the test set can affect the final evaluation results. Compared to the RMSE metric, Model V1 scores better and has a higher percentage improvement in the S-RMSE metric for both translational and rotational ego-motion estimation.

4.4.3. Multi-Radar and Single-Frame

This subsection presents the evaluation results when comparing *DeepEgo* [57] and Module A. As shown in Table 4.2, Module A performs better than *DeepEgo* in the S-RMSE metric. Additionally, it is able to estimate the ego-motion using data from four radars installed at different locations on the vehicle. It is worth mentioning that, due to different aspect angles, the relationship between ego-motion and the measurements of radial velocity and AoA can vary significantly from radar to radar. Consequently, this diversity adds additional challenges to the neural network model due to the increased complexity of input features, which leads to performance degradation when training and testing using data from radars mounted at different locations. However, with the proposed modifications, the model performance remains similar despite the increased complexity, and the model only needs to be trained once.

4.4.4. Multi-Radar and Multi-Frame

As previously depicted in Figure 4.1, *DeepEgo+* consists of two components, Module A & Module B used jointly. Module A processes data coming from each different radar sequentially according to their global timestamps. It estimates the point weights and computes initial velocity estimates through w-LSQ. After that, Module B takes multiple initial esti-

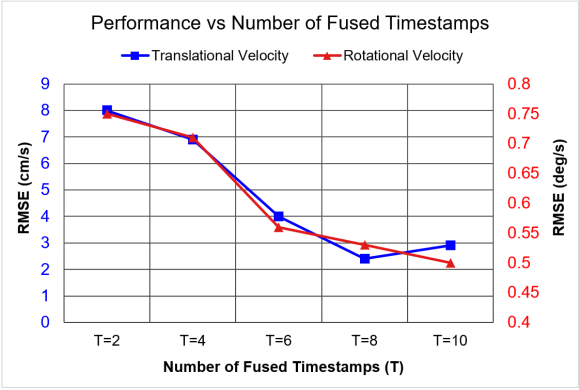


Figure 4.5: Performance of *DeepEgo+* under different numbers of fused timestamps (T). The left vertical axis shows the relationship between T and RMSE in translational velocity estimation, and the right axis provides the error in rotational velocity estimation.

mates as its input and updates the final estimates through smoothing or filtering operations. As the number of timestamps (or initial estimates) for Module B is adjustable, it becomes one of the critical hyperparameters. Clearly, this parameter determines the number of timestamps, denoted as T, to be fused, and also determines how much past and future information can be used for estimation. As shown in Figure 4.5, the translation and rotational velocity estimation errors decrease as T increases. This is logical because the more information the model has about the vehicle's motion, the more accurate estimates it can provide. However, it is not worth setting T to too large a value, as not only will this take up more memory space, but it will also have diminishing returns. Therefore, unless otherwise specified, based on these results T is set to 8 in the following experiments.

As, to the best of our knowledge, there is currently no research on ego-motion estimation using temporally misaligned radar sensor networks, several traditional fusion methods are selected to benchmark *DeepEgo+*. Since these traditional fusion methods cannot directly process radar point clouds, the output of Module A is used as their input. Therefore, these selected methods are compared with the proposed Module B. As shown in Figure 4.6, the proposed method outperforms the other approaches by a large margin, for both translational and rotational ego-motion estimation. Furthermore, unlike other methods, the proposed approach does not require laborious human engineering to adjust parameters after model training. Finally, as discussed later in Section 4.4.7, there is a fundamental limitation why the proposed method outperforms other traditional approaches in dealing with non-zero acceleration, and to the best of our knowledge there is no solution in the relevant literature so far to address this.

4.4.5. Performance on All Data

In order to provide a comprehensive evaluation, the performance of *DeepEgo+* is measured with the L1O method based on the entire dataset, consisting of 64 scenes. The evaluation results are presented in Table 4.3. Firstly, it is worth mentioning that Module A is trained with all radars together, while *DeepEgo* [57] is trained independently and must be opti-

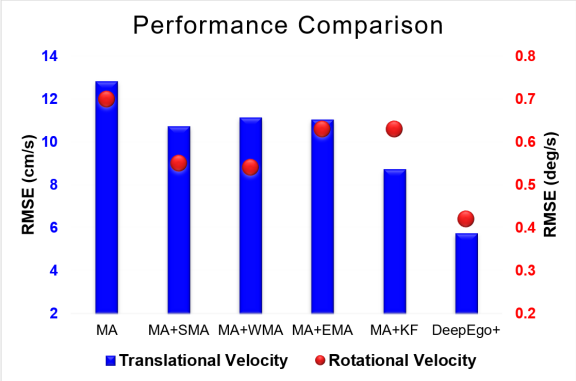


Figure 4.6: Performance of the proposed approach *DeepEgo+* and other fusion methods selected for comparison, including simple moving average (SMA), weighted moving average (WMA), exponential moving average (EMA), and Kalman filter (KF). For a rigorous and fair comparison, these selected methods take the output of the proposed Module A (MA) and *DeepEgo+* is set to its filtering mode. With five test scenes and four radars, the RMSE metric presented in the chart is measured and averaged over 20 test radar sequences.

Table 4.3: Comprehensive comparison evaluated using all 64 scenes in the RadarScenes dataset [15] and the five proposed metrics. The proposed method *DeepEgo+* is tested under both its filtering mode and smoothing mode. Also, the previous approach *DeepEgo* [57] and the proposed Module A are added for comparison. The 'Improvement' row is computed based on the original method in [57] compared with the proposed method *DeepEgo+* in its smoothing mode.

	Translation Velocity (cm/s)				**Rotational Velocity (deg/s)**				**RTE**
Metrics	RMSE	S-RMSE	MedAE	MAE	RMSE	S-RMSE	MedAE	MAE	RTE_50
DeepEgo [57]	11.5	9.0	4.7	6.7	0.85	0.74	0.40	0.56	11.1
Module A	10.6	8.6	4.4	6.3	0.73	0.66	0.34	0.48	9.9
DeepEgo+ [filtering]	6.1	4.2	2.0	3.0	0.51	0.49	0.23	0.34	6.4
DeepEgo+ [smoothing]	5.3	3.6	1.6	2.5	0.44	0.41	0.20	0.29	6.4
Improvement (%)	+53.9%	+60.0%	+66.0%	+62.7%	+48.2%	+44.6%	+50.0%	+48.2%	+41.4%

mally tuned for each radar. Nevertheless, Module A outperforms *DeepEgo* in all presented evaluation metrics. Also, the proposed method performs better in its smoothing mode than in its filtering mode. This is reasonable since in the smoothing mode both past and future information of vehicle motion is available.

While Table 4.3 provides exact numbers for the estimation errors, it is difficult to see how these errors are distributed and their physical interpretation. Figure 4.7 shows the histogram of all estimation errors accumulated across all 64 testing scenes. It is shown that the proposed method achieves notably lower error than DeepEgo [57], especially when the vehicle is turning or driving at a high speed. While the previous results provide a comprehensive study of the performance of the proposed method, it is important to acknowledge that the performance comparison is in part limited, given the following reasons. Firstly, to the best of our knowledge, there are no other works in the open literature that studies instantaneous ego-motion estimation using temporally misaligned radar sensor networks. Secondly, the original method *DeepEgo* in [57] had already shown better performance than several alternative works, so their performance results are not listed in this section. Thirdly, other fusion methods shown in Figure 4.6 do not score as well as *DeepEgo+* due to their

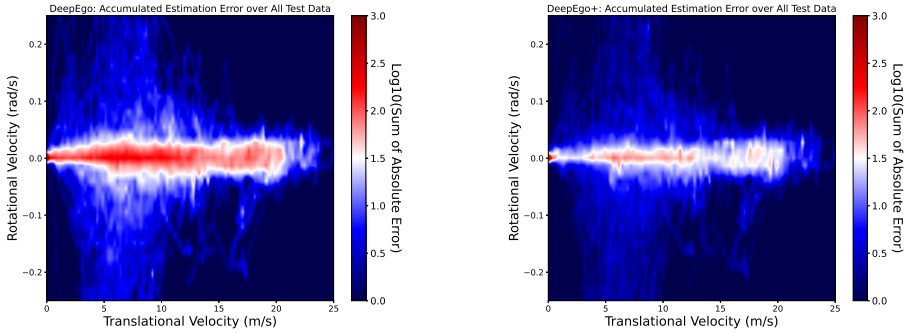


Figure 4.7: Plots of the accumulated error from all 64 testing scenes. To plot this, the absolute errors between the ground truth ego-motion and the model estimates are accumulated, and then the base-10 logarithm is applied for better visualization. The left figure shows the result of the original *DeepEgo* [57], and the right figure shows the proposed method *DeepEgo+*.

limitations, which will be further discussed in Section 4.4.7.

4.4.6. Effect of Sensor Failure

As described in Section 4.3, this study proposes a novel decentralized neural network architecture for temporally misaligned radar sensor fusion. It first computes the initial vehicle motion estimates based on each radar's output, and then fuses these initial estimates based on the order of timestamps. Therefore, due to the decentralized processing structure, the proposed method should be able to work with radar sensor networks that are smaller than the one used for training. It is clear that this feature increases the system's robustness against sensor failure. Thus, it is interesting to simulate sensor failure scenarios and to investigate how model performance deteriorates as the size of the radar sensor network decreases during testing, essentially assuming that data from some of the radars in the network are missing.

Figure 4.8 presents an example of a reconstructed vehicle trajectory which visually illustrates the effect of sensor failure. In this test, after training on data from all radars, the model is tested using only data from the selected radar(s) with the assumption that some radars are malfunctioning, hence not providing any data. It can be seen that when all radars are working, the method provides the best estimation performance as the estimated trajectory approaches the ground-truth trajectory. Even if half of the sensors fail, the proposed method still maintains good performance similar to the original *DeepEgo* [57]. However, it is noted that it is not recommended to intentionally apply the proposed architecture (DeepEgo+) to a too small radar sensor network compared to what was originally used for training. Although the proposed architecture provides robustness against sensor failure, it is best adapted to the size of the sensor network in the training data. As shown in the figure, when only one radar is working out of the original four used for training, the proposed method *DeepEgo+* performs worse than *DeepEgo* [57].

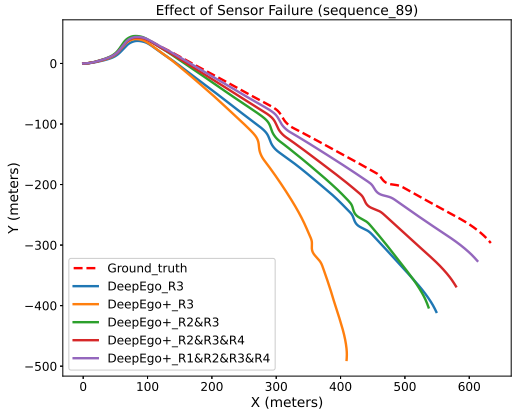


Figure 4.8: The effect of radar sensor failure on the reconstructed vehicle trajectories. The proposed method *DeepEgo+* is trained using all radars, but tested with a varying number of missing/malfunctioning radars. The radar data used for testing do not contribute to model training and validation. The reconstructed trajectories are shown in solid lines, and the ground truth vehicle trajectory as the red dashed line. The term "R3" denotes "Radar 3", the same applies to other terms.

4.4.7. Vehicle Non-Zero Acceleration

One of the key advantages of using radar for automotive applications is that it can directly provide radial velocity measurements of the detected objects. It is known that, for stationary objects, the measured radial velocities are related to aspect angles and the vehicle speed. However, if the speed of the vehicle is not constant, the Doppler frequency and the associated phase shift will vary with slow time (i.e., across radar chirps), and the estimated radial velocity will not match the vehicle speed, which can be instantaneously determined by the odometer sensor. Therefore, the vehicle speed can be overestimated or underestimated, depending on whether the vehicle is decelerating or accelerating. For a better understanding, Figure 4.9 shows the impact of vehicle non-zero acceleration on the estimated vehicle motion. It can be seen from Figure 4.9a that the vehicle first decelerates, then maintains an almost constant speed, and finally accelerates. As expected, Module A overestimates the velocity when the vehicle decelerates, underestimates it when the vehicle accelerates, and overlaps with the ground truth only when the vehicle is at a constant speed. In contrast, the proposed *DeepEgo+* can accurately estimate the velocity of the vehicle regardless of its motion state.

To understand the reasons for this, Figure 4.9b and Figure 4.9c are helpful. They show two radar frames in AoA and radial velocity plots for the vehicle at a constant speed and while decelerating, respectively. Firstly, when the vehicle speed is constant, the radial velocities are not being overestimated or underestimated. As a result, the line representing the ground truth overlaps with the lines representing Module A and *DeepEgo+*, and they pass through detection points caused by stationary objects. As discussed in Section 4.2, locating the stationary points has been widely used by many radar-based ego-motion estimation methods, including the present work. However, as shown in Figure 4.9c, when the

Table 4.4: Normalized cross-correlation (NCC) between vehicle acceleration and measured errors in vehicle translational velocity estimation. NCC is computed using radar frames from all 256 testing scenes (including four radars). To control for other independent variables that may affect the measurement error, only radar frames with an outlier rate lower than 40% were used.

Methods	RANSAC-based [13]	DeepEgo[57]	DeepEgo+
NCC	0.71	0.60	**0.17**

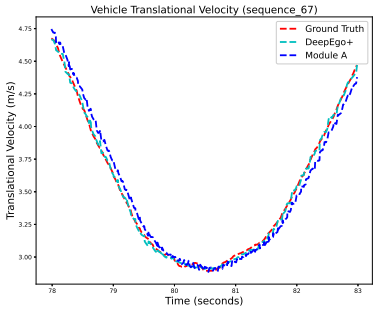
vehicle decelerates, the ground truth line does not match the measured radial velocity of the stationary object, and the radial velocity is underestimated. For methods that rely on accurate radial velocity measurements [13, 47, 57], this phenomenon can significantly affect their performance. For example, Module A and other instantaneous methods (mentioned in Section 4.2.1) will never be able to estimate the true vehicle motion in Figure 4.9c due to underestimation. In addition, it is clear that none of the filtering approaches evaluated in Figure 4.6 is able to correct the estimates from Module A to the ground truth motion. However, since the magnitude of overestimation and underestimation is positively correlated with the magnitude of vehicle acceleration, *DeepEgo+* is able to compensate for its effect by learning its patterns from consecutive radar frames using the proposed temporal neural network within Module B. Note that the data used for evaluation in this section are not being used for model training and validation, which demonstrates the good generalization ability of *DeepEgo+*.

While the previous experiment has intuitively demonstrated the effectiveness of *DeepEgo+* in addressing the impact of vehicle acceleration, Table 4.4 presents a numerical relationship between velocity estimation error and vehicle acceleration using the normalized cross-correlation (NCC). Based on the results, the RANSAC-based instantaneous method [13] and the previous approach *DeepEgo* [57] exhibit a strong correlation between acceleration and estimation error. Meanwhile, *DeepEgo+* shows much weaker correlations than previous studies. Quantitatively, *DeepEgo+* performs 3.5 times better than *DeepEgo*.

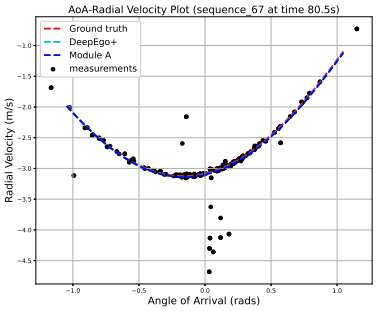
Finally, it must be acknowledged that addressing the challenges induced by non-zero vehicle acceleration was not an explicit goal when developing the proposed approach. Additionally, to the best of our knowledge, none of the works in the literature mentioned in Section 4.2 have investigated this problem or proposed a solution. However, this is deemed to be an essential issue because in realistic automotive scenarios, vehicles will have non-zero acceleration most of the time, especially in urban areas. Although the proposed method is very effective and addresses this challenge, the investigation can be continued in dedicated future work.

4.4.8. Robustness to Outliers

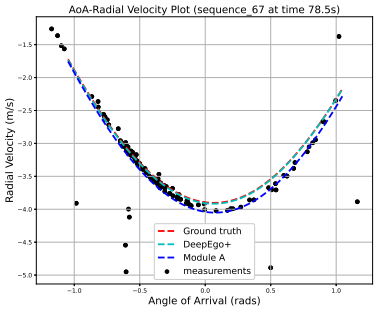
As discussed in Section 4.2.1, one of the main challenges of instantaneous methods is to distinguish inliers from the input radar point cloud, while outliers caused by moving objects and false detections are ubiquitous in real-world driving. Figure 4.10 compares the performance of *DeepEgo+* with two previous studies in terms of the S-RMSE metric measured using different outlier ratios. Firstly, it is reasonable that when the outlier ratio is less than 50%, that is, inliers are in the majority and it is difficult to make mistakes, the score of the RANSAC-based instantaneous method is similar to *DeepEgo*. Despite this, *DeepEgo+*



(a) The vehicle's translational velocity over time.



(b) AoA-radial velocity plot (constant speed).



(c) AoA-radial velocity plot (decelerating).

Figure 4.9: The impact of vehicle non-zero acceleration on ego-motion estimation. In this test, scene 67 ("sequence_67") of the RadarScenes dataset is used as test data. Figure (a) shows a sequence of continuous vehicle motion in which the vehicle first slows down and then accelerates. Figure (b) and Figure (c) show the relationship between AoA and radial velocity measurements at two selected timestamps; measurements are denoted as black dots. For all figures, the red, light-blue, and dark-blue dashed lines are computed based on the vehicle ground-truth motion, the output of DeepEgo+, and the output of Module A, respectively.

still outperforms both comparison methods. It is assumed that this improvement is due to the proposed temporal NNs (Module B), which offset the effect caused by vehicle acceleration. Secondly, when the outlier ratio is between 50% and 90%, where estimation errors are likely to occur, the advantage of the NN-based instantaneous method becomes obvious. Still, *DeepEgo+* has improved performance by nearly 50% compared to *DeepEgo*, demonstrating the importance of sensor fusion. Finally, when the outlier ratio exceeds 90%, the performance of all methods drops significantly because there is little useful information to be captured.

4.4.9. Effect of Vehicle Rotation Rate

In total, the selected scenes contain approximately 432k radar frames, among which around 6k frames (about 1.5%) are captured while the ego-vehicle is performing sharp turns, defined here as rotation rates exceeding 20 deg/s. This motivates an analysis of whether the proposed

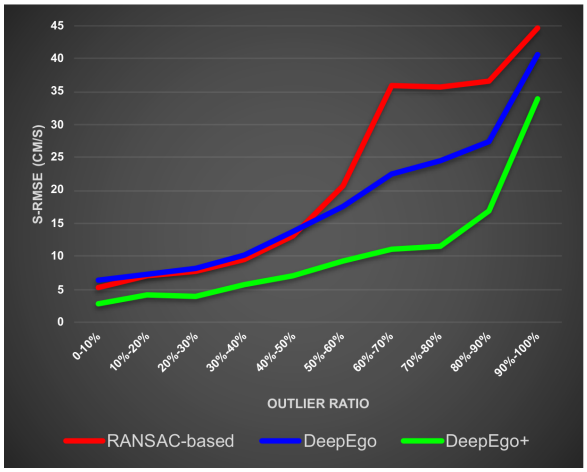


Figure 4.10: The measured S-RMSE error in vehicle translational velocity estimation over different outlier ratios. All frames in the 64 test scenes are regrouped according to their outlier ratios. The red line represents the RANSAC-based instantaneous method [13], the previous approach *DeepEgo* [57] is represented by the blue line, and the proposed *DeepEgo+* is denoted by the green line.

method is affected by "aggressive" vehicle maneuvers. To study this effect independently of outlier ratio, Figure 4.11 shows the S-RMSE metric as a function of vehicle rotation rate, computed using only radar frames with a low outlier ratio (below 10%). As shown in the figure, the dataset spans a wide range of rotation rates, including sharp turns of up to approximately 50 deg/s. However, under low outlier conditions, the estimation error does not increase monotonically with rotation rate for any of the compared methods. Instead, the error remains relatively stable across different rotation rate intervals. Taken together with the results in Figure 4.10, this analysis indicates that performance degradation is primarily driven by measurement or environmental conditions (i.e., high outlier ratios), rather than by vehicle rotation rate or sharp maneuvers alone.

4.4.10. Sensitivity to Point Density

Although the RadarScenes dataset [15] was collected using advanced automotive radars with a range resolution of 0.15 m and an angular resolution of 0.5 deg at boresight, the resulting spatial resolution is still significantly coarser than that of typical automotive LiDAR systems. As a result, the radar point clouds provided by the RadarScenes dataset may contain only a limited number of detection points. These detections are generated after CFAR algorithms, and the point clouds can become even sparser due to aggressive CFAR thresholding or environmental factors such as scenes dominated by free space.

As discussed in Chapter 3.2.1, sparse point clouds pose a significant challenge for scan-matching based ego-motion estimation methods, which typically rely on stable and detailed environment contours to align consecutive radar frames. It is therefore of interest to examine whether instantaneous ego-motion estimation methods exhibit similar sensitivity to point cloud density. In total, the selected scenes from the RadarScenes dataset contain about 432k radar frames, among which around 28k frames (about 6.6%) contain fewer than 60 detection

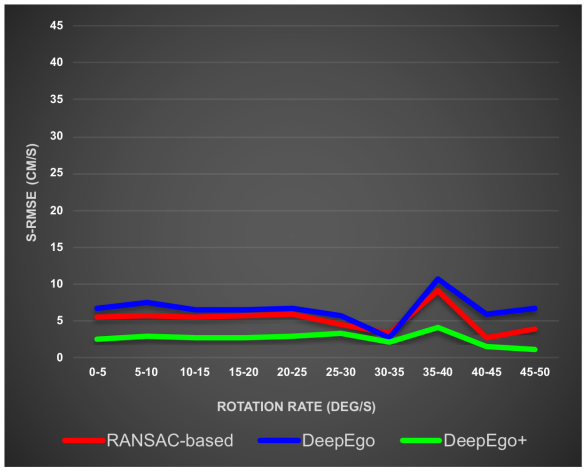


Figure 4.11: The measured S-RMSE error in vehicle translational velocity estimation over different vehicle rotation rates, computed using radar frames with an outlier ratio below 10%. The y-axis range is matched to Figure 4.10 to enable direct comparison of error magnitudes.

points, with the minimum being only 27 points. Figure 4.12 evaluates the sensitivity of estimation performance for the compared methods by grouping radar frames according to the number of detection points, while conditioning on a low outlier ratio (below 10%). As shown, the estimation error remains relatively stable across a wide range of point densities for all compared methods, even in sparse scenarios with fewer than 90 points, no significant degradation is observed.

These results indicate that sparse radar point clouds alone do not necessarily lead to estimation failure when the majority of detections correspond to static objects. This highlights a key advantage of instantaneous methods over scan-matching approaches for ego-motion estimation. Combined with the outlier ratio analysis in Section 4.4.8 and the rotation rate analysis in Section 4.4.9, these findings further support the conclusion that performance degradation in instantaneous methods is primarily driven by high outlier ratios rather than by vehicle sharp turns or sparse radar point clouds.

4.5. Conclusion

This chapter presents a novel solution for estimating the 2D motion of a moving vehicle (ego-motion) equipped with multiple temporally misaligned automotive radar sensors. The proposed method, named *DeepEgo+*, consists of two neural network (NN)-based components: Module A and Module B. *DeepEgo+* successfully addresses several limitations and challenges outlined in the relevant literature for ego-motion estimation. Firstly, *DeepEgo+* uses a decentralized signal processing architecture to provide the first solution for ego-motion estimation using temporally misaligned radar networks. Specifically, the output of each radar is processed locally and independently by Module A without the need for good synchronization between sensors. Secondly, based on the previous work of *DeepEgo* [57], Module A has extensive upgrades to its original loss function and network architec-

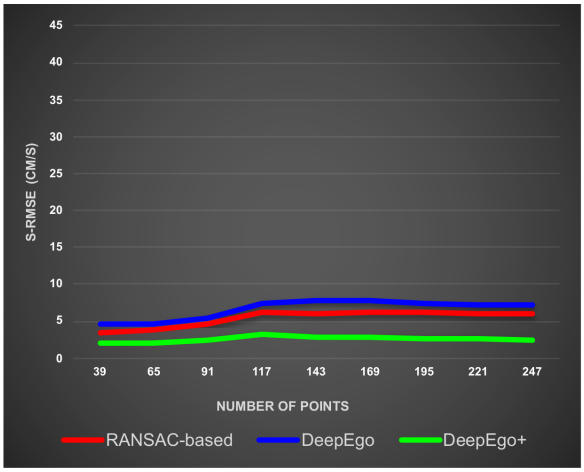


Figure 4.12: The measured S-RMSE error in vehicle translational velocity estimation over different numbers of points in the input radar point cloud, computed using radar frames with an outlier ratio below 10%. The y-axis range is matched to Figure 4.10 to enable direct comparison of error magnitudes.

ture, improving its robustness and learning capabilities. Thirdly, Module B uses temporal NNs to mimic the Kalman filter procedures and estimate the Kalman gain, thus bypassing the conventional, tedious manual tuning process. The late fusion approach applied in Module B further improves the system robustness to sensor failures and radar frames with high outlier ratios, where conventional methods fail. Fourthly, *DeepEgo+* addresses the challenges posed by vehicle non-zero acceleration to ego-motion estimation, and proposes the first effective solution. To verify its performance, *DeepEgo+* is tested using a challenging real-world radar dataset (RadarScenes) containing various driving scenarios on the road. The results demonstrate its superiority over a range of evaluation metrics compared to other methods in the literature. For future research, it would be interesting to compare the performance of early, halfway, and late fusion techniques for radar-based ego-motion estimation.

5

Radar Mounting Angle Estimation under Operational Driving Conditions

Accurate radar extrinsic parameters are essential for ego-motion estimation. The transformation from radar motion to vehicle motion relies heavily on correct values for mounting positions and angles. Conventionally, these parameters are measured in the factory and assumed to remain fixed. In real-world conditions, however, vibrations, material aging, and even minor collisions can alter them, leading to degraded perception performance. Despite its importance, this problem has received little attention. Most existing approaches either depend on special calibration targets, require controlled environments, or rely on carefully designed driving routes—conditions rarely met in practice. This chapter addresses the challenge by proposing a method to estimate radar mounting angles under ordinary driving conditions. The approach is designed to provide accurate mounting angle estimates with low variability across diverse and realistic driving scenarios, filling an important gap in the literature and contributing to more reliable radar-based vehicle perception.

Parts of this chapter have been published in:
S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "Radar Mounting Angle Estimation in Operational Driving Conditions," in IEEE Transactions on Radar Systems. (under review)

5.1. Introduction

Automotive radar has been widely adopted for tasks such as ego-motion estimation [13], environment mapping [116], moving object tracking [117], and road user classification [118]. A common prerequisite for all these applications is accurate knowledge of the radar's extrinsic parameters, in particular its mounting angle relative to the vehicle. Although such parameters are typically set at the production stage, they can drift over time due to vibrations, collisions, or accidents. Even small deviations can severely degrade performance. For instance, as shown in Figure 5.1, a misalignment of only 0.05° in radar mounting angle can cause substantial localization errors. Errors can also be amplified by radar's long detection range, degrading crucial tasks such as mapping and sensor fusion [119]. Regular estimation of the radar mounting angle during operational driving conditions is therefore very important [120].

Traditional approaches determine this angle using handheld compasses, angle sensors, or radar housings with accelerometers and actuators [121–123]. However, these methods are costly, labor-intensive, and often require skilled engineers, making them impractical for regular in-vehicle calibration. Recent research has therefore shifted toward algorithmic solutions [119], which rely on measurements from the radar under calibration together with an additional reference sensor. While these approaches aim to enable fast and accurate mounting angle estimation that operates automatically under normal driving conditions, most existing methods have only been validated in controlled environments [63], with special targets [124], or on carefully designed driving routes [125].

In practice, the key challenge in radar mounting angle estimation is not the formulation of the estimation formula itself, but the quality of the available radar measurements under operational driving conditions. In dense traffic, a large fraction of radar detections originate from moving objects unsuitable to be used as references for calibration purposes; vehicle acceleration affects Doppler estimation; and radar point clouds can be sparse and flickering. These factors collectively cause existing calibration methods to fail unless restrictive assumptions, such as static scenes or controlled routes, are imposed.

This work addresses this bottleneck by formulating a mounting angle estimation approach that remains applicable under normal, uncontrolled driving conditions. Based on the proposed formulation, the approach combines complementary onboard sensor measurements within a kinematic framework to enable reliable radar mounting angle estimation under challenging real-world measurement conditions. For comprehensive evaluation, this study examines two problem formulations and four estimation techniques within this framework. Experiments use the challenging *RadarScenes* dataset [15], covering more than 79 km of urban driving across varied traffic and environmental conditions, as well as varying velocities and trajectories of the ego-vehicle. To the best of our knowledge, this study demonstrates that radar mounting angles can be accurately estimated in unconstrained real-world traffic, without relying on controlled environments, dedicated radar targets, or specific driving maneuvers.

The rest of this chapter is organized as follows. Section 5.2 reviews related work. Section 5.3 presents the proposed signal processing pipeline. Section 5.4 reports evaluation results and comparisons. Section 5.5 concludes with key findings and future research directions.

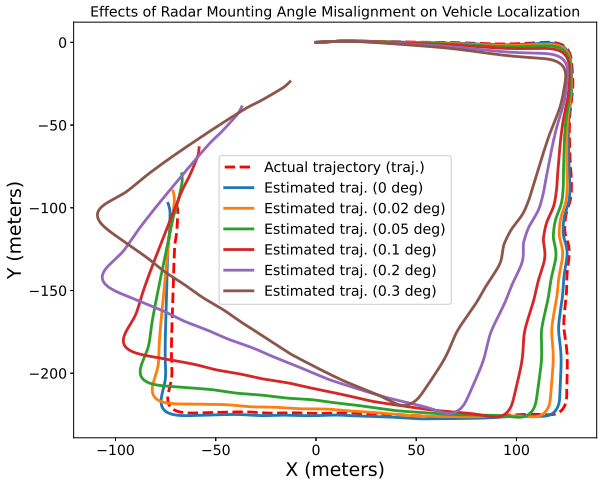


Figure 5.1: The negative impacts of radar mounting angle misalignment on vehicle localization. The vehicle localization algorithm uses extrinsic parameters to convert the estimated radar trajectory into a vehicle trajectory. However, when the extrinsic parameters are incorrect, additional errors will be injected after the conversion.

5.2. Related Work

The main objective of radar mounting angle estimation is to determine the relative angle between the principal beam direction of the radar sensor and the thrust axis of the vehicle. To operate automatically, without human supervision, a second sensor with known extrinsic parameters is typically required as a reference. The relative angle between the radar and the reference sensor is first estimated, and then converted to the vehicle's thrust axis using the reference sensor's extrinsic parameters.

According to the type of reference sensor, existing methods can be divided into four categories: camera-based [126–128], lidar-based [124, 129, 130], radar-based [63, 131, 132], and odometry-based [133–135]. Camera- and lidar-based approaches benefit from the high resolution and rich information provided by these optical sensors. However, their performance is strongly affected by adverse weather and lighting conditions. In addition, some require overlapping fields of view (FoV) between the radar and the reference sensor, which limits where the radar can be physically mounted [128, 129]. Radar-based approaches use another automotive radar as the reference. Compared with optical sensors, they are less affected by environmental conditions. However, they often require strict conditions such as synchronized radar sensors [63, 131], specially designed radar targets (e.g., corner reflectors) [132, 136], or overlapping FoV [137]. These requirements are difficult to meet in real driving scenarios. In contrast, odometry-based approaches avoid these limitations. Odometry sensors operate reliably under all weather conditions and, due to their high refresh rate, do not suffer from synchronization issues. Most odometry-based methods rely on ego-velocity measurements: by comparing the ego-velocities measured by the odometry sensor and the radar, their relative transformation can be estimated using rigid-body

kinematics [138]. This eliminates the need for challenging processing steps such as feature extraction and data association, which are common in other approaches [127, 132]. Despite these advantages, odometry-based methods still face two major limitations:

1. **Sensitivity to outliers.** To handle complex driving scenarios, most odometry-based methods use random sample consensus (RANSAC) [61] or its variants [139] to mitigate the impact of measurement noise and moving objects (i.e., outliers) on radar motion estimation [134, 135, 138]. However, RANSAC is an iterative algorithm with several parameters to tune and assumes that most radar measurements originate from stationary objects. Its performance degrades significantly when outliers dominate, as in dense traffic with many moving targets. Consequently, many studies evaluate their methods only in controlled environments where most surrounding objects are static [134, 135, 140], leaving performance under realistic driving scenarios less extensively validated.

2. **Limited treatment of vehicle acceleration effects.** A second limitation is that most odometry-based methods ignore the effects of vehicle acceleration and deceleration [141]. Acceleration causes range and Doppler migration [142], leading to inaccurate radar motion estimates [143]. To mitigate this, [119] only used radar data when vehicle acceleration was below 0.5 m/s^2. However, such a fixed threshold is impractical in real driving scenarios.

In summary, odometry-based methods offer clear advantages over other approaches. However, the diversity of road users and the dynamic behavior of vehicles pose significant challenges for their application in real-world driving scenarios. Moreover, while different problem formulations and estimation techniques have been proposed [134, 138], a comprehensive comparison of their bottlenecks and trade-offs is still lacking.

5.3. Proposed Method

This section presents the proposed signal processing pipeline for radar mounting angle estimation. An overview of the proposed pipeline will first be provided. Then the design details of each processing component will be explained.

5.3.1. Overview of Proposed Pipeline

The proposed processing pipeline belongs to the category of odometry-based methods and uses an inertial measurement unit (IMU) as the reference sensor. The objective is to estimate the mounting (yaw) angle of an automotive radar system installed on a moving vehicle. The formulation is based on rigid-body kinematics and assumes normal vehicle motion without lateral side-slip, which is a standard assumption in radar odometry literature [13, 15]. As illustrated in Figure 5.2, the pipeline takes radar point clouds and IMU yaw-rate measurements as inputs. The radar point cloud is processed by a neural network (NN)-based radar motion estimator, which outputs radar ego-motion estimates together with estimated point weights [57]. The estimated weights are then used for variance estimation and rejection of sparse radar frames.

On the IMU side, only yaw-rate measurements are used. A yaw-rate measurement model [138] is applied to account for the IMU bias and scale factor. The yaw-rate readings are then

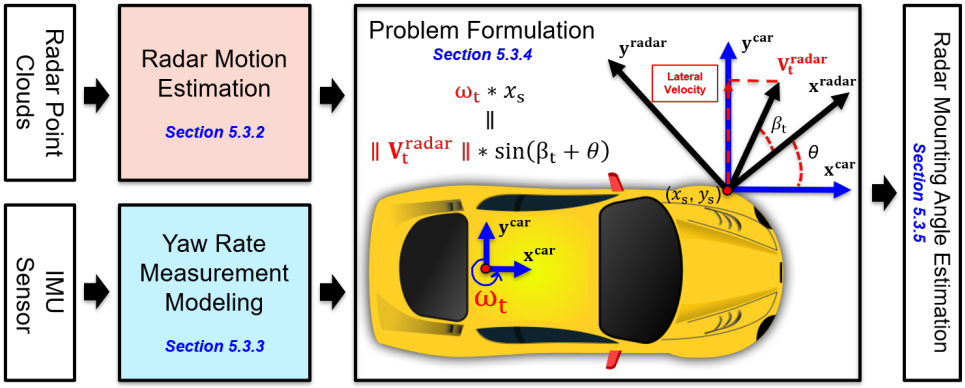


Figure 5.2: Overview of the proposed signal processing pipeline for the problem of radar mounting angle estimation. Radar point clouds and IMU yaw-rate measurements are used as inputs, and the radar mounting angle is estimated as the output. Radar motion is first estimated from the point clouds, while IMU yaw-rate measurements are modeled to account for bias and scale factor effects. The mounting angle is then obtained by enforcing a lateral velocity equality constraint within a kinematic formulation and solving a weighted least-squares problem. Here, $\mathbf{v}_t^{\text{radar}}$ denotes the radar motion vector at timestamp t, ω_t is the vehicle yaw rate (rotational velocity), (x_s, y_s) are the radar coordinates with respect to the vehicle rear center, β_t is the direction of radar motion in the radar coordinate frame, and θ is the unknown radar mounting angle.

de-biased before being used for mounting angle estimation. In the problem formulation, the proposed method exploits the velocity equality in the radar's lateral direction. Specifically, the projection of the estimated radar motion in the lateral direction must equal the lateral velocity induced by vehicle rotation, under the no–side-slip assumption. In the final stage, radar lateral-velocity equations from multiple radar frames are combined to form an overdetermined system. The radar mounting angle and the IMU scale factor are then jointly estimated using the weighted least-squares (w-LSQ) method.

It is important to highlight that the overall structure of the proposed pipeline is driven by the practical constraints encountered in real-world driving. Adverse weather and lighting conditions limit the reliability of optical sensors, motivating the use of an IMU as the reference sensor due to its robustness to environmental conditions, high refresh rate, and low data throughput. At the same time, radar point clouds captured in dense traffic are dominated by measurements from moving objects and are often sparse or affected by vehicle acceleration, which makes classical model-based approaches unreliable. A learning-based method is therefore employed to process radar data, handle these unfavorable conditions, and estimate radar motion. Finally, a kinematic formulation is adopted to relate radar motion and vehicle rotation without introducing vehicle-specific dynamic parameters. Together, these components form a compact pipeline that is required to estimate the radar mounting angle under operational driving conditions.

5.3.2. Radar Motion Estimation

The main objective of the radar motion estimator is to process raw radar point clouds and estimate the radar motion. In the literature, two main approaches exist for radar motion estimation: scan-matching methods [46] and instantaneous methods [13]. It has been shown in

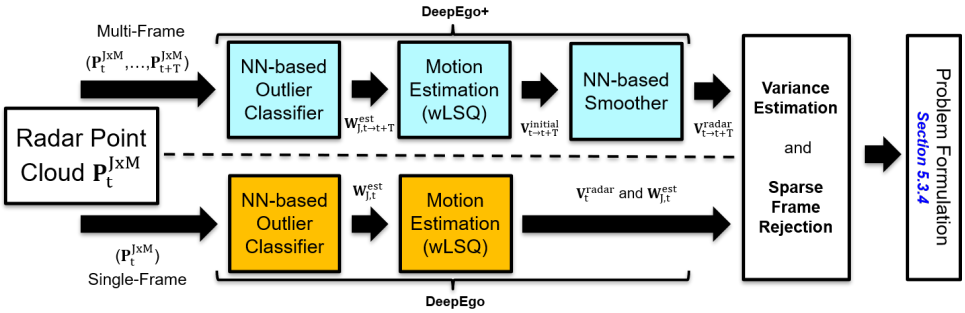


Figure 5.3: Proposed signal processing pipeline for radar motion estimation. The estimator takes radar point clouds as input and outputs motion estimates and point weights. The weights are subsequently used to compute motion variance and reject sparse radar frames. Instead of RANSAC, two neural network-based approaches are employed: *DeepEgo* [57] for single-frame input, and *DeepEgo+* [143], which incorporates temporal layers, for multi-frame input.

5

[57] that, under realistic driving scenarios, instantaneous methods [13, 57] are generally less sensitive to point cloud sparsity and dynamic objects than scan-matching methods. Furthermore, most approaches for radar mounting angle estimation rely on instantaneous methods [63, 131, 135, 138]. However, as discussed in Section 5.2, these methods typically depend on random sampling (e.g., RANSAC [61] or its variants) for outlier rejection, and their performance degrades significantly when the radar point cloud contains a high proportion of outliers. In addition, RANSAC is an iterative algorithm, which can lead to poor runtime performance.

To address the above issues, the proposed pipeline employs two NN-based approaches for two scenarios: (1) a single radar frame [57] and (2) multiple radar frames [143]. The bottom of Figure 5.3 illustrates the single-frame case. At timestamp t, the radar provides a point cloud that can be represented as a $J \times M$ matrix, consisting of J detections (rows), each with M features (e.g., range, Doppler, angle of arrival). The NN-based approach directly processes this structured input, extracts spatial sinusoidal features from the Doppler profile, and estimates a weight vector $\mathbf{w}_{J,t}^{\text{est}}$ for the J points. These weights are then used to eliminate outliers. The final radar motion $\mathbf{v}_t^{\text{radar}}$ is estimated using the w-LSQ method.

For the multi-frame case, motion estimates accumulated over T consecutive radar frames are processed by an additional temporal NN, denoted as the "NN-based Smoother" in Figure 5.3. This temporal NN captures the hidden relationship between non-zero vehicle acceleration and Doppler spectrum broadening,[1] while also smoothing the initial estimates according to a second-order motion model. The output of the temporal NN consists of T motion estimates $\mathbf{v}_{t\to t+T}^{\text{radar}}$ and a weight matrix $\mathbf{w}_{J,t\to t+T}^{\text{est}}$.

To further improve estimation reliability, the estimated radar motion and point weights are used to compute the estimation variance and to reject sparse radar frames. First, the number of inlier points L_t is calculated from the weights:

[1] Vehicle acceleration causes Doppler migration and spectrum broadening, such that the measured radial velocity of a static object no longer matches the instantaneous vehicle motion. The magnitude of the mismatch depends on the acceleration.

$$
\begin{aligned}
L_t &= \sum_{j=1}^{J} q_{j,t}, \\
q_{j,t} &= \begin{cases} 1, & w_{j,t}^{\text{est}} \geq IT, \\ 0, & w_{j,t}^{\text{est}} < IT, \end{cases}
\end{aligned}
\tag{5.1}
$$

where IT denotes the inlier threshold. Using the angle α_t^l and Doppler d_t^l measurements of the L_t inliers, together with the estimated 2D radar motion $\mathbf{V}_t^{\text{radar}} = \left[v_{x,t}^{\text{radar}},\ v_{y,t}^{\text{radar}}\right]^{\mathsf{T}}$, the residual error vector ϵ_t is computed as

$$
\begin{aligned}
\epsilon_t &= \mathbf{A}_t \cdot \mathbf{V}_t^{\text{radar}} - \mathbf{D}_t, \\
\mathbf{A}_t &= \begin{bmatrix} \cos(\alpha_t^1) & \sin(\alpha_t^1) \\ \vdots & \vdots \\ \cos(\alpha_t^{L_t}) & \sin(\alpha_t^{L_t}) \end{bmatrix}, \\
\mathbf{D}_t &= \begin{bmatrix} -d_t^1 \\ \vdots \\ -d_t^{L_t} \end{bmatrix}.
\end{aligned}
\tag{5.2}
$$

The Doppler measurement d_t^l is defined as negative when an object moves *towards* the radar. Unless a sparse frame is detected, the covariance matrix of the radar motion estimate is computed as in [52]:

$$
\text{Cov}(\mathbf{V}_t^{\text{radar}}) = \begin{cases} \dfrac{\epsilon_t^{\mathsf{T}} \epsilon_t}{L_t - 2} (\mathbf{A}_t^{\mathsf{T}} \mathbf{A}_t)^{-1}, & \dfrac{L_t}{J} \geq IRT, \\ \text{diag}(\infty, \infty), & \dfrac{L_t}{J} < IRT, \end{cases}
\tag{5.3}
$$

where IRT is a pre-determined threshold on the inlier ratio L_t/J. The diagonal terms of the covariance matrix, denoted Var_t^{xx} and Var_t^{yy}, represent the variance of the estimated radar motion. These are later used in the mounting angle estimation stage to mitigate the effect of erroneous radar motion estimates and sparse radar measurements.

5.3.3. Inertial Measurement Unit

The proposed method uses yaw-rate readings from an IMU. Compared with other odometry sensors such as GPS, IMUs provide more reliable measurements in urban areas and in the presence of tall buildings. Nevertheless, several factors limit the accuracy of IMU yaw-rate measurements. To account for these effects, this work adopts a standard yaw-rate measurement model [138] that includes a scale factor, a bias, and additive noise:

$$
\omega_t \ = \ s\,\hat{\omega}_t \ + \ b \ + \ \nu_t, \qquad \nu_t \sim \mathcal{N}\left(0, \sigma_\omega^2\right), \tag{5.4}
$$

where ω_t is the measured yaw rate at time t, $\hat{\omega}_t$ is the true yaw rate, s is the (multiplicative) scale factor, b is the constant bias, and ν_t is zero-mean Gaussian noise with variance

σ_ω^2. Ideally, s is constant over the measurement range, but it may vary with temperature, introducing systematic errors over time. Estimating s is therefore important for accurate mounting angle estimation. The bias b is modeled as constant (time-invariant) in this study. When the ego-vehicle is stationary (i.e., $\hat{\omega}_t = 0$), the measurement reduces to $\omega_t = b + v_t$. Over T timestamps, the bias can be estimated by sample averaging:

$$\bar{b} = \frac{1}{T}\sum_{t=1}^{T} \omega_t \tag{5.5}$$

5.3.4. Problem Formulation

As illustrated in Figure 5.2, the proposed method exploits the equality of lateral velocities at the radar position to estimate the mounting angle. Specifically,

$$\left\|\mathbf{v}_t^{\text{radar}}\right\| \cdot \sin\left(\beta_t + \theta\right) = \frac{\tilde{\omega}_t}{s} \cdot x_s, \tag{5.6}$$

where $\mathbf{v}_t^{\text{radar}}$ is the radar motion vector at time t (estimated by the NN-based motion estimator), β_t is its direction in the radar frame, and θ is the radar mounting angle to be estimated. On the right-hand side, $\tilde{\omega}_t \triangleq \omega_t - \bar{b}$ is the de-biased yaw rate, s is the unknown IMU scale factor, and x_s is the value of the radar's mounting location with respect to the x-axis of the origin of the vehicle. In practice, the radar mounting location (x_s, y_s) is either specified by the vehicle manufacturer or can be measured with millimeter accuracy during installation [15]. Solving Equation 5.6 for θ gives:

$$\theta = \arcsin\left(\frac{\tilde{\omega}_t \cdot x_s}{s \cdot \left\|\mathbf{v}_t^{\text{radar}}\right\|}\right) - \beta_t. \tag{5.7}$$

Assuming no scale factor error ($s = 1$), the mounting angle can be estimated over T timestamps by a simple unweighted average as in [138]:

$$\bar{\theta} = \frac{1}{T}\sum_{t=1}^{T}\left[\arcsin\left(\frac{\tilde{\omega}_t \cdot x_s}{\left\|\mathbf{v}_t^{\text{radar}}\right\|}\right) - \beta_t\right]. \tag{5.8}$$

When $s \neq 1$, both s and θ must be estimated jointly. Using the change of variables $s' \triangleq 1/s$, Equation 5.7 can be rearranged to

$$\beta_t(\theta, s') = \arcsin\left(s' \cdot \chi_t\right) - \theta, \tag{5.9}$$

where

$$\chi_t \triangleq \frac{\tilde{\omega}_t \cdot x_s}{\left\|\mathbf{v}_t^{\text{radar}}\right\|}. \tag{5.10}$$

Linearizing $\beta_t(\theta, s')$ at (θ_0, s_0') via first-order Taylor expansion yields

$$\begin{aligned}\beta_t(\theta, s') &\approx \beta_t(\theta_0, s'_0) + (\theta - \theta_0)\,\frac{\partial \beta_t}{\partial \theta}(\theta_0, s'_0) \\ &\quad + (s' - s'_0)\,\frac{\partial \beta_t}{\partial s'}(\theta_0, s'_0) \\ &= \arcsin\left(s'_0 \cdot \chi_t\right) - \theta + \frac{(s' - s'_0)\cdot \chi_t}{\sqrt{1 - (s'_0 \cdot \chi_t)^2}}.\end{aligned} \tag{5.11}$$

since $\frac{\partial \beta_t}{\partial \theta} = -1$ and $\frac{\partial \beta_t}{\partial s'} = \frac{\chi_t}{\sqrt{1 - (s' \cdot \chi_t)^2}}$.

5.3.5. Mounting Angle Estimation

From the linearized expression in Equation 5.11, stacking T radar frames yields an overdetermined linear system (typically $T \gg 2$ with full column rank) in the unknowns θ (mounting angle) and s' (inverse IMU scale factor). Setting the linearization point $s'_0 = 1$, the system can be written as

$$\begin{aligned}&\mathbf{Y} = \mathbf{U}\,\mathbf{X}, \\ &\mathbf{Y} = \begin{bmatrix} \beta_1 - \arcsin(\chi_1) + \dfrac{\chi_1}{\sqrt{1-\chi_1^2}} \\ \vdots \\ \beta_T - \arcsin(\chi_T) + \dfrac{\chi_T}{\sqrt{1-\chi_T^2}} \end{bmatrix}, \\ &\mathbf{U} = \begin{bmatrix} -1 & \dfrac{\chi_1}{\sqrt{1-\chi_1^2}} \\ \vdots & \vdots \\ -1 & \dfrac{\chi_T}{\sqrt{1-\chi_T^2}} \end{bmatrix}, \\ &\mathbf{X} = \begin{bmatrix} \theta \\ s' \end{bmatrix}.\end{aligned} \tag{5.12}$$

Based on the covariance of the radar motion estimates, the w-LSQ solution is

$$\begin{aligned}&\bar{\mathbf{X}} = \left(\mathbf{U}^\top \mathbf{Q}\,\mathbf{U}\right)^{-1} \mathbf{U}^\top \mathbf{Q}\,\mathbf{Y}, \\ &\mathbf{Q} = \operatorname{diag}(\eta_1, \ldots, \eta_T), \\ &\eta_t = \frac{1}{Var_t^{xx} + Var_t^{yy}}.\end{aligned} \tag{5.13}$$

where the matrix $\mathbf{Q}$ is a diagonal weight matrix that down-weights frames with high radar motion uncertainty. Finally, it is worth noting that the proposed method relies on ordinary vehicle motion to enable mounting angle estimation. In particular, non-zero vehicle

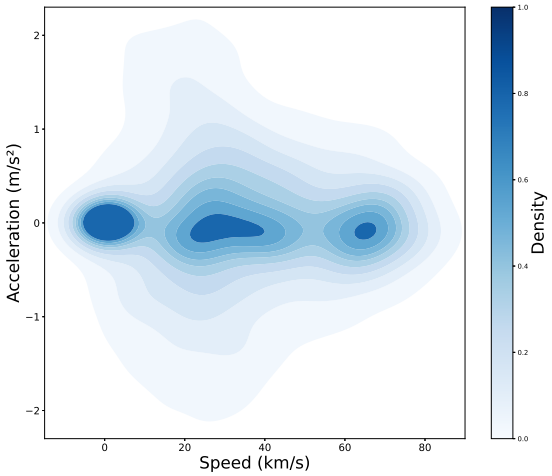


Figure 5.4: Density distribution of ego-vehicle speed and acceleration. For visualization, only 1% of the data (about 4.3k radar frames) was randomly sampled from the 64 selected recordings. This figure shows that the dataset used for evaluation covers a wide range of vehicle motion states during normal driving operations.

rotation is required so that the lateral velocity induced by vehicle yaw motion can be observed at the radar location. In addition, forward motion of the ego-vehicle is also necessary. The above conditions can be satisfied during normal driving operations, without the need for specially designed maneuvers or controlled motion patterns.

5.4. Results and Discussion

This section presents the evaluation results of the proposed pipeline for radar mounting angle estimation. A comprehensive comparison is made with related methods from the literature, as well as with alternative problem formulations and estimation techniques. Finally, the main challenges hindering the deployment of radar mounting angle estimation algorithms in realistic driving scenarios are identified and discussed.

5.4.1. Dataset and Evaluation Protocol

Unlike previous works, this study evaluates performance on the challenging real-world *RadarScenes* dataset [15]. The dataset contains 158 radar recordings from four automotive radars mounted on the front of the vehicle, covering a variety of times and driving scenarios. Since odometry-based methods are only applicable when the ego-vehicle is moving, we use 64 recordings (as in [57, 143]) as testing scenes. The selected recordings amount to more than 2 hours of data, corresponding to over 79 km of driving. Figure 5.4 shows the density distribution of ego-vehicle speed and acceleration in these scenes, indicating that the dataset covers both low/high speed driving and acceleration/braking conditions.

In addition to challenging data, this work also conducts a comprehensive performance comparison. First, the proposed method (both single-frame, SF, and multi-frame, MF) is compared with two representative methods from the literature (Table 5.1). This compari-

Table 5.1: Compared methods for radar mounting angle estimation. Depending on the neural network used, the proposed method operates either on a single frame (SF) or on multiple frames (MF). wMean denotes weighted mean; w-LSQ denotes weighted least squares. The two baseline methods are re-implemented following the descriptions in the original publications due to the lack of publicly available implementations.

Method	Formulation	Estimation Technique
Kellner et al. [138]	Lateral velocity	RANSAC [61] + wMean
Bao et al. [134]	Full velocity	Kabsch algorithm [144]
Proposed (SF)	Lateral velocity	*DeepEgo* [57] + w-LSQ
Proposed (MF)	Lateral velocity	*DeepEgo*+ [143] + w-LSQ

son considers not only estimation accuracy, but also estimation stability and convergence speed. Moreover, this work evaluates the relative trajectory error (RTE)[2] metric [57], which quantifies the impact of mounting angle misalignment on vehicle positioning and also highlights imperfections in the ground truth. Finally, this work explores alternative problem formulations and estimation techniques by modifying the proposed pipeline.

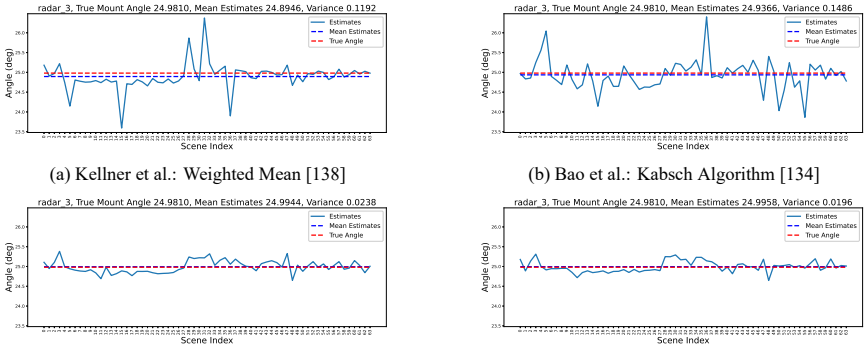
Ground truth mounting angles are obtained from the dataset documentation [15] (referred to as "True Angle"). To ensure generalization, the 64 test scenes are excluded from model training and validation. For the proposed MF method, eight consecutive frames are accumulated and smoothed by the temporal neural network. Following the evaluation scheme in [138], radar data are excluded when the vehicle translational velocity is below $1\,\mathrm{m/s}$ or the rotational velocity exceeds $140°/\mathrm{s}$.

5.4.2. Performance Across Diverse Driving Scenes

As discussed in Section 5.2, many previous studies evaluate radar mounting angle estimation in controlled environments (e.g., parking lots), where most surrounding objects are stationary. In contrast, real driving scenarios involve numerous moving vehicles, diverse road conditions, and a wide range of vehicle speeds and accelerations. It is therefore important to evaluate how mounting angle estimates behave across different driving scenes with varying levels of scene complexity. To this end, the estimation accuracy of all compared methods is evaluated across the 64 testing scenes. Figure 5.5 presents the estimated mounting angle for each scene. The two baseline methods from the literature show large fluctuations around the ground truth angle, and their performance is clearly affected by the driving environment. For example, both methods produce large errors in *Scene Index 36*, which contains many moving vehicles in front of and near the ego-vehicle. Even after averaging over all 64 scenes, the mean estimates of the baselines remain far from the ground truth, with the smallest mean error still around $0.0444°$.

In comparison, the proposed single-frame (SF) method shows substantially reduced variability across scenes, achieving an **80.0%** reduction in error variance and a **69.8%** improvement in mean accuracy compared with the best baseline. When multiple radar frames are available (Figure 5.5d), the proposed multi-frame (MF) method further exploits temporal

[2] RTE measures the difference between two trajectories. It first aligns the starting points of the estimated and reference trajectories, then computes the ℓ_2 distance between their end points (units in meters). For longer trajectories, the path can be divided into shorter segments (e.g., a 2 km trajectory divided into 50 m segments). RTE is then computed on each segment, and the final reported error (e.g., RTE_50) is the mean over all segments.



(a) Kellner et al.: Weighted Mean [138]

(b) Bao et al.: Kabsch Algorithm [134]

(c) Proposed (SF)

(d) Proposed (MF)

Figure 5.5: Radar mounting angle estimation across 64 test scenes for different approaches. Solid blue: estimated angle per scene. Dashed blue: mean of all estimates. Dashed red: ground truth mounting angle of Radar 3 from the *RadarScenes* dataset [15].

motion correlations to mitigate the effects of outliers and non-zero vehicle acceleration, resulting in more consistent estimates across scenes. It should be noted that all results in this section are based on data from a forward-looking radar ("Radar 3" in the dataset), which is particularly challenging because of the large number of moving objects in its FoV. Consequently, even with the proposed method, the estimates cannot be made fully consistent across all scenes, and the variance cannot be driven to zero. Preliminary experiments suggest that fusing data from multiple radars could alleviate this limitation, but a full exploration of multi-radar fusion is left for future work.

5.4.3. Estimation Accuracy

Unlike mounting location, the perception performance of automotive radar is highly sensitive to small misalignments of the mounting angle, since modern radar systems can detect objects at long range. Consequently, the proposed method must provide accurate angle estimates so that downstream radar-based tasks are not degraded. In addition, because modern vehicles are typically equipped with multiple radars mounted at different positions and orientations, the proposed method should perform consistently across sensors. Table 5.2 reports the estimation results over 64 test scenes and four radar sensors. At first glance, the mean and variance of the estimation error are much higher for Radar 2 and Radar 3 compared to Radar 1 and Radar 4. This is expected, since Radars 2 and 3 are forward-looking and therefore observe more dynamic objects. Nevertheless, both proposed methods outperform the two baselines from the literature.

The proposed single-frame approach achieves substantial accuracy improvements, while the multi-frame approach further reduces variance by smoothing motion estimates across frames. Interestingly, when compared with the dataset-provided 'True Angle', the SF and MF methods show similar mean accuracy, even though the MF method would theoretically be expected to perform better. A plausible explanation is that the "True Angle" values in the dataset [15] are quantized with limited resolution. Unfortunately, the actual resolution is not documented. Section 5.4.5 provides an alternative evaluation to indirectly assess how

Table 5.2: Performance comparison over 64 testing scenes and four radars. Mean and variance of the estimated mounting angle are reported. |ΔMin| is the absolute difference between the "True Angle" and the best mean estimate among the four methods. |ΔMax| is the absolute difference for the worst case. Blue highlights the best-performing values, while red highlights the worst.

Methods	Radar 1		Radar 2		Radar 3		Radar 4	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Weighted Mean [138]	**-85.1107°**	0.0051	-24.8973°	0.0305	**24.8946°**	0.1192	85.0030°	0.0059
Kabsch Algorithm [134]	-85.1102°	**0.0290**	**-24.7806°**	**0.8037**	24.9366°	**0.1486**	**84.9930°**	**0.0442**
Proposed (SF)	**-85.0418°**	0.0035	-25.0012°	0.0286	**24.9944°**	0.0238	**85.0256°**	0.0027
Proposed (MF)	-85.0467°	**0.0025**	**-24.9988°**	**0.0184**	24.9958°	**0.0196**	85.0286°	**0.0021**
True Angle	-85.0376°		-24.9916°		24.9810°		85.0269°	
|ΔMin| in Angle	**0.0042°**		**0.0072°**		**0.0134°**		**0.0013°**	
|ΔMax| in Angle	**0.0731°**		**0.2110°**		**0.0864°**		**0.0339°**	

close the proposed estimates are to the actual mounting angles.

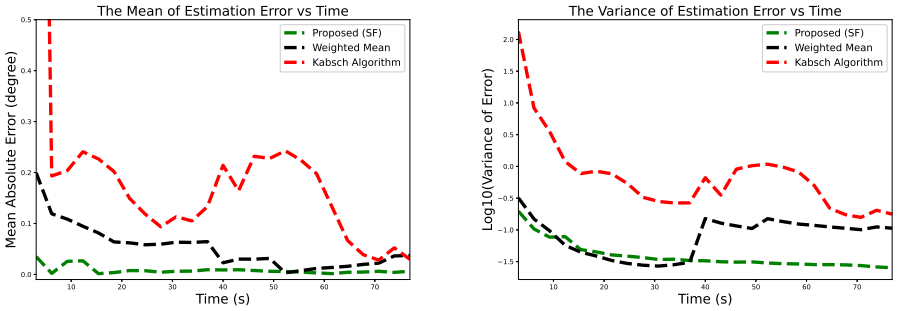
5.4.4. Convergence

The previous results demonstrated that the proposed method achieves accurate radar mounting angle estimation with low scene-to-scene variability. However, those results were either evaluated per scene (Figure 5.5) or averaged across all scenes (Table 5.2). For realistic driving scenarios, it is equally important to assess how quickly (in terms of seconds) the estimation converges to a stable value. To examine this, Figure 5.6 presents the mean absolute error (MAE) and error variance as functions of the time interval, i.e., the number of frames processed by each method. For MAE (Figure 5.6a), the proposed method achieves low estimation error within only a few seconds of driving, whereas the two reference methods require much more time before producing comparable accuracy. For error variance (Figure 5.6b), the proposed method consistently benefits from longer time intervals, yielding smaller variances. In contrast, the baseline methods show no clear convergence: their variance decreases initially, but then increases and fluctuates strongly with longer intervals.

Finally, it is worth noting that these results are based on the 64 test scenes from Radar 3 of the *RadarScenes* dataset, the forward-looking radar with many moving objects in its field of view. The results demonstrate that the proposed method can produce accurate mounting angle estimates with low variance within a short period of driving, even under challenging traffic conditions and with variations in velocities and trajectories of the ego-vehicle.

5.4.5. Trajectory Error

The vehicle trajectory can be reconstructed using recorded timestamps, estimated radar motion, and radar extrinsic parameters (i.e., mounting angle and position). Intuitively, the worse the mounting angle estimate, the greater the deviation of the reconstructed trajectory from the true trajectory. Thus, trajectory error provides an indirect measure of how close an estimated mounting angle is to the actual angle. This comparison can also include the dataset-provided mounting angle (previously referred to as the "True Angle"), although its resolution is undocumented. For quantitative evaluation, this section adopts the RTE metric, which measures the discrepancy between the estimated vehicle trajectory and the ground truth trajectory provided by the on-vehicle odometry system. Because both radar motion estimates and mounting angles influence trajectory reconstruction, in this experi-



(a) MAE with different time intervals

(b) Variance with different time intervals

Figure 5.6: Convergence behavior of different methods with respect to the number of radar frames. (a) Mean absolute error (MAE). (b) Error variance. The red dashed line denotes the Kabsch approach [134], the black dashed line denotes the weighted mean method [138], and the solid green line denotes the proposed method. Each test scene is divided into shorter segments of different lengths; mounting angles are estimated for each segment, and MAE and variance are then computed.

5

Table 5.3: Relative Trajectory Error (RTE) in meters [57], averaged over 64 testing scenes. RTE measures the discrepancy between the estimated and ground truth vehicle trajectories. The estimated trajectory is computed from timestamps, radar motions, and mounting angles. Radar motion is fixed (controlled), while mounting angle is varied (dependent). The baseline RTE is computed from the dataset-provided mounting angles [15]. Blue highlights the best-performing values, while red highlights the worst.

Sources of angle	Radar 1	Radar 2	Radar 3	Radar 4
Weighted Mean [138]	**11.98**	13.41	**12.03**	6.31
Kabsch Algorithm [134]	11.92	**26.98**	7.50	**6.81**
Proposed (SF)	7.30	5.96	6.45	**6.24**
Proposed (MF)	**7.28**	**5.95**	6.51	6.32
Baseline RTE	7.37	6.05	**5.99**	6.27

ment the estimated radar motion is fixed (controlled variable), while the radar mounting angle is varied (dependent variable).

Results are reported in Table 5.3. As expected, larger angle estimation errors lead to larger RTE values. For example, Radar 1 and Radar 3 with the weighted-mean method [138] exhibit higher angle errors in Table 5.2, which correspond to higher RTE values here. More importantly, the proposed methods consistently yield lower RTE than the baselines, demonstrating that improved angle estimates directly enhance trajectory accuracy. Interestingly, the proposed methods also achieve slightly lower RTE than the dataset-provided "True Angle". While this difference is small, it suggests that the proposed approach may in fact yield more accurate angles than those documented in the dataset. Nonetheless, as emphasized in [138], determining mounting angles with high precision is inherently difficult. The RTE metric therefore provides a useful indirect validation of the proposed method's accuracy.

Table 5.4: Performance comparison of the proposed method (MF version) with different problem formulations and estimation techniques. For each radar, results are averaged over 64 testing scenes. |ΔMin| is the absolute difference between the "True Angle" and the best estimate out of the compared methods. Blue highlights the best-performing values.

Methods	Radar 1	Radar 2	Radar 3	Radar 4
Weighted LSQ	**-85.0467°**	**-24.9988°**	**24.9958°**	85.0286°
Weighted Kabsch	-85.0493°	-25.0027°	25.0071°	85.0263°
Weighted Mean	-85.0476°	-25.0005°	25.0001°	**85.0269°**
Weighted ODR	-85.0469°	-25.0107°	25.0043°	85.0285°
True Angle	-85.0376°	-24.9916°	24.9810°	85.0269°
|ΔMin|	**0.0091°**	**0.0072°**	**0.0148°**	**0.0000°**

5.4.6. Further Exploration

As detailed in Section 5.3, the proposed method estimates the radar mounting angle by enforcing the equality of lateral velocities at the radar position. The formulation also incorporates the IMU measurement model, enabling joint estimation of the IMU scale factor and the mounting angle. However, several alternative approaches exist. For example, if both an IMU sensor and a DGPS system are available, then a full-velocity model can be constructed [134] and the mounting angle estimated using the Kabsch algorithm. If the IMU scale factor is close to 1, the IMU model can be ignored and the problem simplified to an averaging scheme [138]. Moreover, the orthogonal distance regression (ODR) can be applied to extend least squares (LSQ) to cases where errors also exist in the independent variables.

To better understand the performance trade-offs, the proposed method was modified accordingly while still using the frame weights from Equation (5.13). Results are shown in Table 5.4. Compared with Table 5.2, the performance gap between different formulations is now much smaller. Since the main difference between these methods and the baselines from the literature lies in the radar motion and frame weighting, this suggests that a major limiting factor in practical mounting angle estimation is the high proportion of outliers and sparse radar measurements. For the chosen dataset, the weighted mean solution appears most practical: it uses the simplest model, runs the fastest, and provides performance comparable to more complex techniques. If the IMU scale factor deviates significantly from 1, the weighted LSQ should instead be used. Finally, if the radar has limited resolution in azimuth and radial velocity, weighted ODR may yield better results.

5.5. Conclusion

This chapter presented a novel signal processing pipeline to address the problem of estimating radar mounting angles under operational driving conditions. Accurate external calibration of automotive radars, and in particular their mounting angles, is crucial for the safe operation of autonomous vehicles. To address this problem, an odometry-based approach was proposed that combines a neural network-based radar motion estimator with an IMU measurement model for bias and scale factor compensation. The mounting angle and IMU scale factor are then jointly estimated using a w-LSQ formulation based on a Taylor-series linearization. The proposed pipeline was validated on the challenging *RadarScenes* dataset, which includes diverse traffic scenes as well as velocity and trajectory variations of the

ego-vehicle.

Experimental results demonstrate that the method achieves accurate mounting angle estimates with low variability across diverse and realistic driving scenarios, while avoiding the need for controlled environments, specially designed radar targets, or tailored driving routes. In addition, the estimation converges within a short period of driving time. Although the formulation involves a single linearization step, experimental results show that this is sufficient in practice, requiring no further iteration. For future work, extending the framework to sensor fusion represents a promising direction. Combining multiple radar sensors or integrating complementary modalities (e.g., cameras or lidars) may further mitigate the impact of dynamic objects in the field of view and enhance calibration accuracy under complex real-world conditions.

6

Toward Holistic Radar Perception: Simultaneous Segmentation and Odometry

With recent advances in radar technology and AI, the literature has demonstrated that automotive radars can support a broad range of perception tasks, from classification and object tracking to the tasks explored in the preceding chapters. This diversity shows the potential of radar sensing. At the same time, it is important to critically examine not only how accurate and robust these applications are, but also the computational price paid for overcoming radar's inherent limitations. Many approaches rely on highly complex neural network backbones, which can make deployment impractical in real-world automotive systems. This raises a central question: what is the most suitable, reliable, and affordable role for automotive radars today, given their current constraints? From the perspective of driving safety, criteria such as robustness, consistency, and accuracy are essential. In response, this chapter proposes a radar-only framework that performs segmentation and odometry simultaneously. The solution is designed to be fast, accurate, and reliable, while addressing a key gap in the radar perception chain. Nearly all downstream radar applications can take advantage of the output of the proposed method. By focusing on this foundation, the chapter defines a realistic and impactful role for radar in automotive perception today.

Parts of this chapter have been published in:
S. Zhu, S. Ravindran, A. Yarovoy, and F. Fioranelli, "Redefining Radar Segmentation: Simultaneous Static-Moving Segmentation and Ego-Motion Estimation using Radar Point Clouds," in IEEE Transactions on Radar Systems. (under review)

6.1. Introduction

Among various perception tasks, radar-based segmentation has gained significant attention in recent years. The main objective is to assign a class label to each point in the radar point cloud [145] or each cell in the radar data cube [146]. The radar point cloud is generated by applying a detector, such as one of the Constant False Alarm Rate (CFAR) algorithms [147], to the radar data cube. Both data formats capture information such as range, radial velocity, and angle of arrival (AoA) of objects in the scanned environment. Performing segmentation on these data is therefore crucial for scene understanding and driving safety. In the radar literature, three categories of segmentation tasks have been explored: semantic segmentation [145, 146, 148–153], instance segmentation [154–156], and panoptic segmentation [157]. These studies have demonstrated the great potential of radar sensors and established a solid foundation for future perception systems. However, most segmentation works have focused only on moving objects, while radar point clouds typically contain detections from moving objects (e.g., cars), static objects (e.g., buildings), and false positives (e.g., unidentified objects or multipath reflections). A broader review shows that, in other radar-based perception tasks, identifying static objects is also important [52, 63, 81, 158, 159]. To locate static detections, some studies assume a static environment [140, 160], while others rely on vehicle ego-motion [63, 159] or random sampling techniques [52, 81] such as the Random Sample Consensus (RANSAC) algorithm [61]. While these approaches help identify static objects, they either require external odometry sensors or assume that most objects are static, and often leave moving objects mixed with false positives, requiring further separation.

Therefore, to bridge the gap between unprocessed radar point clouds and perception applications that require knowledge of either static/moving object positions or vehicle speed, this study redefines the objective of conventional radar segmentation tasks. Specifically, it proposes a unified solution that can simultaneously perform the dual tasks of static–moving object segmentation and vehicle ego-motion estimation, as illustrated in Figure 6.1. To the best of our knowledge, this is the first attempt to enable such a dual task, and the results demonstrate that raw radar point clouds contain sufficient information to achieve both. In addition to this primary contribution, the proposed method offers the following advancements:

1. **Radar-only:** Unlike many other studies, the proposed method performs both tasks using only radar data. For example, it eliminates the need for odometry sensors to measure vehicle ego-motion for radial velocity or motion compensation. This ensures sensor independence and avoids errors introduced by synchronization issues or output glitches.

2. **No Aggregation:** The proposed method handles sparse radar point clouds without requiring aggregation across multiple radars or frames. To capture temporal features, it uses a moving window and processes multiple radar point clouds as input. This preserves temporal information and avoids reliance on direct coordinate transformations or motion-compensated aggregation[1], making the approach robust in highly dynamic scenes.

[1] Radar point cloud aggregation with motion compensation involves transferring several point clouds to a reference cloud and compensating their positions based on vehicle ego-motion. In other words, motion compensation requires knowledge of ego-motion, while direct aggregation does not.

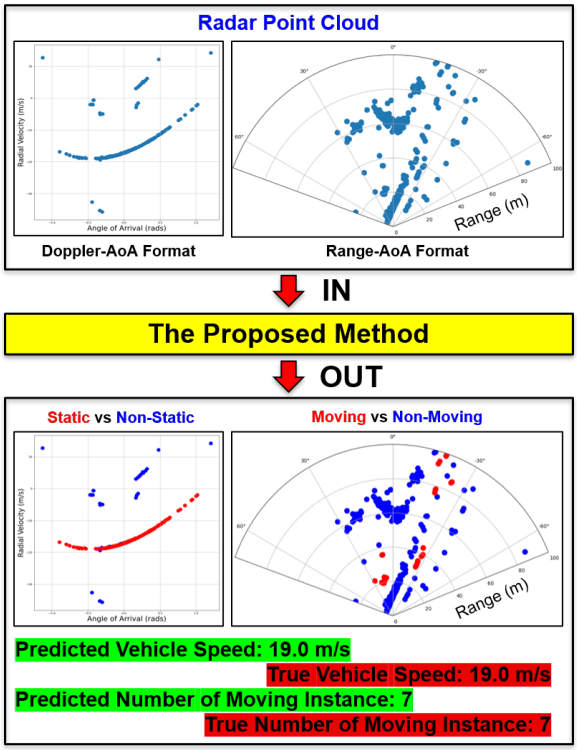


Figure 6.1: The proposed method takes multidimensional radar point clouds as input, uses neural networks (NNs) for automatic feature extraction, and then segments static and moving objects. Based on the measured radial velocity of static objects, the method can estimate the ego-motion of the moving vehicle. The distinct moving instances can also be generated after applying a clustering algorithm to the predicted moving objects. In this example, the RadarScenes [15] dataset is used for testing.

3. **Lightweight:** The proposed method employs simple yet effective neural network backbones for feature extraction, using a multi-layer perceptron (MLP) for spatial features and a recurrent neural network (RNN) for temporal features. The resulting model is lightweight (0.15M parameters) while still providing critical information for understanding vehicle motion and supporting downstream perception tasks.

4. **Dataset:** Since no existing radar dataset fully supports the proposed objective, this chapter reorganizes the ground-truth (GT) class labels of the RadarScenes dataset [15]. Specifically, vehicle ego-motion was used to separate static from non-static objects, the output of the *DeepEgo+* approach was incorporated to compensate for the effects of vehicle acceleration [143], which can otherwise cause mislabeling of static objects, and moving versus non-moving objects were subsequently classified using the dataset's original labels.

Finally, it should be noted that the goal of this study differs from previous radar segmentation work. While earlier studies have focused on assigning detailed class labels, which is

undoubtedly important and meaningful, this study began as a continuation of traditional segmentation but ultimately addressed a missing link in the radar perception chain. Therefore, it is not appropriate to compare this chapter directly with research aimed solely at assigning class labels to pixels or points. Instead, the proposed method should be regarded as complementary to traditional radar segmentation and to other radar perception tasks. The rest of this chapter is organized as follows. Section 6.2 reviews existing research on this topic. Section 6.3 presents the detailed design of the proposed method. Section 6.4 reports performance results. Finally, Section 6.5 draws conclusions and outlines future research directions.

6.2. Related Work

This section reviews the relevant literature. It first outlines prior work on radar-based segmentation and recent advances in the field. It then examines studies on other radar perception tasks to highlight the importance of performing the proposed dual tasks on radar point clouds. Finally, a brief summary of the literature review is provided.

6.2.1. Radar-based Segmentation

According to their objectives, previous radar-based segmentation studies can be divided into three categories: semantic segmentation [145, 146, 148–153], which assigns a class label to each radar point (detection); instance segmentation [154–156], which not only classifies each point but also distinguishes between individual objects within the same class; and panoptic segmentation [157], which combines both approaches by providing semantic labels for all points while also separating instances for object classes. In addition, previous studies can also be divided according to the format of radar data, where except for [146, 152, 153], which use radar cubes (before detection), all of the rest use radar point clouds (after detection). Methods using radar cubes claim they are superior in segmenting small objects, as information can be lost during the detection process [146, 152]. Nevertheless, there are currently no conclusive experimental comparisons demonstrating their effectiveness. For methods that rely on radar point clouds, the RadarScenes dataset [15] appears to be a popular choice since all methods use it to evaluate their performance. Although the RadarScenes dataset provides 10 different classes for moving objects, almost all studies use less than half of them, reflecting the challenges of performing detailed semantic segmentation using sparse and noisy radar data. To handle this challenge, PointNet++ [145, 148, 154, 155] and Transformer [149, 151, 156, 157] have become the most commonly used feature extraction backbones in these studies. Theoretically, Transformer outperforms PointNet++ in handling sparsity and long-range dependencies; experimentally, Transformer also demonstrates better performance than PointNet++ [149, 151, 157].

Based on this brief literature review, it is evident that many studies have extensively explored the topic of radar-based segmentation from various perspectives, such as objectives, data formats, and feature extraction backbones. It is a solid start, especially in such a pioneering field as automotive radar perception. However, there are still areas for further improvement. Firstly, except for studies using stationary ego-vehicle datasets [146, 153], nearly all prior research requires knowledge of vehicle ego-motion provided by odometry sensors. Ego-motion is often used to compensate for the measured radial velocity, which is

then used as an important input object feature. However, if ego-motion is known, segmenting the static background becomes straightforward, as was done in [148]. Furthermore, since almost 97% of radar detections come from static objects [145], the computational complexity of these methods can be significantly reduced by removing static points from the input. Also, with the compensated radial velocity, it is understandable that most studies achieve scores exceeding 99% on the Intersection over Union (IoU) metric for classifying 'static'[2] objects. Last but not least, relying on external sensors may compromise sensor independence and system robustness due to potential erroneous outputs or synchronization issues.

Secondly, to address the sparsity problem, some studies [145, 148, 150, 151] rely on combining radar point clouds over a fixed time period (e.g., 500 ms), regardless of the number of clouds aggregated. However, this approach can adversely increase the inference latency and system memory consumption [157]. In contrast, other studies [149, 156, 157, 161] also merge clouds, but they only allow each radar to contribute once per fused cloud. Given a 60 ms update rate per radar, this can shorten aggregation time while still benefiting from the increased cloud density due to overlapping fields of view (FoVs). Nevertheless, all the above solutions still introduce some degree of inference latency. Moreover, without motion compensation, they may experience performance degradation in highly dynamic scenes, especially when moving objects are present in the overlap region. Furthermore, since the radars in the RadarScenes dataset are fully unsynchronized[3], fusing radar point clouds cannot be done directly but requires a heuristic process [162].

Thirdly, to handle the challenging task of labeling objects in sparse and noisy radar point clouds, previous studies usually adopt NNs with sophisticated feature extraction backbones, such as Transformer and PointNet++. While Transformer outperforms PointNet++, it is typically too bulky to be suitable for radar processing systems that require real-time prediction and immediate feedback [154]. In addition, these backbones are often described as 'data-hungry', but large radar datasets are expensive to generate and annotate. In any case, due to the fundamental limitations of radar sensors, the performance gains from using complex backbones are not as significant as with optical sensors [163, 164], leading one to wonder: why not use radar for tasks that are better suited to its characteristics? For example, recent studies [161, 165] no longer search for specific object types or bounding boxes, but instead focus on the simpler task of class-agnostic segmentation and tracking.

Last but not least, it is worth noting that most previous studies have only focused on segmenting moving objects from radar point clouds, labeling static objects and false positives together as 'static'. From the perspective of various radar perception tasks, it is important to conduct a comprehensive segmentation; the reasons for which will be further explained in the next section.

6.2.2. Other Radar-based Tasks

A radar point cloud typically contains a mix of detections from moving objects (e.g., vehicles), static objects (e.g., buildings), and false positives (e.g., false detections from side-

[2] In the RadarScenes dataset, radar detections from static objects and false positives are both labeled as 'static', whereas in this study, they are treated separately.

[3] The time intervals between individual radar outputs are not uniform, and radar transmit and receive operations are not ordered.

lobes). Most existing radar-based segmentation tasks focus on separating moving objects, leaving static objects mixed with false positives. However, static objects also play a vital role in many radar perception tasks. For example, the measured radial velocity of static objects can be used to estimate the vehicle ego-motion [13, 52] and thus calibrate the radar extrinsic parameters [63, 134]. Additionally, knowing where static objects are located allows for the implementation of algorithms such as semantic grid mapping [148], simultaneous localization and mapping (SLAM) [81, 166], and amplitude and phase calibration [167]. Furthermore, separating static points from the radar point cloud can help perform free space detection [158, 168], road course estimation [159, 169], and multi-object tracking [170]. Among these studies, most rely on knowing the vehicle's ego-motion provided by external sensors to localize static objects; some use neural networks [57, 143]; some assume a majority of static points and employ one additional processing step such as RANSAC [13] or M-Estimator Sample Consensus (MSAC) [81]. However, while these solutions can help localize static objects, they leave moving objects mixed with false positives.

6.2.3. Summary

In summary, it is essential to point out that in the current radar perception processing chain, there is a missing component that can not only explicitly but also simultaneously segment static objects, moving objects, and false positives from the radar point cloud, which, according to the literature review, is considered crucial for various downstream applications. Furthermore, as the first processing unit after CFAR detectors, this component should be able to work independently, extract important segmentation features automatically from sparse and noisy radar point clouds, and provide fast, accurate, and reliable predictions. The realization of this component summarizes the goals of this research, which will be further described in the next section.

6.3. Proposed Method

Figure 6.2 presents the architecture of the proposed method for simultaneous static-moving object segmentation and vehicle ego-motion estimation. The proposed method takes unprocessed radar point clouds as input, which will be detailed in Section 6.3.1; performs automatic spatiotemporal feature extraction, as explained in Section 6.3.2; predicts static and moving objects and estimates ego-motion in Section 6.3.3; and finally outputs detailed object type labels and moving object instances after several processing steps as detailed in Section 6.3.4. Implementation details will be presented in Section 6.3.5.

6.3.1. Network Input Analysis

The proposed method takes T unprocessed and chronologically ordered radar point clouds as input. Typically, the radar point cloud is generated after the application of CFAR algorithms. Each point cloud is assumed to have J radar detection points, and each detection point contains M object features. In this chapter, M is assumed to be greater than or equal to 3, thus containing at least the uncompensated radial velocity, range, and angle of arrival (AoA) information of the detected objects. The reason for including at least the three selected object features and having T consecutive radar clouds is that they contain the necessary spatial and temporal features for the network to distinguish between moving and static

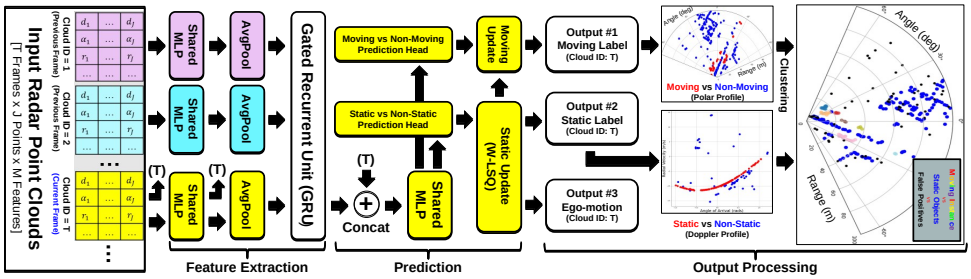


Figure 6.2: Architecture of the proposed neural network for simultaneous static-moving object segmentation and vehicle ego-motion estimation. The network takes multidimensional radar point clouds as input, performs automatic spatiotemporal feature extraction, predicts static labels and moving labels for each detection point, and implements weighted least squares (w-LSQ) for ego-motion estimation. As an illustrative application, moving instances can be generated after applying a clustering algorithm to the grouped moving objects. d_j denotes radial velocity, α_j denotes AoA, r_j denotes range, and j is the detection point index.

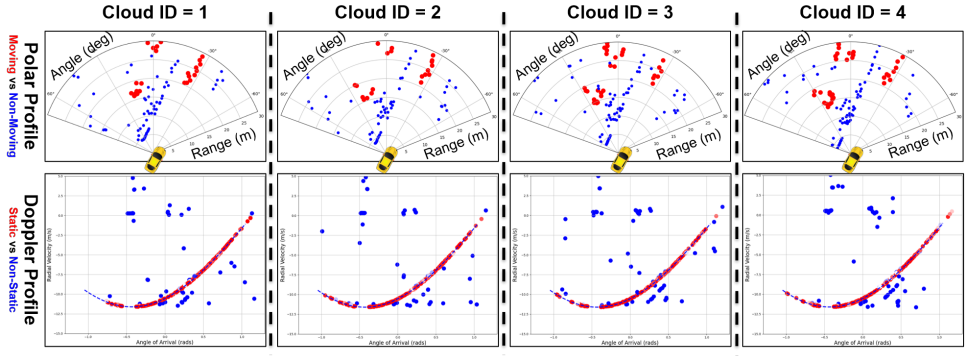


Figure 6.3: An illustration of how moving and static objects appear in radar point clouds across multiple consecutive frames. The first row shows the polar profile, which presents the radar point cloud in the Range-AoA domain. The moving objects, marked in red, exhibit clear spatial concentration and temporal correlation in the polar profile. The second row shows the Doppler profile, which presents the radar point cloud in the radial velocity-AoA domain. The static objects, marked in red, exhibit a distinct sinusoid-like spatial pattern with little temporal variation. In this example, the RadarScenes dataset [15] is used.

objects, which can also be visually seen in Figure 6.3.

In the Doppler profile, not only is there a clear spatial distinction between static and non-static objects, but there is also a strong temporal correlation between consecutive point clouds of the static detections. In this dissertation, static objects refer to detection points whose measured radial velocity is solely determined by the measurement angle and ego-radar motion, thereby forming a characteristic sinusoid-like pattern in the Doppler profile. In contrast, non-static objects are detection points that deviate from this pattern due to additional velocity contributions, such as independent target motion or false positives. The temporal correlation of static detections is dominated by the continuous motion of the ego-vehicle; consequently, the sinusoid-like pattern remains stable over time, enabling estimation of the radar/vehicle motion from the measured features of static objects (see e.g., [13]).

In the polar profile, there are also spatial and temporal correlations between objects in consecutive point clouds. However, because the ego-vehicle is moving and there is no motion compensation, all objects appear to 'move' across frames. Furthermore, due to the nature of radar data, the shape and density of detected objects may vary between frames, making reliable discrimination of static objects more challenging. In contrast, detection points from moving objects are usually more spatially concentrated in the polar profile than in the Doppler profile, especially when they are near the radar. This is because, in the Doppler profile, the measured radial velocity at different points on a moving object can vary greatly depending on the measurement angle. Thus, once static objects are first separated in the Doppler profile, the polar profile can help refine the identification of moving objects. In this study, moving objects are defined as detection points originating from targets physically in motion at the time of measurement, whereas non-moving objects comprise static detections, false positives, and inherently mobile targets that are currently stationary (e.g., parked or waiting vehicles).

In summary, unlike previous studies, the proposed method does not require point cloud aggregation, knowledge of the vehicle's ego-motion, or compensation for radial velocity or ego-motion. In contrast, the authors believe that using T consecutive raw radar point clouds is sufficient to simultaneously distinguish between static and moving objects and estimate the vehicle's ego-motion. Regarding the latency issue, for real-time applications, the requirement of T radar frames can be formulated as a moving window so that the proposed method can provide instantaneous predictions. Lastly, the remaining issue is how to effectively extract relevant features for segmentation, which will be detailed in the next section.

6.3.2. Feature Extraction

As a result of clear spatial distinctions and strong temporal correlations in the input radar point clouds, the proposed method is able to perform effective feature extraction with simple neural network backbones. For spatial feature extraction, this study employs the PointNet architecture [77], which consists of a shared multi-layer perceptron (MLP) followed by average pooling. Specifically, the MLP is applied independently to each radar detection point in each input point cloud. Afterwards, the pooling layer is used to aggregate a global feature vector for each input point cloud. Despite its simple architecture, the combination of MLP and average pooling has demonstrated effectiveness in extracting the sinusoid-like spatial feature in the the Doppler profile for static object segmentation and vehicle ego-motion estimation [57, 143]. Furthermore, because the MLP is shared across input point clouds, the network complexity does not increase with the number of input point clouds. However, it must be acknowledged that this combination has limited ability to capture relationships between neighboring detection points and may therefore be insufficient for tasks requiring fine-grained spatial understanding. Nevertheless, given the sparse radar point clouds, it remains to be seen how much performance improvement more advanced feature extraction backbones (with local details) can bring, as a previous exploration has shown only modest gains [171].

For temporal feature extraction, the proposed method uses the gated recurrent unit (GRU). The GRU is a type of recurrent neural network (RNN) that can extract long-term dependencies in sequential data. In this study, the global feature vectors generated by the previous

pooling layer are first arranged in chronological order. The GRU then processes these feature vectors sequentially, capturing the hidden relationships within them and outputting a feature vector that contains both spatial and temporal information. It is important to mention that the temporal dependencies between radar point clouds are governed by the continuous motion of the ego-vehicle and moving objects. However, since the input data have no radial velocity compensation or motion compensation, the authors hypothesize that temporal feature extraction is more beneficial for the segmentation of moving objects, while static objects already provide strong differentiation in spatial features, and temporal features are only supplementary.

6.3.3. Prediction

The previous section extracts spatial features from the input radar point cloud and captures the temporal dependencies caused by continuous object motion. This section explains how to make predictions for each radar detection point. Firstly, the spatiotemporal feature vector generated by the GRU is replicated to the original input point cloud and the outputs of different layers in the first shared-MLP through feature concatenation. The concatenation outputs a 2D matrix that still contains J points in one dimension, but in the other dimension it contains more global and spatial details in addition to the original M input features. Then, another shared-MLP acts as a decoder, refining the fused features and producing a rich feature vector (per-point) that is based on both spatiotemporal context and local per-point features. After that, the decoder output is sent to two prediction heads, one for static and non-static prediction (static head) and the other for moving and non-moving prediction (moving head). Each head consists of three 1D convolutional layers with the last layer having a sigmoid activation function. The static head outputs a $J \times 1$ vector, where each element contains a value from 0 to 1, indicating the probability of being non-static (0) or static (1). The moving head functions similarly, with its elements representing the probability of a detection point being non-moving (0) or moving (1).

Up to this point, the output of the static head is sufficient for the task of ego-motion estimation. However, since one of the goals is to localize all static objects, the chosen feature extraction backbone has limited ability to capture local context, which is the price of a lightweight network. Consequently, some static objects may be misclassified as non-static and assigned lower weights in the static head, or misclassified as moving and assigned higher weights in the motion head. To address this issue, the proposed method employs two update heads: one for the static weight update and the other for the moving weight update. The initial prediction of the static weight is updated first, based on the fact that knowing the radar motion helps to localize all static objects. Therefore, in the static update head, initial static weights are used to first compute the radar motion via the weighted least squares (w-LSQ) method. Then the estimated radar motion is used to update the static weights for all detection points, as formulated below:

$$\mathbf{D} = \begin{bmatrix} -d_1 \\ \dots \\ -d_J \end{bmatrix}, \ \mathbf{A} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ \dots & \dots \\ \cos(\alpha_J) & \sin(\alpha_J) \end{bmatrix} \tag{6.1}$$

$$\mathbf{V}^{est} = (\mathbf{A}^T \mathbf{W}_{static}^{ini} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}_{static}^{ini} \mathbf{D} \tag{6.2}$$

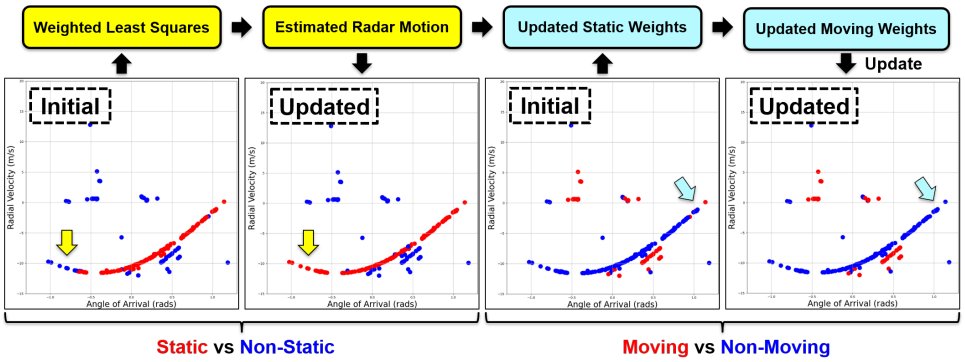


Figure 6.4: An illustration of how the initial weights for static and moving objects are updated in the two weight update heads. The yellow blocks represent the static update head, and the cyan blocks represent the moving update head. In this example, the RadarScenes dataset [15] is used, and the plots show the radar point cloud in the radial velocity-AoA domain.

$$\mathbf{W}_{static}^{ini} = \mathrm{diag}\left(w_{static}^{ini,1}, \dots, w_{static}^{ini,J}\right) \in \mathbb{R}^{J\times J}, \qquad \mathbf{V}^{est} = \begin{bmatrix} v_x^{rad} \\ v_y^{rad} \end{bmatrix} \tag{6.3}$$

$$[e_1, \dots, e_J]^T = \mathbf{A}\mathbf{V}^{est} - \mathbf{D}, \qquad w_{static}^{est,\,j} = \exp\left(-\frac{e_j^2}{2\sigma^2}\right) \tag{6.4}$$

Where d_j is the measured radial velocity, α_j is the AoA measurement, σ is the standard deviation of the assumed Gaussian error distribution in the radial velocity measurement, $\mathbf{W}_{static}^{ini}$ is the diagonal matrix that contains the predicted initial static weights, $w_{static}^{est,\,j}$ is the updated static weight of the j-th point, and $\mathbf{V}^{est}$ is the estimated radar velocity on its x- and y-axis. As for updating the moving weights, the method enforces a consistency constraint that a detection point should not be simultaneously assigned high weights (or confidence) as both static and moving. Therefore, the updated static weights $w_{static}^{est,\,j}$ are used to refine the initial moving weights $w_{mov}^{ini,\,j}$, as shown below:

$$w_{mov}^{est,\,j} = \begin{cases} w_{mov}^{ini,\,j} & w_{static}^{est,\,j} \le c_{static} \\ 0 & w_{static}^{est,\,j} > c_{static} \end{cases} \quad and \quad c_{static} \in (0,\,1) \tag{6.5}$$

where c_{static} is an empirically chosen confidence threshold and $w_{mov}^{est,\,j}$ is the updated moving weight for detection point j. Figure 6.4 presents a visual illustration of the update process of the static weights and the moving weights.

Finally, it is important to clarify that although a single 3-class prediction head (moving–static–false positives) is possible, the problem exhibits a hierarchical structure, as shown previously. In this hierarchy, the initial static prediction provides the basis for estimating radar motion, which in turn updates the initial static prediction and cross-checks subsequent moving predictions. Using two prediction heads allows the architecture to explicitly encode this structure, enabling the network to solve simpler binary classification tasks rather than

implicitly learning the full set of relationships. Moreover, the two-head approach avoids the need to directly model false positives, which do not form a semantically consistent class, making direct modeling less structured than static or moving objects.

6.3.4. Output Processing

For the current radar input point cloud, the proposed method can simultaneously estimate the vehicle ego-motion and provide labels for static and moving objects. Given the radar extrinsic parameters and the estimated radar motion, the vehicle motion can be computed as follows:

$$\omega = \frac{v_y^{rad}\cos(\theta) + v_x^{rad}\sin(\theta)}{x}$$
$$v_x^{car} = v_x^{rad}\cos(\theta) - v_y^{rad}\sin(\theta) + \omega \cdot y \tag{6.6}$$

where v_x^{car} is the vehicle's translational speed, ω is its rotation rate, and the vehicle is assumed to have no lateral speed, i.e., $v_y^{car} = 0$. x and y are the mounting position[4] and θ is the radar mounting angle, with respect to the rear center of the vehicle. The labels for static and moving objects can be obtained directly by applying thresholds to the updated static and moving weights respectively. In this study, both thresholds are set empirically to 0.1. As shown in the rightmost sub-figure of Figure 6.2, the static and moving labels can be merged together to achieve a clear separation of false positives. Furthermore, since moving objects are explicitly separated, clustering algorithms such as DBSCAN can be applied to them to achieve moving instance segmentation. However, the instance segmentation is just one illustrative example, and as discussed in Section 6.2, many radar perception tasks can be connected to the output of the proposed method.

6.3.5. Implementation Details

The proposed method is trained on one NVIDIA A100 GPU provided by the Delft High Performance Computing Centre (DHPC) [87]. The batch size is 64 and the maximum number of training epochs is 400, but training can be stopped when training loss stops improving after more than 10 epochs. The Adam optimizer is used and the initial learning rate is 0.001. The learning rate is decreased by a factor of 0.5 when the training loss stops improving after more than five epochs. The tuning parameter σ^2 is empirically set to 0.013. For the shared-MLP, it contains three 1D convolutional layers, each followed by a batch normalization layer and a ReLU layer for non-linearity. In the second shared-MLP (the decoder), the second 1D convolutional layer is followed by an additional dropout layer with a dropout rate of 0.3, and the randomly generated dropout mask is identical for the feature vector of each detection point. Finally, this study uses two binary cross-entropy losses to measure the difference between the predicted results and the true values of static labels and moving labels, respectively. Since the loss of ego-motion estimation is closely related to the loss of static prediction, errors in ego-motion are not backpropagated. To mitigate the influence of low-quality training examples, the final loss is the sum of the two cross-entropy losses multiplied by the sample weight (described in more detail in [143]).

[4] x must be nonzero.

6.4. Results and Discussion

This section begins by describing additional modifications applied to the RadarScenes dataset [15]. While the dataset itself has been detailed in Chapter 2.2, these modifications are introduced to enable training and evaluation of the dual tasks proposed in this chapter. Subsequently, the evaluation results of the proposed method are presented and compared against selected approaches from the literature. Finally, quantitative results based on the proposed evaluation metrics are reported, followed by qualitative examples that provide a clearer visual interpretation.

6.4.1. Additional Changes to RadarScenes

Although the RadarScenes dataset records accurate vehicle motion and provides manually labeled point clouds, due to the new task proposed in this study, four additional processing steps are required in order to generate GT data for model training and evaluation. Firstly, radar detections of static objects and false positives are not individually labeled in the dataset. To distinguish them, the recorded vehicle motion is used to localize static detections from the radar point cloud. Specifically, similar to Equation 6.4, the vehicle motion is first transformed to radar motion, and then the GT static labels can be calculated. For moving objects, the GT moving labels are generated based on the class labels provided by the dataset, where 0 represents non-moving ('Class 11') and 1 represents moving ('Class 1 to 10'). If a detection is neither labeled as static nor moving, it is defined as one of the false positives. Conversely, if a detection is labeled as both static and moving due to, for example, mislabeling in the GT, it is corrected to static but non-moving, as vehicle GT motion is more reliable and trustworthy than human annotations.

In the second processing step, the effect of vehicle acceleration on the measured radial velocity is resolved. As detailed in Section 4.4.7 (Figure 4.9c), due to the vehicle's non-zero acceleration, the Doppler frequency and the associated phase shift will vary with slow time and the estimated radial velocity will not match the vehicle velocity. Therefore, the GT static labels generated solely based on vehicle motion may be inaccurate. As shown in Chapter 4, *DeepEgo+* can mitigate this effect by using a two-step signal processing method with NNs. The first step locates the static detection points, and the second step compensates for the effect of non-zero acceleration and estimates the vehicle ego-motion. Therefore, in this chapter, if the *DeepEgo+* ego-motion estimation error is below a preset threshold, its output is used to help better localize static objects; otherwise, the vehicle's GT motion is used.

Thirdly, the RadarScenes dataset contains 158 two-minute-long individual sequences from each of the four radars. In almost half of the sequences, the ego-vehicle is (or almost) stationary and monitors moving objects. This is well-suited for tasks such as object detection and motion segmentation. However, for the dual task of ego-motion estimation and static-moving object segmentation, a dataset containing an ego-vehicle in constant motion is desired for both training and evaluation. Therefore, this work uses a minimum driving distance of 500 meters for sequence selection, resulting in 63 radar sequences captured in challenging scenarios such as highways and city traffic. Nevertheless, these 63 radar sequences still contributed more than 2 hours of recording time, which is equivalent to a driving distance of more than 70 km. It is worth mentioning that the reduction in the number of sequences necessarily increases the difficulty of the task, because not only is it more diffi-

Table 6.1: The radar mounting position and the number of labeled moving objects in the selected 63 radar sequences from the RadarScenes [15] dataset.

Radar Name	**Radar 1**	**Radar 2**	**Radar 3**	**Radar 4**
Pointing Direction	Side-looking	Front-facing	Front-facing	Side-looking
Labeled Moving Objects	756	3484	3685	2396
Object's Lifespan < 5 Frames	106	407	207	227

cult and meaningful to distinguish between static and moving objects when the vehicle is in constant motion rather than stationary, but also training neural networks on smaller datasets can lead to some well-known challenges such as overfitting and generalization problems.

The last processing step is to deal with short-lived labeled moving objects. As shown in Table 6.1, due to different installation angles, the number of moving objects observed by the four radars varies greatly. Radar 1 faces the side of the street and picks up minimal objects, while Radar 2 and Radar 3 face forward, cover both lanes, and pick up the most objects. In addition, since Radar 1 and Radar 4 face sideways, moving objects often appear at close range and enter and leave the radar field of view quickly, resulting in a very short lifespan. Moving objects with short lifespans contain few temporal features and may confuse model training and increase false alarm rates. Therefore, moving objects with a lifespan shorter than five radar frames (around 0.3 s) are labeled as non-moving for the sake of training. Also, the data from Radar 1 are not used for performance evaluation because there are only about 10 moving objects per sequence on average. Finally, unless otherwise specified, the following experiments are all conducted using Radar 3 data, and the 'leave-one-out'[5] training, validation, and testing strategy is adopted so the performance on 'unseen' data can be measured.

6.4.2. Comparisons with State of the Art (SOTA)

Before presenting the detailed performance evaluation and comparison, it is worth mentioning that the proposed approach differs from previous studies in two aspects. Firstly, this study aims to achieve both ego-motion estimation and static-moving object segmentation simultaneously, which is a first of its kind and also introduces a different evaluation method. Secondly, motivated by many other radar downstream applications that are premised on separating static [13, 63, 169, 172] or moving objects [148, 173], this study redefines the conventional objectives in radar segmentation and provides a one-step solution for these applications. Therefore, the authors must acknowledge that it becomes challenging and difficult to make a fair comparison of the proposed method with the state-of-the-art (SOTA) methods in the literature given the above differences.

Table 6.2 summarizes a list of representative previous studies in the field of radar-based ego-motion estimation and segmentation. For radar-based segmentation, the closest previous study to this chapter is [161], which also performs moving object segmentation, while other studies seek accurate and detailed class labels for moving objects. Nevertheless, the authors believe that the proposed method is more competitive than previous studies in the

[5] The test radar sequences are taken out, one by one, from the selected 63 radar sequences, and the remaining sequences are used for model training and validation following the 80%-20% rule. After all 63 sequences have been used once as the test sequence, the final performance of the tested method is measured and averaged.

Table 6.2: Comparison between the proposed method and representative studies in the literature. For ego-motion estimation (Ego-M.), *DeepEgo* [57] is selected and its performance is measured in RTE_50 after training with the same radar sequences as the proposed method. For the segmentation task (Seg.), four previous studies are selected and their reported performances in terms of IoU and F1 scores are shown in the table.

References	Radar Task	Main Backbone	Odometry Data	Point Cloud Aggregation	Parameters (M)	IoU / F1	RTE_50
[57]	Ego-Motion	MLP	Not Required	Not Required	0.8	N/A / N/A	16.00
[161]	Segmentation	Transformer	Required	Fuse Multiple Radars	N/A	0.81 / N/A	N/A
[157]	Segmentation	Transformer	Required	Fuse Multiple Radars	4.5	N/A / N/A	N/A
[151]	Segmentation	Transformer	Required	Fuse 500 ms Radar Scans	7.36	N/A / 0.81	N/A
[149]	Segmentation	Transformer	Required	Fuse Multiple Radars	8.4	N/A / 0.80	N/A
Proposed	Ego-M. & Seg.	MLP	Not Required	Not Required	0.15	0.86 / 0.92	1.8

following aspects. Firstly, all listed works require knowledge of the vehicle ego-motion. In most cases, ego-motion is used to compensate for the measured radial velocity, which has been shown to be a key feature for identifying static and non-static objects [145, 149]. However, if ego-motion is known, the input radar point cloud in these studies can be significantly simplified by removing all static objects, making it easier to distinguish moving objects from false positives and saving computational resources. This is because static objects and false positives together contribute almost 97% of radar detections in the RadarScenes dataset. Furthermore, dependence on external odometry sensors can undermine sensor independence and reduce system robustness, as this introduces risks of erroneous outputs or synchronization problems. In contrast, the proposed method can work independently on unprocessed radar point clouds and does not rely on any external sensors or motion compensation. The special network design enables it to capture relevant features from the point cloud, thereby not only separating moving objects but also localizing static objects and estimating vehicle motion.

Secondly, all listed segmentation tasks perform point cloud aggregation across multiple radars or over a period of time. One reason for this is the low angular resolution of radars, while point cloud aggregation helps enrich the geometric features of objects. However, point cloud aggregation requires good sensor synchronization and the knowledge of the relative extrinsic parameters between radars. Furthermore, without motion compensation, the aggregation effect can deteriorate in highly dynamic scenes, where the shape of objects changes with speed. For example, a fast-moving car may look like an elongated truck after aggregation. In addition, temporal information may be lost after aggregation across multiple radar frames. Moreover, the aggregation process inevitably introduces inference delays, which may affect applications that require a real-time response. On the contrary, although the proposed method uses multiple single-frame radar point clouds, they are arranged in time sequence, processed independently, and can form a moving window to provide instantaneous predictions for the current time.

Thirdly, previous studies typically employ complex feature extraction backbones (such as Transformer [79]) to help capture crucial details so that the exact categories of moving objects can be distinguished in sparse and noisy radar point clouds. However, these backbone networks usually require very large datasets for model training, otherwise there may be risks of overfitting and poor generalization ability, while radar data collection is expensive and labeling sparse radar data is very time-consuming. Furthermore, for automotive applications, these ‘large’ networks typically require more computing resources and can incur higher latency, but the performance gain from using complex backbones for radar segmen-

tation is much smaller than for the same task in LiDAR. Therefore, this study breaks this convention and instead separates moving and static objects, which the authors believe is more appropriate and reliable for radar data, more beneficial for other downstream applications, and also helping to build a lighter network. As shown in Table 6.2, even for the dual task, the proposed method is the lightest of all listed methods and can be trained using less but more challenging data.

For ego-motion estimation, the previous SOTA method *DeepEgo* [57] is trained using the same dataset and compared with the proposed method. Firstly, both *DeepEgo* and the proposed method can achieve instantaneous ego-motion estimation without the need for point cloud aggregation and odometry data. In addition, both use lightweight backbone networks for feature extraction. By contrast, the proposed method achieves superior ego-motion estimation performance compared to *DeepEgo*. This is primarily because the proposed method leverages temporal information from previous radar frames to better localize static objects and estimate vehicle motion in the current frame.

Finally, it is worth mentioning that, except for *DeepEgo*, the segmentation performance (i.e., IoU and F1) of the selected works is directly taken from the corresponding references[6]. This is because the work proposed in this chapter differs significantly from previous studies, not only in the main objective but also in the requirements for the size of training data, point cloud processing, and external sensor information. Therefore, performance comparison with compromises in re-implementation will be unfair to either the previous studies or the proposed approach in this chapter.

6.4.3. Performance over Moving Window Lengths

The length of the input moving window is an important hyperparameter of the proposed method because it determines how many historical radar point clouds and how much temporal information the proposed NN can exploit. As explained in Section 6.3.2, this temporal information is crucial for localizing moving objects since radial velocity is not compensated and the input point cloud is sparse. To show its effect, Figure 6.5 provides the performance of the proposed method for moving object segmentation and ego-motion estimation with different window lengths. As expected, the missed detection rate decreases rapidly with the increase in input length, indicating that more moving objects are correctly segmented. However, the RTE_50 metric does not change significantly, reflecting that longer moving window lengths may have little effect on ego-motion estimation performance, which was also expected since static objects already show unique patterns in the single-frame Doppler profile (Figure 6.3). Finally, unlike point cloud aggregation, moving the window does not affect timely predictions, but it still requires more memory resources than single-frame methods. Therefore, it is recommended to adjust this parameter based on application requirements. In this study, the input window length is set to 8 for all experiments.

6.4.4. Performance over Distances

One of the advantages of radar sensors is their long detection range. The automotive radar used in the RadarScenes dataset can cover a detection range of up to 100 meters. However, the spatial cross-range resolution of radar is finer at close range and coarser at long range.

[6]The definitions of IoU and F1 may also differ from this chapter.

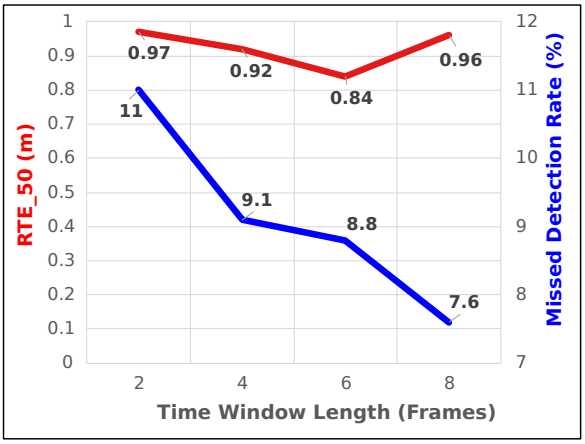


Figure 6.5: The performance of the proposed method for moving object segmentation and ego-motion estimation with different lengths of the input moving window (in radar frames). The blue solid line represents the missed detection rate of the model, and the red solid line represents the RTE_50.

This is because, with a fixed azimuth resolution, the area covered by one resolution cell increases with distance, even if the range resolution remains constant. Therefore, distant moving objects may only produce a few detection points, and it is important to understand how this will degrade the segmentation performance of the proposed method. Figure 6.6 shows the missed detection rate of the proposed method measured at different range thresholds. When the radar's FoV is limited to a maximum range of 15 meters, the proposed method misses only 5.2% of TPs, e.g., a moving object appears in 100 radar frames but is missed in only about five frames. However, as the maximum range increases from 15 to 50 meters, the missed detection rate and the total number of TPs within the radar FoV increase rapidly. Finally, the deterioration slows down after 50 meters, reaching a missed detection rate of 7.5%, and a total of 187 K TPs are detected within 100 meters.

6.4.5. Ablation Study on Input Features

As mentioned in Section 6.3, the proposed method requires that the input radar point cloud contains at least three types of object features, namely, range, AoA, and radial velocity. To understand which features are important for the task of ego-motion estimation and moving object segmentation, this section conducts an ablation study on the selected input features. As shown in Table 6.3, first, the radial velocity measurements are the most valuable object feature for both ego-motion estimation and moving object segmentation. For ego-motion estimation, the measured radial velocity and AoA help clearly distinguish between static and non-static objects, as shown by the Doppler profiles. This can explain the degradation in ego-motion performance when AoA is removed from the input data. For moving object segmentation, even if the angle and range information are preserved, it is very difficult to distinguish between moving and non-moving objects without radial velocity. This is because radial velocity helps separate static objects, making moving objects more visible than the false positives in the radar point clouds. However, it is also interesting to note

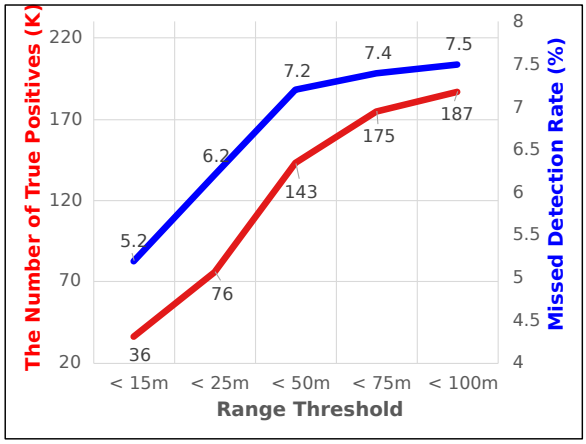


Figure 6.6: The effect of thresholding on the measurement range of Radar 3. The threshold changes the size of the radar field of view, thereby changing the number of moving objects within it. The red solid line shows the relationship between the range threshold and the number of TPs. Note that the number of TPs every moving object can contribute is the same as its lifespan measured in number of frames. The blue solid line shows the relationship between the range threshold and the performance of the proposed method in terms of missed detection rate.

Table 6.3: Effects of different input features on ego-motion estimation and moving object segmentation.

Input Conditions	F1 Score	RTE_50 [m]
No Range	0.91	1.99
No Azimuth AoA	0.85	62.3
No Radial Velocity	0.58	58.7
All Features	0.93	0.96
No Range ($< 15m$)	0.93	N/A
All Features ($< 15m$)	0.97	N/A

that even without angle information, the proposed method still retains the ability to detect moving objects, albeit with poor ego-motion estimation performance. Finally, among the three tested input features, the range information appears to have the least impact on the performance of the dual task. This can also be intuitively understood from the Doppler profile, where moving objects, static objects, and false positives also show clear temporal and spatial distinctions across multiple radar frames. However, as predicted in Section 6.3.1, range information becomes more important for nearby moving objects, since these objects can occupy many angular cells and be spatially separated in the Doppler profile. As shown in the table, the performance gap between 'All Features' and 'No Range' is larger when a 15-meter range threshold is applied.

6.4.6. Performance over Radar Positions

Previous experiments were conducted using data from Radar 3 because this sees the most moving objects, which is in line with the goals of this study. However, it is also important to show that the proposed method can work at other positions or mounting angles. Therefore,

Table 6.4: The performance of the proposed method under different radar installation positions and angles. In this experiment, the proposed method is trained and evaluated separately using data from different radars. The last row of the table applies a maximum threshold of 15 meters to the detection range of Radar 4. The model's ego-motion estimation performance at the given threshold is not measured and is therefore marked as 'N/A'.

Conditions	FDR (%)	MDR (%)	F1 Score	S-RMSE V_x^{car} (cm/s)	S-RMSE ω (deg/s)	RTE_50 (m)
Radar 2	6.4	7.7	0.93	0.47	0.11	1.4
Radar 3	6.4	7.6	0.93	0.37	0.12	0.96
Radar 4	11.5	16.1	0.86	2.03	0.11	0.41
Radar 4 ($< 15m$)	7.8	5.0	0.94	N/A	N/A	N/A

in addition to Radar 3, this section also applies the proposed method to data from Radar 2 and Radar 4. As shown in Table 6.4, the proposed method performs almost the same on Radar 2 and Radar 3. However, when using data from Radar 4, while the model can still perform well in ego-motion estimation, its segmentation performance degrades. One reason for this is that Radar 4 is looking sideways at the passing lane, and moving objects can move perpendicular to the direction the radar is pointing, affecting measured radial velocities. Furthermore, side-looking radars can also capture random objects on the street that are either far away (a few detection points) or briefly within the radar's FoV. To examine scenarios closer to real-world use, a maximum threshold of 15 meters is applied to the detection range of Radar 4, so the radar only covers the overtaking and oncoming lanes. Under this condition, the model performs just as well on Radar 4 as on Radars 2 and 3, demonstrating the effectiveness of the proposed method even under adverse mounting angles.

6.4.7. Qualitative Result: Static-Moving Object Segmentation

While the previous sections quantitatively evaluated the performance of the proposed method, this section provides qualitative tools for better visual understanding. As shown in Figure 6.7, in addition to providing vehicle ego-motion, the proposed method can also achieve simultaneous segmentation of static and moving objects in a variety of challenging scenarios, such as driving on a narrow and busy street, driving at high speed in an open area, or driving but being surrounded by slow-moving pedestrians. Different than previous studies, the proposed method can directly segment sparse radar point clouds without the need for point cloud aggregation. It is also worth noting that the predicted static and moving objects can be used by many radar downstream tasks. For example, as shown in the figure, clustering algorithms such as DBSCAN can be applied to generate moving instances, and then classic multi-target tracking algorithms can be used to estimate their motion states or trajectories. Finally, detections that are neither labeled as moving nor static are classified as false positives in this study. Typically, reflections coming from side-lobes and multipath can be labeled as false positives. However, as shown in the third column of the figure, detections originating from the static treetops to the left of the ego-vehicle are also marked as false positives in both the GT and the prediction. This is because the radar sensors used in the RadarScenes dataset only have azimuth and range resolution, but elevation also affects the measured radial velocity, leading to incorrect predictions and GTs for static objects that are not at the same level as the radar sensor. However, if in future works these detections can be correctly segmented, it may be possible to also estimate their heights [174].

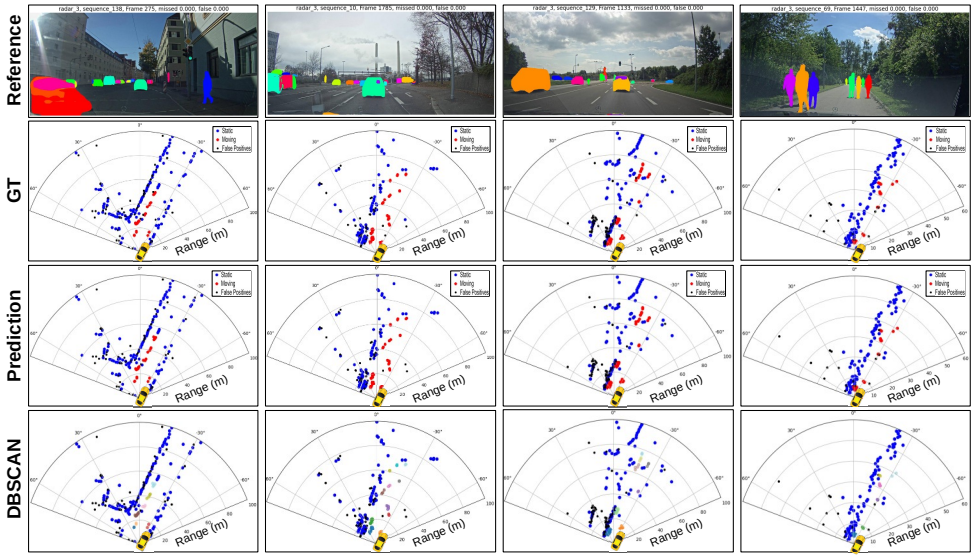


Figure 6.7: Qualitative results of the proposed method for static and moving object segmentation in 2D polar plots. The proposed method is tested in four different driving scenarios (shown in four columns). The first row shows images from an on-vehicle reference camera, the second row shows the ground truth, the third row shows the model predictions, and the last row shows the clustering output after applying DBSCAN to the predictions. In the second and third rows, moving objects are marked in red, static objects are marked in blue, and false positives are marked in black.

6.4.8. Qualitative Result: Localization and Mapping

In addition to segmentation, this method can also simultaneously estimate the 2D motion of the moving ego-vehicle, including forward velocity and rotation rate. Furthermore, by incorporating temporal information, this study can also calculate the vehicle 2D trajectory, thereby constructing a point cloud map. Figure 6.8 shows vehicle trajectories calculated from the model's output on four test sequences from Radar 3. Although in each scene the ego-vehicle travels more than 500 meters and the trajectory accumulates errors in the ego-motion estimation, the estimated trajectory still closely follows the GT vehicle trajectory, demonstrating the reliable performance of the proposed method for vehicle localization. Furthermore, thanks to the explicit separation of static objects, the outlines of streets, road edges, and surrounding infrastructure can be clearly seen in the zoomed-in figure, which is of great value for applications such as mapping, drivable road space detection, and semantic segmentation. A more vivid example of using predicted labels for environment mapping is shown in Figure 6.9, which shows a dynamic environment with six (groups of) walking pedestrians captured by Radar 2 and Radar 3. The trajectories of these moving instances can be clearly observed in the original accumulated radar point cloud. In contrast, filtering the radar point cloud based on the model prediction removes these trajectories and false positives, leaving behind a distinct outline of the environment.

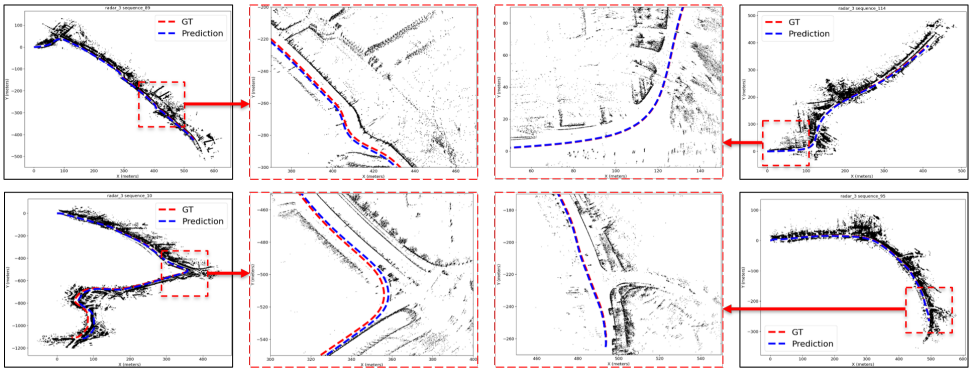


Figure 6.8: Qualitative results of the proposed method for vehicle ego-motion estimation. In this experiment, the proposed method is tested using four sequences from Radar 3, and the estimated ego-motion is converted into vehicle trajectories and displayed on a 2D plane. The red dashed line represents the ground truth trajectory calculated based on the vehicle true motion state, and the blue dashed line represents the vehicle trajectory calculated based on the estimated motion state. The black dots are predicted static objects, accumulated over all radar frames of the tested sequence.

6.5. Conclusion

Knowledge of ego-vehicle velocity and the positions of moving and static objects is sufficient for many radar perception tasks and ensures driving safety, especially in harsh environmental conditions where optical sensors cannot operate. Therefore, unlike traditional radar segmentation research, which requires significant effort to overcome the fundamental limitations of existing radars with limited success, this research reframes the radar segmentation objective as a dual task, which is simpler but more meaningful and reliable for radar data. Specifically, the outcome of this research is a neural network-based solution that can work independently, perform automatic feature extraction, separate moving and static objects, and provide vehicle motion status, all at the same time. According to the literature review, this approach could have a significant impact in radar signal processing, as the authors found that understanding vehicle motion and locating static and moving objects are crucial initial steps in many radar perception tasks. The method has been thoroughly evaluated on the RadarScenes dataset using challenging scenes, novel evaluation metrics, and refined object labels. Results confirm both the feasibility of the dual task using unprocessed radar point clouds and the superior performance of the proposed approach. The network is extremely lightweight (0.15 M parameters) yet achieves high scores in moving object segmentation (IoU = 0.86, F1 = 0.92) and accurate ego-vehicle motion estimation and localization (RTE_50 = 1.8 m).

For future work, extending the approach to estimate and track the velocities of other moving objects beyond the ego-vehicle would be a promising direction. Furthermore, a systematic analysis of single-task training and joint-task training can further quantify the advantages and disadvantages of the proposed framework and provide deeper insights for future research directions.

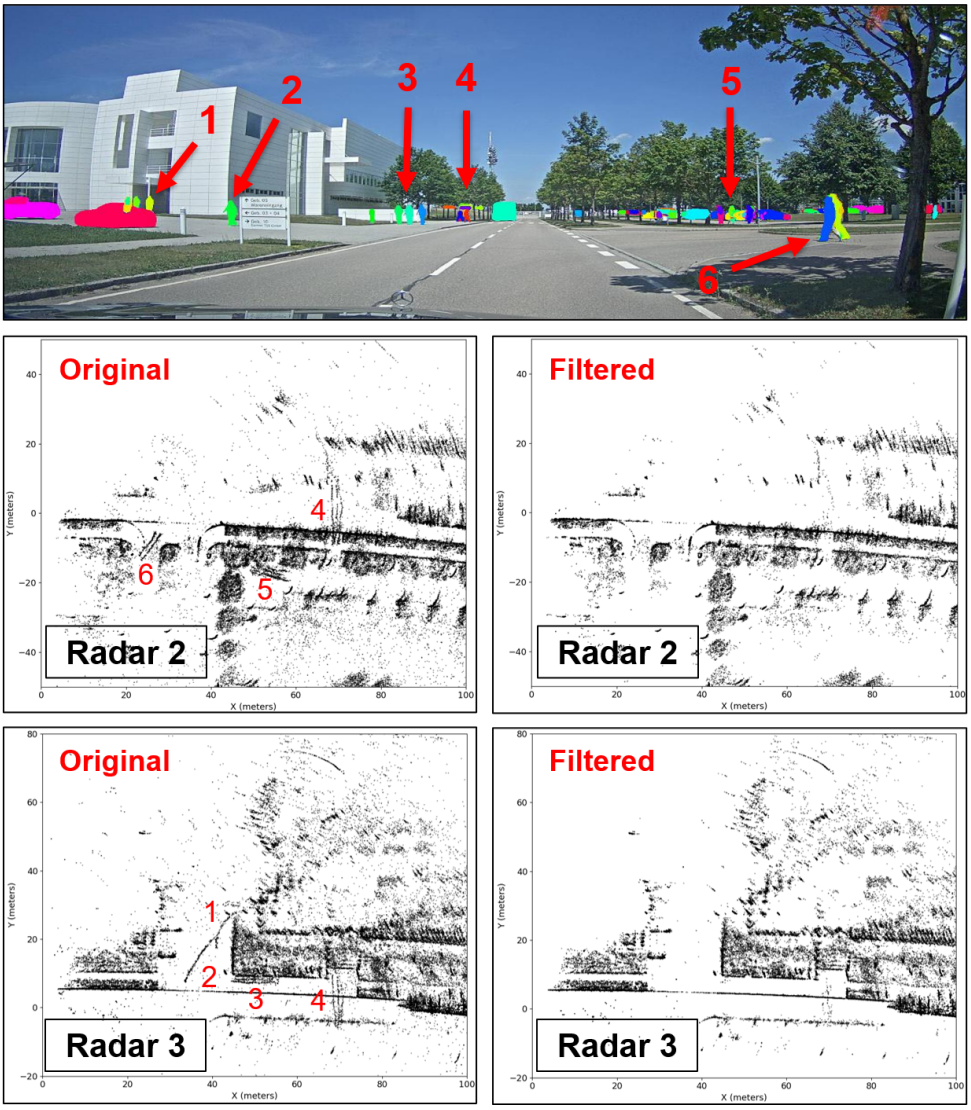


Figure 6.9: Point cloud map constructed using the output of the proposed method. The first row shows the image from the reference camera, in which there are six (groups of) moving pedestrians. The second row shows the point cloud images generated using Radar 2, and the last row is generated using Radar 3. The original map is generated by fusing multiple radar point clouds so that the trajectories of moving objects can be seen. The filtered map is generated using the model predictions, thus only showing the static environment.

7

Conclusions

7.1. Major Results and Novel Contributions

This dissertation has demonstrated that automotive radar can be elevated from a supporting sensor to a primary perception modality for robust autonomous driving. Four core contributions are summarized in the following paragraphs:

- *DeepEgo: Instantaneous Vehicle Ego-Motion Estimation (Chapter 3)*

  A novel approach, named *DeepEgo*, has been developed to estimate the instantaneous ego-motion of a moving vehicle using only one radar sensor. Unlike prior methods that rely on the majority-inlier assumption, its key novelty lies in robust performance even under high outlier ratio scenarios. To this end, the proposed method employs a hybrid architecture that combines a deep-learning-based frontend with a weighted least squares backend. The frontend performs automatic feature extraction on the input radar point cloud and predicts point-wise weights to separate reliable inliers (e.g., reflections from static objects) from outliers (e.g., moving objects and false positives), while the backend estimates vehicle motion from the weighted inputs. A novel loss function is proposed to guide training, and the system can be trained end-to-end in a self-supervised manner using odometry data, eliminating the need for manual annotation. In contrast to conventional methods, *DeepEgo* operates on a single radar frame (instantaneous estimation), requires no iterative optimization, and retains a degree of interpretability through its hybrid design. To the best of the author's knowledge, this is the first vehicle motion estimation method to be extensively validated on large-scale real-world radar data. Experimental results on the RadarScenes dataset showed that it achieved approximately 50% higher accuracy and operated 129 times faster than state-of-the-art baselines. In particular, the method yielded a small root mean square error (RMSE), with the translational velocity error about 11.5 cm/s and the rotational velocity error around 0.85 deg/s. Importantly, while it performed comparably to traditional methods under favorable conditions, it significantly outperformed them in more challenging real-world settings. The novelty and practical value of *DeepEgo* have also been recognized through a granted patent. **In summary, *DeepEgo* provides the foundation for low-cost, robust, and interpretable radar-only odometry, offering a viable alternative to GPS, LiDAR, or camera-based vehicle odometry systems.**

- *DeepEgo+: Multi-Radar Fusion for Robust Ego-Motion Estimation (Chapter 4)*

  The second major contribution of this dissertation is *DeepEgo+*, a novel framework that extends the capabilities of *DeepEgo* by enabling ego-motion estimation through the fusion of multiple unsynchronized automotive radars. This contribution addresses a key open challenge: although vehicles are commonly equipped with several radars, prior work has relied on the impractical assumption of perfect synchronization, preventing robust multi-radar odometry in real-world settings. To the best of the author's knowledge, *DeepEgo+* is the first method to demonstrate accurate and robust fusion of radar networks without requiring synchronization. The framework adopts a decentralized late-fusion architecture in which each radar independently computes an initial motion estimate, which is subsequently fused by a neural-network–based Kalman filter. This architecture makes the system inherently resilient to synchro-

nization issues, sensor failures, and environmental outliers, while also compensating for the strong influence of vehicle acceleration that affects traditional radar odometry methods. Additionally, although not a primary design goal, *DeepEgo+* demonstrates for the first time in the literature its ability to offset the effects of vehicle acceleration on motion estimation. Experimental evaluation on the RadarScenes dataset showed that *DeepEgo+* improved estimation accuracy by approximately 50% compared to state-of-the-art baselines, including *DeepEgo*. Specifically, it achieves remarkably small estimation errors, with an RMSE of only 5.3 cm/s for translational velocity and 0.44 deg/s for rotational velocity. Sensitivity to acceleration was also significantly reduced, as reflected by a drop in the normalized correlation coefficient (NCC) from 0.71 in prior work to 0.17 with *DeepEgo+*. The method also maintained accurate performance in scenarios with high outlier ratios and under partial sensor failures, confirming its robustness in challenging real-world conditions. The novelty and practicality of *DeepEgo+* have likewise been recognized through a granted patent. **In summary, *DeepEgo+* establishes the feasibility of scalable, robust, and synchronization-free radar fusion, marking an important step toward the practical deployment of multi-radar perception systems in autonomous driving.**

- *From Ego-Motion Estimation to Radar Extrinsic Calibration (Chapter 5)*

  The third contribution of this dissertation addresses the problem of radar extrinsic calibration, with a particular focus on estimating the radar mounting angle under operational driving conditions. To the best of the author's knowledge, this is the first work to demonstrate that the radar mounting angle can be accurately determined in complex driving scenarios without requiring controlled environments, dedicated radar targets, or specially designed driving routes. The proposed method exploits the principle of lateral velocity equality on a rigid body and formulates a system of linear equations, using vehicle yaw rates from an IMU together with radar ego-motion estimated by a neural network to infer the mounting angle. By design, the method is more robust than previous studies, since it relies on the combination of radar and IMU, two sensing modalities that remain reliable under adverse weather and lighting conditions, rather than on more fragile alternatives such as GPS. Moreover, unlike traditional calibration methods, which are typically performed offline and assume static environments or majority inliers, the proposed solution can work in real-time during normal driving, making it both practical and reliable for long-term automotive deployment. Experimental evaluation on the RadarScenes dataset demonstrated a mean absolute error below 0.02 degrees with convergence within 25 seconds, significantly outperforming state-of-the-art approaches such as RANSAC- and Kabsch-based baselines. Notably, the proposed method also achieved lower relative trajectory error than even the ground-truth angles provided in the dataset, further underlining its accuracy. **In summary, this contribution introduces a robust and practical online radar calibration method that enables vehicles to automatically correct misalignments during operation. This can enhance the reliability of radar perception pipelines and support the long-term robustness of autonomous driving systems.**

- *Simultaneous Radar Segmentation and Ego-Motion Estimation (Chapter 6)*

The final contribution of this dissertation is the development of a unified radar perception framework that jointly performs static–moving segmentation and ego-motion estimation. To the best of the author's knowledge, this is the first radar-only method capable of solving both tasks simultaneously, thereby filling a key gap in the radar perception processing chain that is crucial for many downstream applications. The framework employs simple yet effective neural building blocks, designed from key observations of radar point clouds and object characteristics. In particular, multi-layer perceptrons are used for spatial feature extraction, recurrent networks capture temporal dependencies, and novel update heads refine segmentation using the estimated motion, tightly coupling the two tasks. Unlike existing approaches, which typically segment only moving objects and rely on external odometry or multi-frame aggregation, the proposed framework operates directly on single-frame radar data without auxiliary sensors. This makes it both sensor-independent and computationally efficient, while at the same time providing a holistic solution to two fundamental perception problems. Evaluation on the RadarScenes dataset demonstrated state-of-the-art performance, achieving an intersection-over-union (IoU) of 0.86 and an F1 score of 0.92 for segmentation, while also delivering accurate odometry with a relative trajectory error (RTE_{50}) of 1.8 m. Moreover, the network has only 0.15M trainable parameters, the smallest among comparable studies, and remained robust across diverse real-world scenarios, including urban traffic and highway driving, confirming its generalization ability. **In summary, this contribution advances radar perception toward holistic and lightweight scene understanding, showing that segmentation and motion estimation can be solved jointly in a compact, radar-only framework that is accurate, real-time-capable, and practical for automotive deployment.**

7.2. Recommendations for Future Research

Owing to the scope and time constraints of this PhD project, some research directions and extensions of the ideas developed here could not be fully explored. The following section highlights potential research lines that can serve as a continuation of this dissertation.

- *Cross-Sensor Generalization of Radar Odometry*

  To apply the work presented in Chapter 3 to autonomous vehicles, *DeepEgo* needs to be trained using data from the specific radar installed on a specific vehicle. In other words, if *DeepEgo* is trained with one radar but applied to the data of another radar with different extrinsic parameters (e.g., a different mounting angle), its performance may degrade significantly. This limitation arises from a fundamental challenge of deep learning: without explicit prior knowledge, neural networks can only generalize within the distribution of their training data. As a result, deployment becomes difficult and performance degradation may occur when extrinsic parameters change unexpectedly. To address this issue, it is necessary to develop an advanced version of *DeepEgo* that, once trained, can operate robustly across radars with different parameter settings. One potential approach is to employ a pretrained feature extraction backbone extensively trained on simulated radar data, and then adapt this backbone by connecting it to a prediction head and fine-tuning it with real radar data, thereby learning to handle real-world imperfections. Another promising direction is to apply

azimuth-shift augmentation, where during training, all detection points are shifted by a random azimuth offset and the ego-motion is adjusted accordingly. This moves the radar point cloud across the angular domain and may reduce the model's overfitting to the radar's specific field of view.

- *Acceleration-Aware Radar Ego-Motion Estimation*

  Although not a primary objective, the work presented in Chapter 4 demonstrated that the proposed distributed neural network architecture could offset the impact of vehicle acceleration on ego-motion estimates. However, this comes at the cost of reduced interpretability in the intermediate outputs of the neural network–based Kalman filter, since it must internally compensate for inaccuracies in the initial ego-motion estimates rather than explicitly modeling the effects of continuous vehicle motion. To address this limitation, it would be meaningful to incorporate a specific model of the acceleration/deceleration effect. As vehicle acceleration can be estimated during the Kalman filtering process, one possible approach would be to calculate the corresponding Doppler frequency shift induced by acceleration and compensate for it in the initial ego-motion estimates.

- *Exploiting Multi-Radar Systems for Self-Calibration*

  The work presented in Chapter 5 employs the IMU as a reference sensor for radar calibration. While the IMU is more robust under adverse weather and lighting conditions compared to alternatives such as GPS or optical sensors, and the proposed method achieves fast convergence, it can still suffer from occasional glitches and erroneous outputs. At the same time, modern autonomous vehicles are typically equipped with multiple radars, which presents an opportunity to develop a calibration framework that leverages the radar sensor network itself to detect misalignments and perform automatic correction. A key challenge in this direction is to address the synchronization issues inherent to multi-radar systems, since most existing approaches assume synchronized radar nodes.

- *Extending Joint Perception from Point Clouds to Radar Cubes*

  The work presented in Chapter 6 provides a unified solution for radar perception based on point cloud representations. However, to apply it in practice, the method still requires a preprocessing step using detection algorithms such as CFAR, since it does not operate directly on raw radar data cubes. Recent studies [175, 176] suggest that methods operating on radar cubes may be superior in segmenting small objects, as valuable information can be lost during the detection process [146, 152]. Nevertheless, there are currently no conclusive experimental comparisons to validate this claim. A meaningful future direction would therefore be to investigate this question experimentally. If confirmed, the next step would be to extend the current joint segmentation and ego-motion estimation framework to operate directly on radar cubes, enabling a model that can perform detection, segmentation, and ego-motion estimation simultaneously.

Bibliography

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[2] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J*, vol. 3016, pp. 1–16, 2014.

[3] S. Ayyasamy, "A comprehensive review on advanced driver assistance system," *Journal of Soft Computing Paradigm*, vol. 4, no. 2, pp. 69–81, 2022.

[4] S. Behere and M. Törngren, "A functional architecture for autonomous driving," in *Proceedings of the First International Workshop on Automotive Software Architecture*, pp. 3–10, 2015.

[5] R. Fan, S. Guo, and M. J. Bocus, *Autonomous driving perception*. Springer, 2023.

[6] B. Yang, J. Li, and T. Zeng, "A review of environmental perception technology based on multi-sensor information fusion in autonomous driving," *World Electric Vehicle Journal*, vol. 16, no. 1, p. 20, 2025.

[7] L. Fan, J. Wang, Y. Chang, Y. Li, Y. Wang, and D. Cao, "4d mmwave radar for autonomous driving perception: a comprehensive survey," *IEEE Transactions on Intelligent Vehicles*, 2024.

[8] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, "Radarslam: A robust simultaneous localization and mapping system for all weather conditions," *The International Journal of Robotics Research*, vol. 41, no. 5, pp. 519–542, 2022.

[9] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, "Radar slam: A robust slam system for all weather conditions," *arXiv preprint arXiv:2104.05347*, 2021.

[10] C. He, C. Meng, C. He, X. Fan, B. Wang, Y. Yan, and Y. Zhang, "See through vehicles: Fully occluded vehicle detection with millimeter wave radar," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 740–754, 2024.

[11] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.

[12] P.-C. Kung, C.-C. Wang, and W.-C. Lin, "A normal distribution transform-based radar odometry designed for scanning and automotive radars," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14417–14423, IEEE, 2021.

[13] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 869–874, IEEE, 2013.

[14] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann, "Instantaneous lateral velocity estimation of a vehicle using doppler radar," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 877–884, IEEE, 2013.

[15] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J. Tilly, J. Dickmann, and C. Wöhler, "RadarScenes: A Real-World Radar Point Cloud Data Set for Automotive Applications," Mar. 2021.

[16] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[17] F. Engels, P. Heidenreich, M. Wintermantel, L. Stäcker, M. Al Kadi, and A. M. Zoubir, "Automotive radar signal processing: Research directions and practical challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 865–878, 2021.

[18] G. Hakobyan and B. Yang, "High-performance automotive radar: A review of signal processing algorithms and modulation schemes," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 32–44, 2019.

[19] X. Li, X. Wang, Q. Yang, and S. Fu, "Signal processing for tdm mimo fmcw millimeter-wave radar sensors," *IEEE Access*, vol. 9, pp. 167959–167971, 2021.

[20] M. A. Richards *et al.*, *Fundamentals of radar signal processing*, vol. 1. Mcgraw-hill New York, 2005.

[21] H. Rohling, "Ordered statistic cfar technique-an overview," in *2011 12th International Radar Symposium (IRS)*, pp. 631–638, IEEE, 2011.

[22] I. Roldan, A. Palffy, J. F. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "See further than cfar: a data-driven radar detector trained by lidar," in *2024 IEEE Radar Conference (RadarConf24)*, pp. 1–6, IEEE, 2024.

[23] P. S. Diao, T. Alves, B. Poussot, and S. Azarian, "A review of radar detection fundamentals," *IEEE Aerospace and Electronic Systems Magazine*, vol. 39, no. 9, pp. 4–24, 2022.

[24] B. Tan, Z. Ma, X. Zhu, S. Li, L. Zheng, L. Huang, and J. Bai, "Tracking of multiple static and dynamic targets for 4d automotive millimeter-wave radar point cloud in urban environments," *Remote Sensing*, vol. 15, no. 11, p. 2923, 2023.

[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231, 1996.

[26] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations research letters*, vol. 5, no. 4, pp. 171–175, 1986.

[27] S. Liu, J. Tang, Z. Zhang, and J.-L. Gaudiot, "Computer architectures for autonomous driving," *Computer*, vol. 50, no. 8, pp. 18–25, 2017.

[28] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on robotics and automation*, vol. 12, no. 6, pp. 869–880, 1996.

[29] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, "Imu-based localization and slip estimation for skid-steered mobile robots," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2845–2850, IEEE, 2007.

[30] Y. Gu, Y. Wada, L. Hsu, and S. Kamijo, "Vehicle self-localization in urban canyon using 3d map based gps positioning and vehicle sensors," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 792–798, IEEE, 2014.

[31] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. I–I, Ieee, 2004.

[32] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and systems*, pp. 1–9, Berkeley, CA, 2014.

[33] A. Xenaki, B. Gips, and Y. Pailhas, "Unsupervised learning of platform motion in synthetic aperture sonar," *The Journal of the Acoustical Society of America*, vol. 151, no. 2, pp. 1104–1114, 2022.

[34] 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.

[35] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 864–870, IEEE, 2016.

[36] Z. Hong, Y. Petillot, and S. Wang, "Radarslam: Radar based large-scale slam in all weathers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5164–5170, IEEE, 2020.

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

[38] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band," *IEEE transactions on microwave theory and techniques*, vol. 60, no. 3, pp. 845–860, 2012.

[39] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar—from first efforts to future systems," *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 135–148, 2021.

[40] S. Zollo and B. Ristic, "On polar and versus cartesian coordinates for target tracking," in *ISSPA'99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No. 99EX359)*, vol. 2, pp. 499–502, IEEE, 1999.

[41] I. Roldan, F. Fioranelli, and A. Yarovoy, "Self-supervised learning for enhancing angular resolution in automotive mimo radars," *IEEE Transactions on Vehicular Technology*, 2023.

[42] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. De Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, "milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 109–122, 2020.

[43] S. Zhu, F. Fioranelli, and A. Yarovoy, "Radar-only instantaneous ego-motion estimation using neural networks," in *2023 20th European Radar Conference (EuRAD)*, pp. 201–204, 2023.

[44] R. Zhu, Y. Liu, Z. Dong, Y. Wang, T. Jiang, W. Wang, and B. Yang, "Adafit: Rethinking learning-based normal estimation on point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6118–6127, 2021.

[45] M. Barjenbruch, D. Kellner, J. Klappstein, J. Dickmann, and K. Dietmayer, "Joint spatial-and doppler-based ego-motion estimation for automotive radars," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 839–844, IEEE, 2015.

[46] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn, and J. Dickmann, "A fast probabilistic ego-motion estimation framework for radar," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2015.

[47] M. Rapp, M. Barjenbruch, M. Hahn, J. Dickmann, and K. Dietmayer, "Probabilistic ego-motion estimation using multiple automotive radar sensors," *Robotics and Autonomous Systems*, vol. 89, pp. 136–146, 2017.

[48] C. D. Monaco and S. N. Brennan, "Radarodo: Ego-motion estimation from doppler and spatial data in radar images," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 475–484, 2020.

[49] K. Haggag, S. Lange, T. Pfeifer, and P. Protzel, "A credible and robust approach to ego-motion estimation using an automotive radar," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6020–6027, 2022.

[50] P. Meiresone, D. Van Hamme, W. Philips, and T. Verbelen, "Ego-motion estimation with a lowpower millimeterwave radar on a uav," in *International Conference on Radar Systems (RADAR 2022)*, vol. 2022, pp. 371–376, IET, 2022.

[51] W. Li, R. Chen, Y. Wu, and H. Zhou, "Indoor positioning system using a single-chip millimeter wave radar," *IEEE Sensors Journal*, vol. 23, no. 5, pp. 5232–5242, 2023.

[52] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using multiple doppler radars," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1592–1597, IEEE, 2014.

[53] J. Schlichenmaier, L. Yan, M. Stolz, and C. Waldschmidt, "Instantaneous actual motion estimation with a single high-resolution radar sensor," in *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pp. 1–4, IEEE, 2018.

[54] K. Thormann and M. Baum, "Single-frame radar odometry incorporating bearing uncertainty," in *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, pp. 1–7, IEEE, 2023.

[55] M. Michaelis, P. Berthold, T. Luettel, and H.-J. Wuensche, "Generating odometry measurements from automotive radar doppler measurements," in *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, pp. 1–8, IEEE, 2023.

[56] A. Galeote-Luque, V. Kubelka, M. Magnusson, J.-R. Ruiz-Sarmiento, and J. Gonzalez-Jimenez, "Doppler-only single-scan 3d vehicle odometry," *arXiv preprint arXiv:2310.04113*, 2023.

[57] S. Zhu, A. Yarovoy, and F. Fioranelli, "Deepego: Deep instantaneous ego-motion estimation using automotive radar," *IEEE Transactions on Radar Systems*, 2023.

[58] S. A. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE access*, vol. 7, pp. 97466–97486, 2019.

[59] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments," *IEEE Transactions on robotics*, vol. 39, no. 2, pp. 1476–1495, 2022.

[60] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3, pp. 2743–2748, IEEE, 2003.

[61] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[62] S. Lim, J. Jung, S.-C. Kim, and S. Lee, “Radar-based ego-motion estimation of autonomous robot for simultaneous localization and mapping,” *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21791–21797, 2021.

[63] T. Grebner, V. Janoudi, P. Schoeder, and C. Waldschmidt, “Self-calibration of a network of radar sensors for autonomous robots,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 6771–6781, 2023.

[64] F. J. Abdu, Y. Zhang, M. Fu, Y. Li, and Z. Deng, “Application of deep learning on millimeter-wave radar signals: A review,” *Sensors*, vol. 21, no. 6, p. 1951, 2021.

[65] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, “Lo-net: Deep real-time lidar odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8473–8482, 2019.

[66] Y. Cho, G. Kim, and A. Kim, “Unsupervised geometry-aware deep lidar odometry,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2145–2152, IEEE, 2020.

[67] W. Wang, B. Wang, P. Zhao, C. Chen, R. Clark, B. Yang, A. Markham, and N. Trigoni, “Pointloc: Deep pose regressor for lidar point cloud localization,” *IEEE Sensors Journal*, vol. 22, no. 1, pp. 959–968, 2021.

[68] 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.

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

[70] 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.

[71] S. Lim, J. Jung, B.-h. Lee, J. Choi, and S.-C. Kim, “Radar sensor-based estimation of vehicle orientation for autonomous driving,” *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21924–21932, 2022.

[72] R. Weston, M. Gadd, D. De Martini, P. Newman, and I. Posner, “Fast-mbym: Leveraging translational invariance of the fourier transform for efficient and accurate radar odometry,” in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2186–2192, IEEE, 2022.

[73] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

[74] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[75] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 922–928, 2015.

[76] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9224–9232, 2018.

[77] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[78] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[79] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268, 2021.

[80] F. Fent, F. Kuttenreich, F. Ruch, F. Rizwin, S. Juergens, L. Lechermann, C. Nissler, A. Perl, U. Voll, M. Yan, *et al.*, "Man truckscenes: A multimodal dataset for autonomous trucking in diverse conditions," *Advances in Neural Information Processing Systems*, vol. 37, pp. 62062–62082, 2024.

[81] M. Holder, S. Hellwig, and H. Winner, "Real-time pose graph slam based on radar," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1145–1151, IEEE, 2019.

[82] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[83] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[84] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, Spie, 1992.

[85] J. Civera, O. G. Grasa, A. J. Davison, and J. M. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of field robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[86] M. Heller, N. Petrov, and A. Yarovoy, "A novel approach to vehicle pose estimation using automotive radar," *arXiv preprint arXiv:2107.09607*, 2021.

[87] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 2)." `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2`, 2024.

[88] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Agha-Mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5739–5746, IEEE, 2020.

[89] C. Doer and G. Trommer, "x-rio: Radar inertial odometry with multiple radar sensors and yaw aiding," *Gyroscopy and Navigation*, vol. 12, no. 4, pp. 329–339, 2021.

[90] Y. Z. Ng, B. Choi, R. Tan, and L. Heng, "Continuous-time radar-inertial odometry for automotive radars," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 323–330, IEEE, 2021.

[91] M. Steiner, O. Hammouda, and C. Waldschmidt, "Ego-motion estimation using distributed single-channel radar sensors," in *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pp. 1–4, IEEE, 2018.

[92] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche, "Probabilistic vehicle tracking with sparse radar detection measurements.," *Journal of Advances in Information Fusion*, vol. 17, no. 2, 2022.

[93] V. Kubelka, E. Fritz, and M. Magnusson, "Do we need scan-matching in radar odometry?," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13710–13716, IEEE, 2024.

[94] V.-J. Štironja, L. Petrović, J. Peršić, I. Marković, and I. Petrović, "Rave: a framework for radar ego-velocity estimation," in *2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 1–6, IEEE, 2024.

[95] H. Kim, H. Jang, and A. Kim, "2d ego-motion with yaw estimation using only mmwave radars via two-way weighted icp," *arXiv preprint arXiv:2404.00830*, 2024.

[96] A. Galeote-Luque, V. Kubelka, M. Magnusson, J.-R. Ruiz-Sarmiento, and J. Gonzalez-Jimenez, "Doppler-only single-scan 3d vehicle odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13703–13709, IEEE, 2024.

[97] F. Trombetta, A. Albaba, M. Bauduin, H. Sahli, and A. Bourdoux, "Radar-network-based odometry and elevation estimation," in *2024 IEEE Radar Conference (RadarConf24)*, pp. 1–6, IEEE, 2024.

[98] S. Lovett, K. MacWilliams, S. Rajan, and C. Rossa, "Enhancing doppler ego-motion estimation: A temporally weighted approach to ransac," in *2024 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6, IEEE, 2024.

[99] P. R. M. De Araujo, A. Noureldin, and S. Givigi, "Towards land vehicle ego-velocity estimation using deep learning and automotive radars," *IEEE Transactions on Radar Systems*, 2024.

[100] P. K. Rai, N. Strokina, and R. Ghabcheloo, "4dego: ego-velocity estimation from high-resolution radar data," *Frontiers in Signal Processing*, vol. 3, p. 1198205, 2023.

[101] M. Pawłowski, A. Wróblewska, and S. Sysko-Romańczuk, "Effective techniques for multimodal data fusion: A comparative analysis," *Sensors*, vol. 23, no. 5, p. 2381, 2023.

[102] F. Hau, F. Baumgärtner, and M. Vossiek, "The degradation of automotive radar sensor signals caused by vehicle vibrations and other nonlinear movements," *Sensors*, vol. 20, no. 21, p. 6195, 2020.

[103] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[104] H. Chen, Y. Liu, and Y. Cheng, "Drio: Robust radar-inertial odometry in dynamic environments," *IEEE Robotics and Automation Letters*, 2023.

[105] S. Yang, M. Choi, S. Han, K.-H. Choi, and K.-S. Kim, "4d radar-camera sensor fusion for robust vehicle pose estimation in foggy environments," *IEEE Access*, 2023.

[106] Y. Zhuang, B. Wang, J. Huai, and M. Li, "4d iriom: 4d imaging radar inertial odometry and mapping," *IEEE Robotics and Automation Letters*, 2023.

[107] R. Yanase, D. Hirano, M. Aldibaja, K. Yoneda, and N. Suganuma, "Lidar-and radar-based robust vehicle localization with confidence estimation of matching results," *Sensors*, vol. 22, no. 9, p. 3545, 2022.

[108] J. Michalczyk, R. Jung, and S. Weiss, "Tightly-coupled ekf-based radar-inertial odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12336–12343, IEEE, 2022.

[109] M. Hoffmann, L. Krabbe, C. Schüßler, P. Gulden, and M. Vossiek, "Instantaneous ego-motion estimation using a coherent radar network," in *2022 19th European Radar Conference (EuRAD)*, pp. 321–324, IEEE, 2022.

[110] Z. Zeng, X. Dang, Y. Li, and X. Liang, "Multi-view fusion automotive radar slam," *IEEE Sensors Journal*, 2023.

[111] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[112] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the American Society of Mechanical Engineers*, vol. 82, no. 1, pp. 35–45, Mar 1960.

[113] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," *University of North Carolina*, 1995.

[114] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518, Springer, 1992.

[115] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar, "Kalmannet: Neural network aided kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[116] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations," in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pp. 1–4, IEEE, 2015.

[117] A. Eltrass and M. Khalil, "Automotive radar system for multiple-vehicle detection and tracking in urban environments," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 783–792, 2018.

[118] N. Scheiner, N. Appenrodt, J. Dickmann, and B. Sick, "Radar-based feature design and multiclass classification for road user recognition," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 779–786, IEEE, 2018.

[119] R. M. Burza, "Overview of radar alignment methods and analysis of radar misalignment's impact on active safety and autonomous systems," *Sensors*, vol. 24, no. 15, p. 4913, 2024.

[120] G. Yan, Z. Luo, Z. Liu, Y. Li, B. Shi, and K. Zhang, "Sensorx2vehicle: Online sensors-to-vehicle rotation calibration methods in road scenarios," *IEEE Robotics and Automation Letters*, 2024.

[121] H. S. Ham, "Alignment system and method for radar apparatus," Mar. 10 2015. US Patent 8,973,278.

[122] J. Boyd, "Automated radar heading calibration with collaborating participants and multi-sensor fusion," Master's thesis, Texas A&M University-Corpus Christi, 2021.

[123] R. Pinnock, "Radar apparatus for a vehicle and method of detecting misalignment," Mar. 26 2024. US Patent 11,940,555.

[124] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6dof calibration of 3d lidar and radar," in *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2017.

[125] K.-r. Choi, G.-h. Seo, J.-e. Lee, S.-h. Jeong, and J.-n. Oh, "Automatic radar horizontal alignment scheme using stationary target on public road," in *2013 European Microwave Conference*, pp. 1863–1866, IEEE, 2013.

[126] E. Wise, Q. Cheng, and J. Kelly, "Spatiotemporal calibration of 3-d millimetre-wavelength radar-camera pairs," *IEEE Transactions on Robotics*, 2023.

[127] L. Cheng and S. Cao, "Online targetless radar-camera extrinsic calibration based on the common features of radar and camera," in *NAECON 2023-IEEE National Aerospace and Electronics Conference*, pp. 294–299, IEEE, 2023.

[128] C. Schöller, M. Schnettler, A. Krämmer, G. Hinz, M. Bakovic, M. Güzet, and A. Knoll, "Targetless rotational auto-calibration of radar and camera for intelligent transportation systems," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3934–3941, IEEE, 2019.

[129] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Online multi-sensor calibration based on moving object tracking," *Advanced Robotics*, vol. 35, no. 3-4, pp. 130–140, 2021.

[130] L. Heng, "Automatic targetless extrinsic calibration of multiple 3d lidars and radars," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10669–10675, IEEE, 2020.

[131] Q. Cheng, E. Wise, and J. Kelly, "Extrinsic calibration of 2d millimetre-wavelength radar pairs using ego-velocity estimates," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 559–565, IEEE, 2023.

[132] K. T. Olutomilayo, M. Bahramgiri, S. Nooshabadi, and D. R. Fuhrmann, "Extrinsic calibration of radar mount position and orientation with multiple target configurations," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[133] A. Bobaru, C. Nafornita, and V. C. Vesa, "Unsupervised online horizontal misalignment detection algorithm for automotive radar," in *2022 14th International Conference on Communications (COMM)*, pp. 1–5, IEEE, 2022.

[134] Y. Bao, T. Mahler, A. Pieper, A. Schreiber, and M. Schulze, "Motion based online calibration for 4d imaging radar in autonomous driving applications," in *2020 German Microwave Conference (GeMiC)*, pp. 108–111, IEEE, 2020.

[135] C. Doer and G. F. Trommer, "Radar inertial odometry with online calibration," in *2020 European Navigation Conference (ENC)*, pp. 1–10, IEEE, 2020.

[136] T. Grebner, M. Linder, N. Kern, P. Schoeder, and C. Waldschmidt, "6d self-calibration of the position and orientation of radar sensors in a radar network," in *2022 19th European Radar Conference (EuRAD)*, pp. 157–160, IEEE, 2022.

[137] M. Z. Ikram and A. Ahmad, "Automated radar mount-angle calibration in automotive applications," in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–5, IEEE, 2019.

[138] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann, "Joint radar alignment and odometry calibration," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 366–374, IEEE, 2015.

[139] E. Wise, J. Peršić, C. Grebe, I. Petrović, and J. Kelly, "A continuous-time approach for 3d radar-to-camera extrinsic calibration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13164–13170, IEEE, 2021.

[140] R. Izquierdo, I. Parra, D. Fernández-Llorca, and M. Sotelo, "Multi-radar self-calibration method using high-definition digital maps for autonomous driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2197–2202, IEEE, 2018.

[141] P. S. Bokare and A. K. Maurya, "Acceleration-deceleration behaviour of various vehicle types," *Transportation research procedia*, vol. 25, pp. 4733–4749, 2017.

[142] Y. Li, P. Cao, W. Xia, J. Zhou, Y. Chu, W. Zhang, and J. Zhang, "Radar high-speed target range-doppler-azimuth coherent extension detection for autonomous vehicles," *IEEE Sensors Journal*, 2024.

[143] S. Zhu, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "Deepego+: Unsynchronized radar sensor fusion for robust vehicle ego-motion estimation," *IEEE Transactions on Radar Systems*, 2025.

[144] J. Lawrence, J. Bernal, and C. Witzgall, "A purely algebraic justification of the kabsch-umeyama algorithm," *Journal of research of the National Institute of Standards and Technology*, vol. 124, p. 1, 2019.

[145] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic segmentation on radar point clouds," in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 2179–2186, IEEE, 2018.

[146] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-view radar semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15671–15680, 2021.

[147] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE transactions on aerospace and electronic systems*, no. 4, pp. 608–621, 2007.

[148] O. Schumann, J. Lombacher, M. Hahn, C. Wöhler, and J. Dickmann, "Scene understanding with automotive radar," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 188–203, 2019.

[149] M. Zeller, J. Behley, M. Heidingsfeld, and C. Stachniss, "Gaussian radar transformer for semantic segmentation in noisy radar data," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 344–351, 2022.

[150] F. Fent, P. Bauerschmidt, and M. Lienkamp, "Radargnn: Transformation invariant graph neural network for radar-based perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 182–191, 2023.

[151] Z. Zhang, J. Liu, and G. Jiang, "Spatial and temporal awareness network for semantic segmentation on automotive radar point cloud," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3520–3530, 2023.

[152] Y. Wu, J. Liu, G. Jiang, W. Liu, and D. Orlando, "Mask-radarnet: Enhancing transformer with spatial-temporal semantic context for radar object detection in autonomous driving," *arXiv preprint arXiv:2412.15595*, 2024.

[153] Y. Zhang, L. Zhang, P. Pi, T. Li, Y. Chen, S. Peng, and Z. Ma, "Tarss-net: Temporal-aware radar semantic segmentation network," *Advances in Neural Information Processing Systems*, vol. 37, pp. 4906–4933, 2024.

[154] J. Liu, W. Xiong, L. Bai, Y. Xia, T. Huang, W. Ouyang, and B. Zhu, "Deep instance segmentation with automotive radar detection points," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 84–94, 2022.

[155] W. Xiong, J. Liu, Y. Xia, T. Huang, B. Zhu, and W. Xiang, "Contrastive learning for automotive mmwave radar detection points based instance segmentation," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1255–1261, IEEE, 2022.

[156] M. Zeller, V. S. Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss, "Radar instance transformer: Reliable moving instance segmentation in sparse radar point clouds," *IEEE Transactions on Robotics*, vol. 40, pp. 2357–2372, 2023.

[157] M. Zeller, D. C. Herraez, B. Ayan, J. Behley, M. Heidingsfeld, and C. Stachniss, "Semrafiner: Panoptic segmentation in sparse and noisy radar point clouds," *IEEE Robotics and Automation Letters*, 2024.

[158] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze, and F. Küçükay, "High resolution radar-based occupancy grid mapping and free space detection.," in *VEHITS*, pp. 70–81, 2018.

[159] T. Giese, J. Klappstein, J. Dickmann, and C. Wöhler, "Road course estimation using deep learning on radar data," in *2017 18th International Radar Symposium (IRS)*, pp. 1–7, IEEE, 2017.

[160] 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: Results of the 7th International Conference*, pp. 151–161, Springer, 2010.

[161] M. Zeller, V. S. Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss, "Radar velocity transformer: Single-scan moving object segmentation in noisy radar point clouds," *arXiv preprint arXiv:2507.03463*, 2025.

[162] M. Zeller, V. Singh Sandhu, B. Mersch, J. Behley, M. Heidingsfeld, and C. Stachniss, "Dataset for moving instance segmentation based on radarscenes," Nov. 2023.

[163] H. Reichert, B. Serfling, E. Schüssler, K. Turacan, K. Doll, and B. Sick, "Real time semantic segmentation of high resolution automotive lidar scans," *arXiv preprint arXiv:2504.21602*, 2025.

[164] K. Zhang, Y. An, Y. Cui, and H. Dong, "Semantic segmentation of 3d point clouds in outdoor environments based on local dual-enhancement," *Applied Sciences*, vol. 14, no. 5, p. 1777, 2024.

[165] Z. Pan, F. Ding, H. Zhong, and C. X. Lu, "Ratrack: moving object detection and tracking with 4d radar point cloud," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4480–4487, IEEE, 2024.

[166] Y. Li, Y. Liu, Y. Wang, Y. Lin, and W. Shen, "The millimeter-wave radar slam assisted by the rcs feature of the target and imu," *Sensors*, vol. 20, no. 18, p. 5421, 2020.

[167] N. Petrov, O. Krasnov, and A. G. Yarovoy, "Auto-calibration of automotive radars in operational mode using simultaneous localisation and mapping," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2062–2075, 2021.

[168] M. P. Ronecker, X. Diaz, M. Karner, and D. Watzenig, "Deep learning-driven state correction: A hybrid architecture for radar-based dynamic occupancy grid mapping," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2184–2191, IEEE, 2024.

[169] F. Xu, H. Wang, B. Hu, and M. Ren, "Road boundaries detection based on modified occupancy grid map using millimeter-wave radar," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1496–1503, 2020.

[170] A. Pearce, J. A. Zhang, R. Xu, and K. Wu, "Multi-object tracking with mmwave radar: A review," *Electronics*, vol. 12, no. 2, p. 308, 2023.

[171] S. Zhu, F. Fioranelli, A. Yarovoy, S. Ravindran, and L. Chen, "Hierarchical architecture and feature mixing for ego-motion estimation using automotive radar," in *ICMIM 2024; 7th IEEE MTT Conference*, pp. 99–102, VDE, 2024.

[172] A. N. Ramesh, C. M. León, J. C. Zafra, S. Brüggenwirth, and M. A. González-Huici, "Landmark-based radar slam for autonomous driving," in *2021 21st International Radar Symposium (IRS)*, pp. 1–10, IEEE, 2021.

[173] X. Cao, C. Zhu, and W. Yi, "Phd filter based traffic target tracking framework with fmcw radar," in *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 468–475, IEEE, 2022.

[174] A. Laribi, M. Hahn, J. Dickmann, and C. Waldschmidt, "A new height-estimation method using fmcw radar doppler beam sharpening," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 1932–1396, IEEE, 2017.

[175] B. Sun, I. Roldan, and F. Fioranelli, "Automatic labelling & semantic segmentation with 4d radar tensors," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2025.

[176] I. Roldan, A. Palffy, J. F. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "A deep automotive radar detector using the radelft dataset," *IEEE Transactions on Radar Systems*, 2024.

Acknowledgements

How do you write an acknowledgment? When I first asked this question, I posed it to ChatGPT, the very tool that has accompanied me through countless days and nights of learning, writing, and overthinking. As always, it answered my question calmly: "Start with gratitude, be honest, and speak from the heart." And so here I am, trying to do exactly that.

For years, I thought acknowledgments were only about thanking others: the supervisors who guided me, the colleagues who helped me, the committee members who generously invested their time to improve this work, the friends who stood by me through the brightest and darkest days, and my parents who raised me with love and supported my decision to go abroad even though they knew how much they would miss me. And of course, I thank all of them with deep sincerity. But today, I also want to thank someone I have never thanked before: myself. Not just one version of me, but the parts of myself that carried me through this long journey. These parts lived inside me quietly, often unnoticed, yet they shaped who I am today. I want to finally honor them.

The first part of myself I want to honor is the brave one: the version of me who has always dared to do what I wanted to do. It was this courage that led me to spend three years competing in electronics design simply because I loved it, even though my bachelor's major had little to do with it. It was this same courage that made me give up a comfortable, well-paid job and choose to study abroad at a time when everyone around me expected me to settle down. It was also this courage that made me give up my previous citizenship, despite having no support from my family and facing threats of losing them entirely. I thank this brave self for giving me independence, critical thinking, and the strength to pursue the life I want. Without it, I would not be standing here today. Life is all about experience and decisions, and as long as I make mine with courage, I will never regret them.

The second part of myself I want to honor is the one who is free: free from the weight of other people's expectations and from the pressure to live according to what society thinks I should be. This does not mean that I am undisciplined or impolite. It simply means that I know who I am: what I am good at, what I am not, what matters to me, and what I can let go of. To give a few examples, I care less about how much money I earn or how wealthy I might become, as long as I am working on things I truly love and surrounded by people I truly appreciate. I care less about whether a journey ends beautifully, as long as I tried my best, enjoyed the process, and learned something valuable along the way. This part of me has allowed me to live with fewer constraints and fewer burdens. I have always believed that other people's opinions should not define who I am; they can be heard, but they do not need to be obeyed. This free self has given me the clarity to walk through life with honesty, intention, and lightness.

The third part of myself I want to honor is the one who accepts. Over time, I have learned to accept that society is divided, that not everyone will like me, and that some people may be shaped by their experiences in ways that make them biased, unkind, or even racist. I have

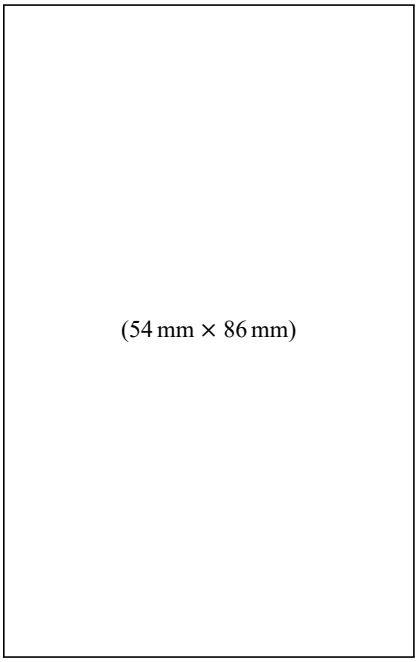
accepted that I cannot change their behavior or opinions, but I can choose not to let their judgments shape who I am. I have also learned to accept my own imperfections: that I am not a perfect person or always a perfect friend, that even after completing a PhD degree I sometimes feel I know nothing, that I am not always honest or selfless, and that I sometimes exaggerate. I accept these parts of myself, not as excuses, but as starting points for growth. And finally, I accept that I do miss my family and my friends, even though I rarely show my emotions. This quiet longing is part of me too, and I will carry it as I continue to pursue my life and my dreams. I am thankful for this part of myself, as acceptance has taught me to move forward with honesty, humility, and an open heart.

Lastly, I want to thank you. I value my time deeply, and in the same way, I value yours. Thank you for spending a few minutes of your life getting to know another side of this person, the side he does not always feel comfortable showing. I hope that, in some way, his story may inspire you too.

The Hague, November 2025
Written on a sleepless night before my ramen party
Simin Zhu

About the Author

Simin Zhu received his B.Sc. degree in Electrical Engineering and Automation from Central South University, China, in 2016. He subsequently worked as a hardware engineer at Huawei Technologies Co., Ltd. from 2016 to 2018. In 2019, he joined Delft University of Technology (TU Delft), the Netherlands, where he obtained his M.Sc. degree in Electrical Engineering in 2021. Since December 2021, he has been pursuing the PhD degree with the Microwave Sensing, Signals and Systems (MS3) group at TU Delft. His research interests include radar signal processing, deep learning for perception, sensor fusion, and automotive radar applications. He is the author of several publications in leading IEEE journals and conferences and the co-inventor of two patents on radar-based ego-motion estimation.



List of Publications

Journal Papers

4. **S. Zhu**, S. Ravindran, A. Yarovoy and F. Fioranelli, "Redefining Radar Segmentation: Simultaneous Static-Moving Segmentation and Ego-Motion Estimation Using Radar Point Clouds," in IEEE Transactions on Radar Systems (under review).

3. **S. Zhu**, S. Ravindran, C. Lihui, A. Yarovoy, and F. Fioranelli, "Radar Mounting Angle Estimation in Operational Driving Conditions," in IEEE Transactions on Radar Systems (under review).

2. **S. Zhu**, S. Ravindran, L. Chen, A. Yarovoy, and F. Fioranelli, "DeepEgo+: Unsynchronized Radar Sensor Fusion for Robust Vehicle Ego-Motion Estimation," in IEEE Transactions on Radar Systems, vol. 3, pp. 483-497, 2025.

1. **S. Zhu**, A. Yarovoy and F. Fioranelli, "DeepEgo: Deep Instantaneous Ego-Motion Estimation Using Automotive Radar," in IEEE Transactions on Radar Systems, vol. 1, pp. 166-180, 2023.

Conference Papers

2. **S. Zhu**, F. Fioranelli, A. Yarovoy, S. Ravindran, and L. Chen, "Hierarchical Architecture and Feature Mixing for Ego-Motion Estimation using Automotive Radar," ICMIM 2024; 7th IEEE MTT Conference, Boppard, 2024, pp. 99-102.

1. **S. Zhu**, F. Fioranelli, and A. Yarovoy, "Radar-only Instantaneous Ego-motion Estimation Using Neural Networks," 2023 20th European Radar Conference (EuRAD), Berlin, Germany, 2023, pp. 201-204.

Book Chapters

1. I. Roldan Montero, **S. Zhu**, A. Yarovoy, and F. Fioranelli, "Enhancing Automotive Radar Sensing Through Deep Learning," in *Radar Machine Learning for Autonomous Driving*, Artech House, Inc. Norwood, MA, 2025. (accepted, forthcoming)

Patent

2. S. Zhu, A. Yarovoy, F. Fioranelli, S. Ravindran, L. Chen, "Methods and devices for multi-radar, multi-frame ego-motion estimation," application number: PCT/US2024/031392, filed in 2024, accessible as WO2025250125A1.

1. **S. Zhu,** A. Yarovoy, F. Fioranelli, S. Ravindran, "An apparatus for determining ego-motion," in a US Patent Application, patent filed in 2023, WO2024183926A1.

Publications Beyond This Thesis

3. **S. Zhu**, R. G. Guendel, A. Yarovoy, and F. Fioranelli, “Continuous Human Activity Recognition with Distributed Radar Sensor Networks and CNN–RNN Architectures,” in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-15, 2022.

2. S. Yuan, **S. Zhu**, F. Fioranelli, and A. Yarovoy, “3-D Ego-Motion Estimation Using Multi-Channel FMCW Radar,” in IEEE Transactions on Radar Systems, vol. 1, pp. 368-381, 2023.

1. F. Fioranelli, **S. Zhu**, and I. Roldan, “Benchmarking Classification Algorithms for Radar-Based Human Activity Recognition,” in IEEE Aerospace and Electronic Systems Magazine, vol. 37, no. 12, pp. 37-40, 1 Dec. 2022.