Human Remote Sensing In Long-Term Care Facility Using Low-Cost FMCW Radar

by

Huy Trinh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2026

© Huy Trinh 2026

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contribution

This thesis consists in part of material that I authored or co-authored for publication. The contributions of all authors and collaborators are described below. Except where otherwise stated, I was responsible for preparing and integrating the thesis material, including the research work, data analysis, writing, editing, figure preparation, and final revisions under the supervision of Dr. George Shaker and Dr. Elliot Creager.

Research presented in Chapters 3 and 4:

The research presented in Chapters 3 and 4 was primarily conducted by me under the supervision of Dr. George Shaker and Dr. Elliot Creager. I was the main author of this work and was responsible for the research development, radar measurements, experimental implementation, data processing, data analysis, manuscript writing, editing, and integration of the material into this thesis. Phuong T. Thai contributed to proofreading, editing, and writing support for the introduction of the post-fall floor-occupancy detection manuscript. Dr. George Shaker and Dr. Elliot Creager provided research supervision, technical guidance, manuscript feedback, and intellectual input throughout the work.

Citations:

- **Chapter 3** is adapted from:
  - Trinh, H., Creager, E., & Shaker, G. (2026). Doppler-Domain Warping and Resampling for Semi-Static Human Occupancy Detection Using Low-Resolution SIMO FMCW Radar. **Accepted**.
  - Trinh, H., Thai, P.T., Creager, E., & Shaker, G. (2026). Reliable Quasi-Static Post-Fall Floor-Occupancy Detection Using Low-Cost Millimetre-Wave Radar. **Accepted**.
- **Chapter 4** is adapted from:
  - Trinh, H., & Shaker, G. (2026). Lightweight Range-Angle Imaging Based Algorithm for Quasi-Static Human Detection on Low-Cost FMCW Radar. **Accepted**.

Research presented in Chapter 5:

The research presented in Chapter 5 includes material from co-authored manuscripts on physics-informed digital-twin modelling and radar-based fall detection. For the physics-informed digital-twin framework manuscript, I was the main author and contributed to the research concept, experimental analysis, data processing, manuscript writing, editing, and integration of the material into this thesis. Dr. George Shaker and Dr. Elliot Creager provided supervision, technical guidance, and manuscript feedback.

For the radar-based fall detection digital-twin case study, Sebastian Ratto was the main author and contributed to the study design, measurement design, conceptual development, and manuscript writing. I contributed to data analysis, experimental implementation, manuscript proofreading, editing, and integration of the material into this thesis. Dr. Ahmed N. Sayed contributed to the literature review, manuscript editing, and research ideas. Khairy Elbadrawy contributed to the main idea development and preliminary simulation of fall scenarios. Andrew P. Sligar contributed as an industry partner through technical discussions and software-related guidance. Dr. George Shaker provided research supervision, technical guidance, and manuscript feedback.

Citations:

- **Chapter 5** is adapted from:
  - Trinh, H., Ratto, S., Creager, E., & Shaker, G. (2026). A Physics-Informed Digital Twin Framework for Calibrated Sim-to-Real FMCW Radar Occupancy Estimation. **Accepted**.
  - Ratto, S., Trinh, H., Sayed, A.N., Elbadrawy, K., Sligar, A.P., & Shaker, G. (2026). Radar-Based Fall Detection for Assisted Living: A Digital-Twin Representation Case Study. **Accepted**.

Overall, I was responsible for organizing the co-authored research material into a coherent thesis document, writing and revising the thesis text, preparing figures and tables, conducting the main analyses for the chapters where I served as lead author, and addressing feedback from my supervisors and examiners.

Abstract

This thesis studies privacy-preserving human remote sensing in Long Term Care (LTC) environments using low-cost 60 GHz FMCW radar. The main challenge we are addressing is the reliable sensing of weak or quasi-static human states in LTC environments, such as quiet occupancy, prolonged sitting or lying, and post-fall floor presence, which are clinically and operationally important yet difficult to detect with conventional methods. These scenarios are precisely the cases in which sensing reliability is most critical for resident safety, caregiver response, and building operation. The thesis, therefore, investigates how far a low-cost, low-resolution radar platform can be pushed through signal processing, machine learning methods, and simulation-driven data generation to deliver useful room-level awareness without relying on cameras or wearables. The work is organized around a practical deployment view rather than a single algorithmic contribution. It begins with the sensing hardware and baseline signal-processing chain, then develops methods for quasi-static occupancy detection and post-fall floor-occupancy detection, extends these ideas to imaging-style radar representations suitable for edge deployment, and finally studies simulation and digital-twin approaches for sim-to-real radar learning. Across these working stages, the unifying theme is the design of scalable, privacy-preserving, and energy-aware radar sensing methods for ambient assisted-living environments.

Acknowledgements

This Master's thesis was conducted at the Wireless Sensors and Devices Laboratory (WSDL), led by Professor George Shaker, and the Socially Embedded Machine Intelligence (SEMI) Lab, led by Professor Elliot Creager. During this period, I was also fortunate to serve as a Faculty Affiliate Researcher at the Vector Institute.

I would like to express my deepest gratitude to my supervisors, Professor George Shaker and Professor Elliot Creager, for their continuous guidance, encouragement, and support throughout my Master's journey. I am especially grateful to Professor George Shaker for giving me the opportunity to begin this work and to grow as a researcher. Through his mentorship, I was able to develop knowledge and research experience in radar sensing, digital twin systems, and machine learning. I am equally thankful to Professor Elliot Creager for his valuable feedback, support for this work and for giving me the chance to attend the SEMI lab. This gives me other interesting perspectives, which have greatly enriched this work.

I would also like to sincerely thank my friends, colleagues, and collaborators who supported me throughout this journey. I am especially grateful to Ms. Bun, Ms. Ca, Bong, and Mango for their kindness and support when I first arrived in Canada and for helping me feel at home during an important transition in my life. I am also thankful to Phong Trinh, Thong Nguyen, and Amirbahador Mansoori for their friendship, encouragement, and willingness to help whenever I needed it. My sincere appreciation also goes to Dr. Hajar Abedi, Ahmad Ansariyan, Natalie Thai, Sebastian Ratto V, and Mohammad Abuyaghi, whose support, discussions, and contributions helped strengthen and polish this thesis. In addition, I would like to acknowledge Rushabh Solanki and Rebecca Ma for their constant encouragement, thoughtful conversations, and support from the very beginning of this journey. Your presence and willingness to listen meant a great deal to me.

Last but not least, I would like to express my deepest gratitude to my mother, my father, and my younger sister Na for their unconditional love, patience, and unwavering support throughout my life and academic journey.

Finally, I would like to thank everyone, whether mentioned here by name or not, who in some way contributed to this milestone and made this thesis possible.

Dedication

This is dedicated to the memory of my paternal grandfather and maternal grandparents.

Table of Contents

Author's Declaration ii

Statement of Contribution iii

Abstract v

Acknowledgements vi

Dedication vii

List of Figures xii

List of Tables xxi

List of Abbreviations xxiii

1 Introduction 1

1.0.1 Motivation 1

1.0.2 Objective 2

1.0.3 Thesis Outline 3

2 Background Information 4

2.1 Frequency Modulated Continuous Wave Radar 4

2.1.1 BGT60TR13C . . . 4
2.2 Angle-of-Arrival Estimation . . . 8
2.2.1 Digital Beamforming (DBF) from Infineon . . . 8
2.2.2 Capon/MVDR Beamforming . . . 10
2.3 Constant False Alarm Rate . . . 12
2.3.1 Cell-Averaging Constant False Alarm Rate (CA-CFAR) . . . 12
2.3.2 Order-Statistic Constant False Alarm Rate (OS-CFAR) . . . 13
2.4 Deep Learning . . . 13
2.4.1 Perceptron . . . 14
2.4.2 Activation Functions . . . 15
2.4.3 Feed-Forward Neural Networks and Training . . . 16
2.4.4 Convolutional Neural Network (CNN) . . . 18
2.4.5 Recurrent Neural Networks (RNN) . . . 21
2.4.6 Long Short-Term Memory (LSTM) . . . 22
2.4.7 Bidirectional LSTM . . . 23

3 Quasi-static Activities, Occupancy And Post-fall Detections 25

3.1 Doppler-Domain Warping and Resampling for Semi-Static Human Occupancy Detection . . . 25
3.1.1 Related Work . . . 25
3.1.2 Dataset Collection . . . 27
3.1.3 Proposed RASSO Algorithm . . . 30
3.1.4 Signal-to-Noise Ratio Analysis of Baseline and RASSO methods . . . 35
3.1.5 Digital Beam Forming comparison . . . 39
3.1.6 Benchmarking of RASSO against E-RESPD . . . 41
3.1.7 Cross-subject Evaluation . . . 44
3.1.8 Data-Driven Classification . . . 46
3.1.9 Uncertainty Quantification (CNN-LSTM, Subject 2-4) . . . 48

3.2 Post-Fall Floor-Occupancy Detection . . . 51
3.2.1 Experiment Setup . . . 52
3.2.2 Results . . . 54
3.3 Conclusion . . . 60

4 Imaging-Based Radar Representation Processing for Occupancy Detection at the Edge 61
4.1 Chapter Motivation and Scope . . . 61
4.2 Radar Image Preprocessing Evaluation . . . 62
4.2.1 Adaptive Preprocessing . . . 62
4.2.2 Adaptive-Thresholding . . . 63
4.2.3 Entropy-Based Denoising . . . 63
4.2.4 Experiment Setup . . . 64
4.2.5 Results . . . 64
4.3 Proposed Imaging-Based Algorithms . . . 67
4.3.1 Experiment Setup . . . 68
4.3.2 Proposed Quality Scoring Algorithm . . . 68
4.3.3 Temporal Tracking Algorithm . . . 71
4.3.4 Lightweight Algorithm for Quasi-Static Detection . . . 72
4.3.5 Results . . . 74
4.4 Conclusion . . . 77

5 Digital-Twin Simulation and Calibrated Sim-to-Real Radar Sensing for Occupancy and Fall Analysis 79
5.1 Chapter Motivation and Scope . . . 79
5.2 Digital Twin Framework for Radar Occupancy Estimation . . . 80
5.2.1 Experiment Setup . . . 82
5.2.2 Noise-Floor Domain Randomization and CDR . . . 83
5.2.3 Results . . . 88

5.2.4 Conclusion . . . 91
5.3 Digital-Twin Representation for Fall Detection . . . 91
5.3.1 Digital-Twin Simulation Setup . . . 92
5.3.2 Dataset Construction . . . 95
5.3.3 Processing Method . . . 96
5.3.4 Results . . . 99
5.3.5 Conclusion and Discussion . . . 100

6 Conclusion **104**
6.1 Summary . . . 104
6.2 Discussion . . . 105
6.3 Ongoing Research and Future Work . . . 107
6.3.1 Minew Radar-Based Occupancy Detection and People Counting . . . 107
6.3.2 Multi-People Close-Interaction Based Localization . . . 110
6.3.3 Radar-Based Sim2Real Fall Detectxion . . . 112
6.4 Publications . . . 114

References **116**

List of Figures

2.1 Infineon XENSIV™ BGT60TR13C radar next to 1 Euro coin. In all experiments, the sensor module is mounted on the standard Infineon evaluation board as shown in Figure 2.3. 4

2.2 BGT60TR13C radar sensor block diagram. The signal sensed by the three receiver channels (RX1, RX2, and RX3) is mixed with the transmitted signal from TX1, processed, and then converted digitally through the ADC. . . . 5

2.3 Our baseline signal processing pipeline (green-dash). 7

2.4 BGT60TR13C radar package outline. 9

2.5 2D CA-CFAR. 12

2.6 Illustration of a perceptron. The inputs $x_1, \ldots, x_N$ are weighted by $w_1, \ldots, w_N$, summed with bias b, and passed through an activation function $\sigma(\cdot)$ to produce the neuron output. 15

2.7 Common activation functions used in neural networks, including ReLU, hyperbolic tangent, and sigmoid. 15

2.8 Example of a feed-forward neural network. Information flows from the input layer through hidden layers to the output layer without recurrence. 17

2.9 Example of the convolution operation in a Convolutional Neural Network (CNN). A local kernel is applied across the input feature map to generate an output activation map 18

2.10 Illustration of average pooling and max pooling. 19

2.11 Conceptual diagram of batch normalization. Intermediate activations are normalized using the mini-batch mean and variance, then rescaled and shifted by learnable parameters to stabilize network training. 20

2.12 Basic RNN architecture. 22
2.13 Architecture of the LSTM network. 23
2.14 Architecture of the biLSTM network. 24

3.1 Testing setups for data collection in the MIRADA apartment at the Schlegel–UW Research Institute for Aging, operated by the Wireless Sensors and Devices Laboratory. The yellow 2 m arrow refers only to the approximate sensor-to-monitored-floor separation visible in this illustrative figure. The actual mounting height used in all recordings is $\approx$ 2.5 m and tilted 30° downward. 27
3.2 Experimental setup and viewpoints. Two wall-mounted radars (TV-side and window-side) observe quasi-static occupancy from different walls. The five annotated floor locations (P1–P5) indicate the positions where the subject lies or sits during each trial, covering near-wall, near-furniture, and open-floor regions. 28
3.3 Few measurement activities: Standing (a), Lay on Sofa (b), Sitting (c), Lay on Floor (d). 29
3.4 Our baseline signal processing pipeline (green-dash) and our proposed RASSO (red-dash box) 30
3.5 RASSO forward and inverse Doppler warp for edge/corner frequency f_c ($\equiv f_e$) corresponding to (3.2)–(3.3). Note that the corner frequency f_c shown is identical to the edge frequency f_e. Top: forward (left) and inverse (right) mapping between linear Doppler f and warped coordinate D. Bottom: uniform sampling in D (blue) induces denser sampling near $f = 0$ (zoomed view). 32
3.6 Range–Azimuth feature maps (frame 100) obtained from a Doppler-averaged Capon beamformer: (left) Conventional Range-Azimuth (RA), (right) Range-Angle based Semi-Static Occupancy (RASSO)-enhanced RA. Both plots show power integrated over all Moving Target Indicator (MTI)-filtered Doppler bins. 33
3.7 Range-Azimuth feature maps of 2-minute data collected when the subject was lying on the floor (a) before applying RASSO and (b) after applying RASSO. The RASSO-processed maps in (b) exhibit noticeably reduced background clutter and enhanced target energy, indicating an improved signal-to-noise ratio that supports more robust recognition. 35

3.8 Conventional and RASSO-enhanced Range-Azimuth maps (frame 360) for a representative "Lay-On-Floor" frame (Subject 1). **Top row**: baseline RA before (left) and after CA-CFAR (right). **Bottom row**: RASSO RA before (left) and after CA-CFAR (right). All maps use normalized intensity, and the green dashed box marks the evaluation ROI around the lying subject. . 36

3.9 Effect of the MTI forgetting factor α on detection performance and CA-CFAR sensitivity on the development set (Subject 1). The curves show the true-positive rate (TPR), false-alarm rate (FAR), and the re-tuned CA-CFAR scaling k_{CFAR} (right axis) as α is swept; k in the legend corresponds to k_{CFAR}. The results show that MTI must be enabled and that performance remains nearly unchanged across a broad range of small nonzero α values, while larger α values reduce detection performance. The main takeaway is that performance is broadly stable for small nonzero α, whereas overly large α degrades performance; accordingly, $\alpha = 0.01$ is selected as a representative operating point. Based on this stable regime, $\alpha = 0.01$ is used in the remainder of the experiments. 37

3.10 Selection of the Doppler edge frequency f_e for the RASSO mapping. The mean z-score SNR inside the bounding box (Eq. (3.8)) is plotted versus f_e; the elbow around $f_e = 0.03$ Hz (dashed line) is chosen as the operating point in all experiments. 38

3.11 False–alarm rate (FAR) on empty-room sequences as a function of the CA–CFAR sensitivity k_{CFAR} for the three processors: (**left**) SDK Digital Beamforming (DBF), (**middle**) our Conventional baseline, and (**right**) our RASSO-enhanced pipeline. Curves are computed on negatives only (no person present), after MTI (α=0.01), five-frame mean smoothing, $[0, 1]$ normalization, and CA–CFAR with guard G=4 and train T=10. Dots indicate individual sequence measurements; the solid line is the mean FAR. The red vertical dashed line marks the operating point chosen for each method (smallest k_{CFAR} at which mean FAR meets the target constraint used throughout the paper). 39

3.12 Frame-wise z-score SNR for the Lay-On-Floor sequence (Subject 1), comparing conventional RA and RASSO RA over a 30–60 s interval. RASSO RA yields a consistently higher SNR, indicating stronger target concentration and reduced background variance. 40

3.13 Balanced Accuracy (BA) at matched-FAR operating points for SDK DBF, our Conventional baseline, and our RASSO-enhanced processor. For fairness, the CA–CFAR sensitivity k_{CFAR} for each method is tuned on empty-room sequences only to meet the same FAR target as Figure 3.11. 41

3.14 (**Left Column**) Range-Azimuth feature map (frame 80) of the subject lying on the floor obtained by (**Top**) Digital Beam Forming (DBF), (**Middle**) Our Conventional, (**Bottom**) RASSO enhanced on Our Conventional. The right column shows their corresponding CFAR detection. 42

3.15 Visualization of 6 feature maps on first row and their corresponding CFAR detections on second row (from left to right): Macro E-RESPD, Micro E-RESPD, Baseline Range Doppler, RASSO-enhanced Range-Doppler, Baseline Range-Azimuth, RASSO-enhanced Range-Azimuth. 42

3.16 Recall–FAR characteristics for five RA/RD feature maps: HOOD Macro E-RESPD, HOOD Micro E-RESPD, Baseline Range–Doppler, Baseline Range–Azimuth, and RASSO Range–Azimuth. For each feature map, we sweep the CA–CFAR sensitivity parameter k_{CFAR} on the development subject and, for every value of k_{CFAR}, compute the corresponding recall and false–alarm rate (FAR). The legend reports the area under the Recall–FAR curve (AUC). Each curve corresponds to a single feature map, and different points along the same curve are obtained for different values of k_{CFAR}. Light blue vertical dashed lines indicate reference operating regions at FAR = 1% and 5% used for pointwise comparisons. The recall values at these FAR levels are reported in Table 3.2. 43

3.17 Architecture of models (top): SimpleCNN and (bottom) CNN-LSTM. . . . 46

3.18 Macro-F1 bootstrap replicates (hybrid beeswarm) with 95% CIs for Subjects 2–4. Each dot is one session-level bootstrap replicate. The solid dot and whiskers are the full-set estimate and its 95% CI. Orange (RASSO-Enhanced) consistently exceeds blue (Baseline). 49

3.19 Per-class uncertainty. Bootstrap join scatter of $F1_{\mathrm{empty}}$(x) vs. $F1_{\mathrm{person}}$(y). The grey diagonal marks equal per-class performance. 50

3.20 Paired improvement distributions. Histograms of Δ = Macro-$\mathrm{F1}_{\mathrm{RASSO}}$ − Macro-$\mathrm{F1}_{\mathrm{Baseline}}$ with median (solid green), 95% CI (shaded band), and a red dashed line at zero. 50

3.21 Placements of pressure ulcers. 51

3.22 Few example post-fall floor postures from the dataset. Subjects lie on the floor in multiple quasi-static poses (a) supine with arms extended, (b) side-prone, and (c) supine with knees bent. Furniture is intentionally placed near the subject to create multipath and partial occlusions. 52

3.23 CA-CFAR sensitivity tuning on Subject 1's dataset. Macro-F1, frame-level false positive rate (FPR), and true positive rate (TPR) are shown as the CA-CFAR scale factor k_{CFAR} is swept. The selected operating points k satisfy the FPR cap (0.1) while maximizing Macro-F1 for each method. 54

3.24 Qualitative comparison of DBF and our proposed detectors on frame 9 for subject 1 lying on the floor at position 1 (window view), which both of them produce detections within the ground-truth bounding box. 55

3.25 Qualitative comparison of DBF and our proposed detectors on frame 579 for subject 1 lying on the floor at position 1 (window view). Only our proposed method can detect within the ground-truth bounding box. 56

3.26 Paired per-trial comparison of frame-positive rate. Each point corresponds to one trial (subject, view, and location). Trials are sorted by improvement Δ. 57

3.27 Frame-positive rate by viewpoint. Boxplots compare the vendor DBF and proposed method for TV-side and Window-side views. 58

3.28 Reliability curve (coverage across thresholds). The y-axis shows the fraction of trials with frame-positive rate $\geq \tau$. 59

4.1 Spectrogram after adding 5 different White Gaussian noise levels. 64

4.2 Spectrogram results for Walking (W): original spectrogram (top-left), noisy spectrogram (top-right) and 4 methods under white Gaussian noise (WGN) (Signal-to-Noise Ratio (SNR) = 10 dB). 65

4.3 Spectrogram results for Walking (W): original spectrogram (top-left), noisy spectrogram (top-right) and 4 methods under WGN (SNR = -5 dB). . . . 66

4.4 Scores calculated from our proposed algorithm corresponding to the feature map. 70

4.5 Range-Azimuth and Range-Elevation feature maps of frame 51 (class 2) and their tracking results (columns 2 and 4). 71

4.6 End-to-end radar's imaging processing pipeline. Raw complex IQ frames from the 60 GHz FMCW radar are converted to range–Doppler maps via range and Doppler FFTs with DC removal, windowing, and MTI clutter suppression. Capon beamforming over two receive channels produces a normalized range–azimuth (RA) intensity map, which is then processed by three detectors: 2-D CA–CFAR, 2-D OS–CFAR, and the proposed percentile-gated lump detector. 72

4.7 Percentile ablation on a representative Range–Azimuth frame (*lying on sofa*). From left to right: normalized RA image and detection masks obtained by keeping only the top 90%, 95%, and 99% of pixel intensities. . . 74

4.8 Qualitative comparison of all three detectors on frame 21 for subject 1 lying on the floor, which all of them produce detections within the ground-truth bounding box. 75

4.9 Qualitative comparison of all three detectors on frame 106 for subject 1 lying on the floor. Only our proposed method can detect within the ground-truth bounding box. 76

4.10 Raspberry Pi 4B Board with BGT60TR13C radar. 76

5.1 Simulation setup showing the created radar scenes. A 3D Mixamo humanoid model is animated to represent walking motion and imported into the virtual corridor environment. The FMCW radar observes different scenarios—(left) a single walking subject and (right) two subjects walking at different speeds, producing corresponding range–Doppler signatures. 81

5.2 Experiment setup for 2 people walking scenario. 82

5.3 Overview of the proposed sim-to-real pipeline. **Yellow blocks:** Start of simulated data and real data preprocessing **Green path**: simulated radar frames from the digital twin are converted to range–Doppler (RD) maps and passed through the noise-floor CDR block, then clipped, normalized, and colour-mapped for network training. **Purple path**: CDR uses 10 seconds of unlabeled real empty-room data to estimate the target noise-floor statistics that drive the calibration. **Blue path**: real radar frames for evaluation are converted to RD maps and only clipped and normalized, without any randomization or CDR modification. 84

5.4 Effect of CDR on global RD magnitude statistics. The histogram compares naive simulation, real empty-room measurements, and CDR-processed simulation. The horizontal axis shows RD magnitude in dB, while the vertical axis uses logarithmic density to reveal both the dominant low-noise peak and lower-density clutter tails. Naive simulation is concentrated at unrealistically low background levels, whereas CDR shifts the simulated distribution toward the real empty-room statistics. 86

5.5 Range-Doppler maps for (a,d) baseline simulation, (b,e) simulation after CDR, and (c,f) real measurements for the empty room and one-person walking. 87

5.6 Static-to-total energy ratio in RD space. Each point is one frame. CDR moves simulated points toward the real cluster, indicating that the global clutter level has been calibrated. 87

5.7 Qualitative visualization of early feature maps from the first residual block of ResNet-18 for representative simulated and real radar range-Doppler inputs. (a,d) inputs; (b,e) features from the baseline RT model trained only on unmodified simulation; (c,f) features from the CDR-trained model. 88

5.8 Confusion matrices for real binary occupancy detection across 3 methods. . 90

5.9 Confusion matrices for real people counting across 3 methods. 90

5.10 Digital-twin simulation setup. (a) Top-down and side-view schematic of the 6×8 m furnished room, showing the ceiling-mounted FMCW radar, its field of view, and example reflecting objects in the modelled environment. (b) Perceive-EM renderings of one fall at three time instants. 92

5.11 Discrete grid over subject position and yaw relative to the radar, with one example fall trajectory shown. 94

5.12 Empirical amplitude histograms (in dB) over all pixels for the three per-RX Doppler–time spectrograms. The means are similar but not identical, suggesting that each receive patch sees a slightly different multipath power distribution rather than an exact copy of the same signal. 96

5.13 Example Doppler–time spectrograms at the fixed range bin $r^\star \approx 3.9$ m for a non-fall (a) and a fall (b) clip in the digital twin. 98

5.14 Test accuracy versus epoch for each representation mode. The spectrogram modes approach and stay near 99%, while the time-pooled RDM mode follows a lower curve with larger fluctuations over epochs. 99

5.15 Normalized confusion matrices on the balanced test set for (a) `spec`, (b) `spec3`, and (c) `rdm` (rows: true labels, columns: predicted). 100

5.16 Doppler–time spectrograms for one fall: (left) synthetic from the digital twin and (right) measured from a 60 GHz radar. Both show a roughly triangular Doppler–time pattern during descent, followed by near-stationary returns; this is a qualitative check on gross signal structure. 101

5.17 Pixel-wise amplitude histograms (log-scale density) for the synthetic and measured fall spectrograms in Figure 5.16. For this example, the measured data show a tighter distribution around the modal value, consistent with receiver processing that the idealized twin omits. 101

6.1 Overview of the multi-room Minew radar setup and synchronized visualization prototype. **Top-left**: GUI showing synchronized 2-D and 3-D views for the living room. **Top-right**: GUI view of all monitored rooms in the shared apartment coordinate frame. Bottom: ceiling-mounted Minew MSR01 radar nodes installed in the living room (left) and bathroom (right). 107

6.2 Demonstration of the multi-room synchronized radar visualization at Doors Open Waterloo Region 2025. Left: a simpler apartment state with two detected targets. Right: a more complex state with up to five simultaneously visualized targets across three rooms. The 2-D floor-plan view and 3-D room view are generated from the same room-registered radar coordinates. . . . 108

6.3 Prototype point-cloud visualization workflow for the Minew radar board. (A) Hardware connection through a TTL–USB serial adapter. (B) Prototype data-acquisition setup connected to a Raspberry Pi 4B. (C) Example Python-based 2-D and 3-D visualization of the received point cloud for a quasi-static human target. 110

6.4 Preliminary close-interaction localization experiments. (A) and (B) show two assisted-living interaction scenarios involving two people in proximity. (C) shows the corresponding radar-derived range–angle visualizations used to study whether the two nearby subjects can be separated under overlapping reflections and clutter. 111

6.5 Digital-twin setup for a representative fall case. A humanoid model performs a forward fall with bent knees in a living-room scene modelled after the Research Institute for Aging (RIA) environment, while a wall-mounted virtual radar captures the synthetic radar response. 112

6.6 Doppler–time spectrograms for a representative fall case in which the subject falls forward with bent knees. Left: measured real data. Middle: raw simulated data from the digital twin. Right: simulated data after calibrated domain randomization (CDR). 113

6.7 Effect of calibrated domain randomization on the global intensity statistics of the simulated fall spectrogram. The histogram compares the baseline ray-tracing simulation, the measured real target data, and the CDR-adjusted simulation in the dB domain. 113

List of Tables

3.1 Radar hardware and waveform configuration (BGT60TR13C). 27

3.2 Summary of AUC of the Recall–FAR curve and Recall at two fixed false-alarm operating points (FAR = 1% and 5%) among 5 methods. Best values per column are shown in **bold**. 44

3.3 CA-CFAR Evaluation Across Subjects (k_{CFAR} = 1.2 for Baseline Range-Azimuth and k_{CFAR} = 1.6 for RASSO-Enhanced). Best values for each F1 metric per subject are shown in **bold**. 45

3.4 RASSO Results – Frame Based (SimpleCNN) 47

3.5 RASSO Results – Sequence Based (CNN-LSTM) 48

3.6 Participant-level frame-positive rate by viewpoint and location (higher is better). Best values per view and location for each subject are shown in bold. 57

4.1 Evaluation metrics result for walking (W) under White Gaussian Noise (SNR = 10 dB). 66

4.2 Evaluation metrics result for walking (W) under White Gaussian Noise (SNR = -5 dB). 67

4.3 Detection accuracy (%) per Subject. Best values for each metric per Subject are shown in **bold**. 75

4.4 Average runtime analysis on Raspberry Pi 4 for 3 methods. Best values for each metric are shown in **bold**. 77

5.1 Simulated fall and non-fall motions in the digital-twin dataset. 95

5.2 Input dimensions and interpretation of the H and W axes for each radar mode (balanced dataset of 2,880 clips). 97

5.3 Test performance on the balanced synthetic test set ($N_{\text{test}} = 576$) after 100 epochs. 100

List of Abbreviations

ADC analog-to-digital converter 6, 7, 93

AIP antenna-in-package 5

AOA angle-of-arrival 5, 9, 11, 26, 34

API Application Programming Interface 93

APr Adaptive Resolution Preprocessing 62, 64–66

ATh Adaptive Thresholding Algorithm 63–65

AUC Area Under the Curve 41, 43, 44

BA Balanced Accuracy 40

BiLSTM Bidirectional LSTM 23, 24

CA-CFAR Cell Averaging Constant False Alarm Rate 10, 12, 13, 35, 41, 43, 53, 55, 56, 73, 75–77

CDR Calibrated Domain Randomization 82–86, 88–91, 113, 114

CFAR Constant False Alarm Rate 3, 26, 31, 36–39, 45, 47, 53, 60, 73, 104, 105

CI Confidence Interval 48, 49

CNN Convolutional Neural Network xii, 3, 18, 19, 21, 24, 26, 47, 48, 60, 83, 84, 91, 92, 97, 98, 100, 103, 104

CPI Coherent Processing Interval 93, 96, 97

CUT Cell–Under–Test 12, 13, 53

CW continuous wave 25

DBF Digital Beamforming 8, 10, 39, 40, 45, 53, 54, 56, 58–60

DC Direct Current 6, 8, 31, 37, 44, 82

DR Domain Randomization 83–85, 89

DT Digital Twins 91

EBD Entropy-Based Denoising 63, 65

EM Electromagnetic 79–81, 91–93, 100, 102

FAR False Alarm Rate 36–38, 40, 41, 43–45, 53, 60

FFT Fast Fourier Transform 6, 7, 10, 30, 31, 33, 34, 63, 82, 83, 93, 96

FIFO First In First Out 6

FMCW Frequency Modulated Continuous Wave 2, 3, 5, 6, 25, 28, 52, 61, 62, 73, 80, 82, 85, 92, 93, 96, 102, 104, 105

FN False Negatives 36

FNR False Negative Rate 53

FP False Positives 53

FPR False Positive Rate 53, 54

GAN Generative Adversarial Network 65

GUI Graphical User Interface 109

HAR Human Activity Recognition 62

IF intermediate-frequency 6

JSON JavaScript Object Notation 82, 108, 109

LiDAR Light Detection and Ranging 106

LSTM Long Short-Term Memory 3, 14, 22–24, 26, 48, 60

LTC Long Term Care v, 1–3, 29, 30, 51, 52, 60–62, 72, 77, 91, 104

MAE Mean Absolute Error 64

mD micro-Doppler 61–67

MIMO Multiple-Input Multiple-Output 26

mmWave Millimeter-wave 28, 52

MQTT Message Queuing Telemetry Transport 108, 109

MSE Mean Squared Error 64, 65

MTI Moving Target Indicator xiii, 6, 8, 10, 11, 31, 33, 34, 36–39

MVDR Minimum Variance Distortionless Response 8, 10, 11, 26, 44

OS-CFAR Order-Statistic Constant False Alarm Rate 13, 73, 75–77, 111

PCB Printed Circuit Board 110

PCC Pearson Correlation Coefficient 64

PLL phase-locked loop 6

PO Physical Optics 80

PSNR Peak Signal-to-Noise Ratio 64, 65

RA Range-Azimuth xiii, 10, 11, 33–36, 39, 41, 43–47, 55, 67–73, 78

RASSO Range-Angle based Semi-Static Occupancy xiii, 3, 6, 8, 10, 26, 30–35, 38–41, 43–49, 60, 104

RD Range-Doppler 31, 34, 41, 43, 44, 80, 81, 83–89, 91

RDC Range–Doppler Cube 7, 33

RDM Range-Doppler Map 7, 8, 99, 100, 103

RE Range-Elevation 11

ReLU Rectified Linear Unit 16, 47, 98

RIA Research Institute for Aging xix, 112

RMSE Root Mean Squared Error 64, 65

RNN Recurrent Neural Network 14, 21–23

ROC Receiver Operating Characteristic 41

ROI Region of Interest 53

RT Ray-Tracing 81, 83, 88, 89

SBR Shooting and Bouncing Rays 80, 93

SDK Software Development Kit 8, 10, 11, 39, 40

SIMO Single-Input and Multiple-Output 7

SNR Signal-to-Noise Ratio xvi, xxi, 35, 37–39, 46, 62, 64–67, 75, 77

SPI Serial Peripheral Interface 6

SSIM Structural Similarity Index Measure 64–66

STFT Short-Time Fourier Transform 63

SVM Support Vector Machine 63

TNR True Negative Rate 40, 54

TP True Positives 36

TPR True Positive Rate 36, 37, 40, 41, 53

TTL Transistor-Transistor Logic 109

USB Universal Serial Bus 109

UWB Ultra-Wideband 25

VCO voltage-controlled oscillator 6

WGN white Gaussian noise xvi, 64, 65

Chapter 1

Introduction

1.0.1 Motivation

Population ageing is reshaping health-care systems worldwide and is placing increasing pressure on LTC, retirement residences, hospitals, and home-based care. The scale of this shift is substantial: by 2030, one in six people globally will be aged 60 years or older, and by 2050 the world's population aged 60+ is projected to reach 2.1 billion [1]. Canada is experiencing the same demographic transition. Statistics Canada reports more than 8.1 million people aged 65 and older as of July 1, 2025 [2]. As the number of older adults rises, care systems are being asked to support safety, independence, and quality of life while facing persistent staffing and cost constraints [3]. Within this context, falls remain one of the most important and costly adverse events affecting older adults. In Canada, falls are the leading cause of injury-related hospitalizations and deaths among adults aged 65 years and older, and they can result in disability, increased caregiver burden, admission to long-term care, and reduced quality of life [4]. The burden is also economic: the direct cost of fall-related injuries among adults aged 65 and older was estimated at 5.6 billion dollars in 2018 [5]. More broadly, what matters operationally in assisted-living environments is not only the fall event itself, but also the possibility that a resident may remain on the floor, inactive or unobserved, for a prolonged period after the event. These realities motivate the need for sensing systems that can operate continuously and unobtrusively, especially in private indoor environments where residents spend much of their time.

Existing monitoring technologies each address part of this need, but each also has important limitations. Camera-based active and assisted living systems can provide rich scene information, yet privacy and dignity are major barriers to their acceptance, and bedrooms

and bathrooms are among the least acceptable locations for video-based monitoring [6, 7]. Wearable systems avoid visual surveillance, but their real-world usefulness depends on adherence, comfort, charging, proper placement, and the user's willingness and ability to keep the device on [8]. These limitations are especially problematic in frail, cognitively impaired, or post-fall situations, where the person may not be able to initiate help or manage a device reliably [9]. As a result, there is a clear need for a monitoring modality that is proactive rather than purely reactive, and that does not rely on either continuous video capture or continuous user compliance. In these settings, there is a growing need for continuous, unobtrusive monitoring of whether a resident is present, for how long, and in which part of the room—information that can support fall response, staff scheduling, and energy-efficient building control. Radar sensing is a promising candidate for this role due to the fact that it is contactless, non-visual, and can operate regardless of ambient lighting conditions. Radar technology has a long history of application in various sectors, including aerospace, defence, and automotive systems [10, 11]. More recently, radar has been explored for indoor monitoring in homes, hospitals, and LTC facilities [12–14]. For room-scale LTC monitoring, the sensing system must not only be accurate, but also affordable, energy efficient, and scalable across many rooms. This rules out an exclusive reliance on expensive high-resolution hardware and motivates the use of low-cost 60 GHz Frequency Modulated Continuous Wave (FMCW) radar platforms. However, such platforms offer limited angular resolution and often struggle in the very conditions that matter most clinically: weakly moving or quasi-static states such as quiet room occupancy, prolonged lying, and post-fall floor presence. The motivation of this thesis, therefore, is to investigate how far these low-cost radar systems can be pushed—through signal processing, learning-based models, and simulation-assisted data generation—to provide useful, privacy-preserving room-level awareness for long-term care deployment.

1.0.2 Objective

The objective of this thesis is to develop and evaluate privacy-preserving radar-based methods for indoor human remote sensing in long-term care environments using low-cost FMCW hardware. In particular, the thesis focuses on scenarios in which the target is weakly moving or quasi-static, since these are among the most challenging and most clinically relevant conditions for assisted-living monitoring. The work considers three closely related problems: quasi-static occupancy detection, post-fall floor-occupancy detection, and representation learning for radar-based occupancy and fall-related sensing.

To achieve this objective, the thesis pursues four technical goals. First, it establishes a practical sensing and processing baseline using the Infineon BGT60TR13C radar, including

range–Doppler generation, clutter suppression, beamforming, and Constant False Alarm Rate (CFAR)-based detection. Second, it develops signal-processing methods that improve sensitivity to weak micro-motions and floor-level occupancy under cluttered indoor conditions, most notably through Doppler-domain warping and resampling for semi-static occupancy detection. Third, it studies radar representations and lightweight data-driven models suitable for edge deployment, including CNN- and CNN–Long Short-Term Memory (LSTM)-based processing of radar-derived maps. Fourth, it investigates digital-twin and sim-to-real strategies for radar occupancy estimation and fall-related sensing, with the goal of reducing the dependence on large labelled real-world datasets.

Overall, the thesis aims to show that clinically meaningful room-level awareness can be obtained from low-cost radar systems when hardware-aware signal processing, learning-based classification, and simulation-based data generation are designed together. The broader goal is not only improved detection accuracy, but a sensing framework that is privacy-preserving, deployment-oriented, and extensible to future LTC monitoring tasks such as people counting, close-interaction localization, and sim-to-real fall detection.

1.0.3 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 reviews the background required for the thesis, including the FMCW radar hardware used in this work, dataset collection, angle-of-arrival estimation, CFAR-based detection, and the neural-network models referenced later
- Chapter 3 presents methods for quasi-static activity occupancy and post-fall floor-occupancy detection, including the proposed RASSO processing, its evaluation against baselines, and the post-fall study in a furnished LTC-like environment.
- Chapter 4 shifts to imaging-based radar representation processing, where radar spectrograms and range–angle maps are treated as image-like inputs for denoising, scoring, tracking, and lightweight edge detection.
- Chapter 5 studies simulation and digital-twin frameworks for radar occupancy estimation and fall-related sensing, including calibrated sim-to-real learning and synthetic fall data generation.
- Chapter 6 summarizes the main findings, discusses ongoing extensions of the work, and outlines future research directions.

Chapter 2

Background Information

2.1 Frequency Modulated Continuous Wave Radar

2.1.1 BGT60TR13C

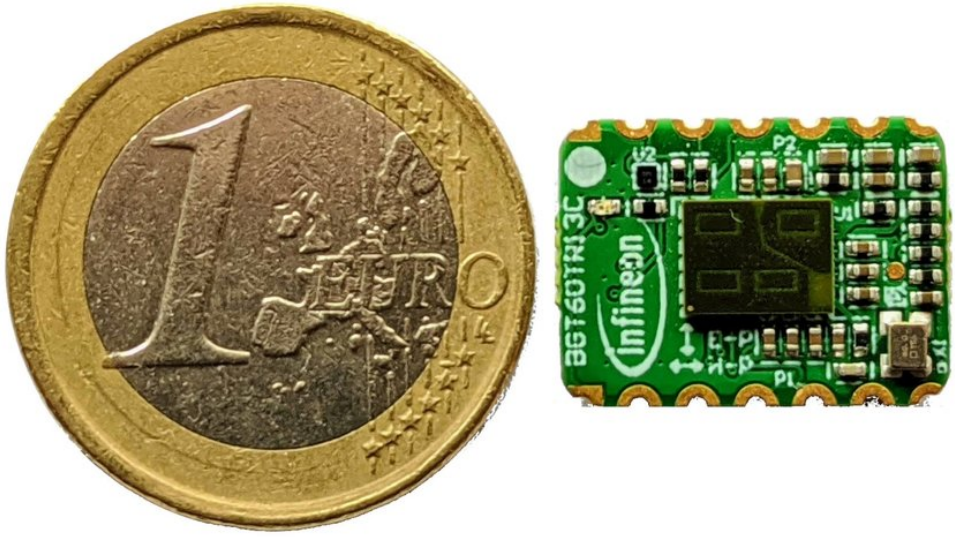

Figure 2.1: Infineon XENSIV™ BGT60TR13C radar [15] next to 1 Euro coin. In all experiments, the sensor module is mounted on the standard Infineon evaluation board as shown in Figure 2.3.

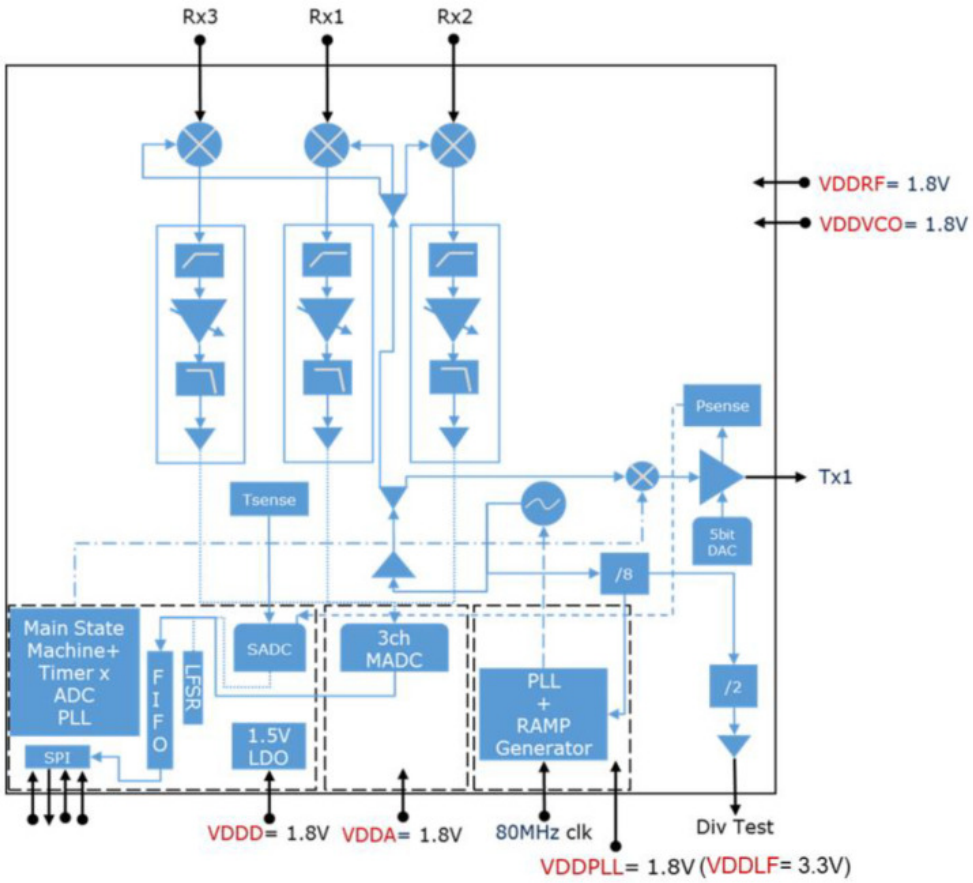

Figure 2.2: BGT60TR13C radar sensor block diagram [16]. The signal sensed by the three receiver channels (RX1, RX2, and RX3) is mixed with the transmitted signal from TX1, processed, and then converted digitally through the ADC.

The sensing module used in this work is the Infineon XENSIV^{TM} BGT60TR13C radar module as shown in Figure 2.1, a compact 60 GHz FMCW radar sensor designed for short-range high-resolution sensing applications. The sensor integrates an antenna-in-package (AIP) structure operating in the 58–63.5 GHz millimeter-wave band and supports an ultra-wide FMCW bandwidth of up to approximately 5.5 GHz, enabling fine range resolution and sensitivity to small motions [15]. The radar front-end contains one transmit (TX) antenna and three receive (RX) antennas, allowing the acquisition of spatial information that can later be used for angle-of-arrival (AOA) estimation and beamforming [17]. As shown in Figure 2.2, the BGT60TR13C internally integrates several functional blocks including

a voltage-controlled oscillator (VCO), phase-locked loop (PLL), ramp generator, analog front-end, and. The PLL and ramp generator produce the frequency-modulated chirp signal that is transmitted by the TX antenna. Reflected echoes from targets are received by the three RX channels, amplified, mixed with the transmitted reference signal, analog-to-digital converter (ADC) and converted to baseband intermediate-frequency (IF) signals before digitization by the ADC. The digitized samples are stored in an internal First In First Out (FIFO) buffer and transferred to an external microcontroller via a digital interface such as Serial Peripheral Interface (SPI) for further processing [15]. This architecture enables compact integration while maintaining low power consumption and flexible configuration through programmable radar parameters.

The BGT60TR13C operates using the FMCW radar principle. During each chirp, the transmitted carrier frequency increases linearly with time according to the modulation bandwidth B and chirp duration T_{chirp}, producing a frequency slope $S = B/T_{\text{chirp}}$. After propagation and reflection from a target at distance d, the received signal experiences a round-trip delay $\tau_r = 2d/c$, where c denotes the speed of light. Mixing the received signal with the transmitted reference signal produces a beat frequency f_r proportional to the target range d,

$$f_r = S\tau_r = S\frac{2d}{c}, \tag{2.1}$$

where τ_r is the round-trip delay, and c is the speed of light. Hence, the range estimate is

$$d = \frac{cf_r}{2S}. \tag{2.2}$$

Let $x_m[n, q]$ denote the complex IF samples for receiver m, fast-time sample index $n = 0, \ldots, N_{\text{sample}} - 1$ and chirp index $q = 0, \ldots, N_{\text{chirp}} - 1$. As indicated by the Direct Current (DC) Bias Removal block in Figure 2.3, we first subtract the mean across fast-time samples in each chirp to remove static offsets. We then apply two separate one-dimensional Blackman–Harris windows along fast time and slow time: $w_r[n]$ and $w_d[q]$, respectively. In the implementation, these two windows are applied before the corresponding Fast Fourier Transform (FFT), but for compactness, Figure 2.3 shows them as a single Blackman–Harris window block. The range FFT is then computed along the fast-time axis, yielding the Range-FFT $X_m[r, q]$ for receiver m, range bin r and chirp index q. For each fixed range bin r, we next apply the slow-time window $w_d[q]$ and perform a Doppler FFT across chirps. This produces the complex Range–Doppler map $X_m[r, d]$, where d is the Doppler bin index. These per-antenna Range–Doppler cubes form the input to the subsequent MTI, Capon beamforming (Section 2.2.2) and RASSO (Section 3.1).

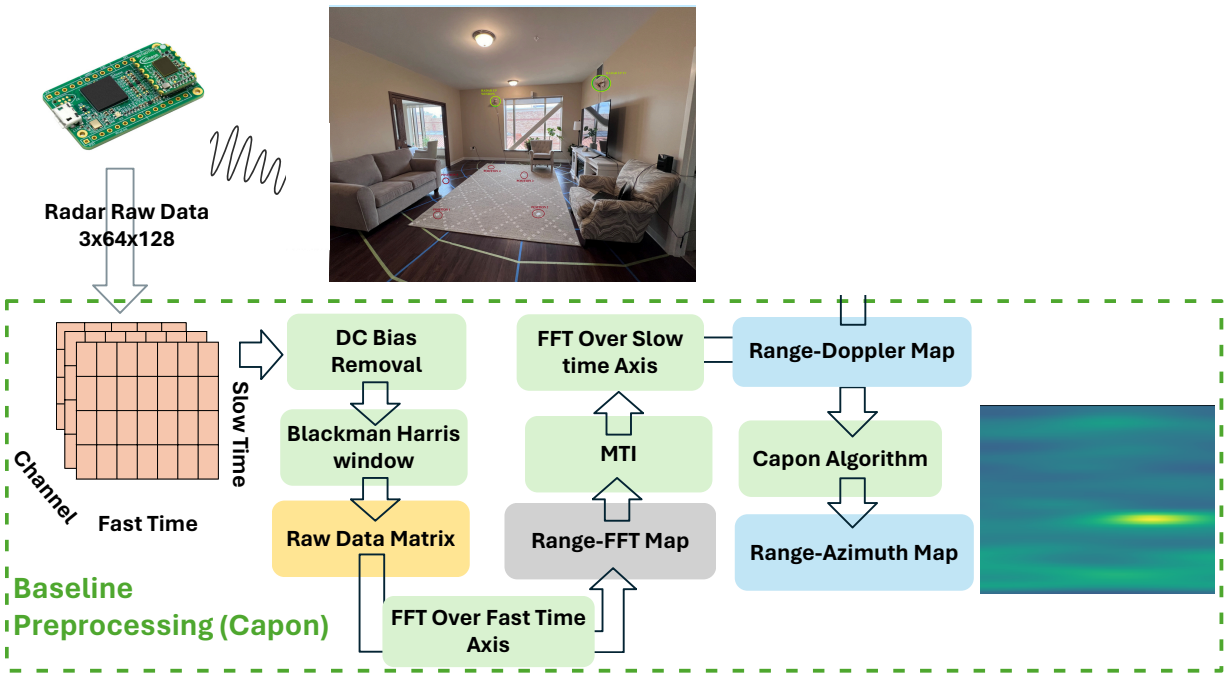

Figure 2.3: Our baseline signal processing pipeline (green-dash) [18], [19], [20].

Range-Doppler Map Generation

Applying the range FFT and Doppler FFT as described above to the ADC data cube $x_m[n, q]$ produces, for each receiver m, a complex Range–Doppler map $X_m[r, d]$ whose elements represent the reflected signal strength at range bin r and Doppler bin d. Here, N_r and N_d denote the total numbers of range and Doppler bins after the FFTs, respectively, and N_{RX} denotes the number of receive antennas (for our BGT60TR13C board $N_{\text{RX}} = M = 3$). Each element $X_m[r, d]$ in the Range-Doppler Map (RDM) represents the reflected signal strength (amplitude and phase) from a specific range bin r and a specific Doppler (velocity) bin d. In our Single-Input and Multiple-Output (SIMO) radar system with $M = 3$ receivers, this process yields M separate RDMs, $X_m[r, d]$, one for each receive antenna m:

$$X_m[r, d] \in \mathbb{C}, \quad r \in [0, N_r - 1],\ d \in [0, N_d - 1],\ m \in [0, N_{\text{RX}} - 1].$$

Stacking these across receivers produces the **Range–Doppler Cube (RDC)**,

$$\mathbf{Y}[r, d, m] \ = \ X_m[r, d], \qquad m = 0, \ldots, N_{\text{RX}} - 1,$$

which supplies both inter–element phase (for digital beamforming and Capon/Minimum Variance Distortionless Response (MVDR) in Sections 2.2.1–2.2.2) and Doppler–axis samples (for our RASSO , Doppler-axis warping and resampling in Section 3.1).

Clutter suppression A critical challenge for indoor sensing is static clutter (walls, furniture, etc), which creates a massive signal at zero Doppler velocity (the DC component) in the RDM. We therefore apply a first–order exponential MTI [21] filter independently on each receive channel m. Let $\mathrm{RDM}_{k,m}(r,d)$ denote the Range–Doppler map at frame k for receiver m. For every bin (r,d) we update the clutter estimate $C_{k,m}(r,d)$ and clutter–suppressed output $Y_{k,m}(r,d)$ as

$$C_{k,m}(r,d) = \alpha\, C_{k-1,m}(r,d) + (1-\alpha)\,\mathrm{RDM}_{k,m}(r,d), \tag{2.3}$$

$$Y_{k,m}(r,d) = \mathrm{RDM}_{k,m}(r,d) - C_{k,m}(r,d), \tag{2.4}$$

where $\alpha \in (0,1)$ is the forgetting factor. For notational brevity, subsequent equations use $Y_k(r,d)$ and $C_k(r,d)$, but the MTI in (2.3)–(2.4) is applied identically to each receive channel m. We refer to this MTI-based removal as **static clutter suppression** or **clutter suppression** throughout the paper.

2.2 Angle-of-Arrival Estimation

2.2.1 Digital Beamforming (DBF) from Infineon

Following the Range-Doppler preprocessing and MTI described in Section 2.1.1, we estimate target bearing by *standard* (delay-and-sum) Digital Beamforming (DBF) using the vendor's baseline implementation for the BGT60TR13C device. Particularly for our specific quasi-static application, we adapt the Infineon Software Development Kit (SDK) code sample `range-angle-map.py` to our sensor configuration specified in Table 3.1, modify MTI/static-suppression settings to be compatible with our processors for a fair comparison.[1]

[1] See Infineon's DBF application notes for BGT60TR13C, we follow their geometry and weight definition. [17, 22]

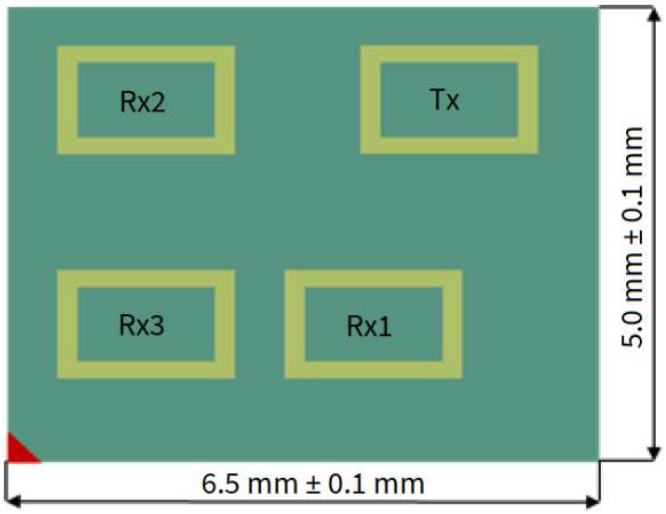

Figure 2.4: BGT60TR13C radar package outline [22].

As laid out in the Fig. 2.4, three receivers (Rx1, Rx2, Rx3) are arranged in an L-shape to enable 2-D AOA sensing. We adopt Rx3 as the reference element and origin. With inter-element spacings $d_x = d_y = \lambda/2$, the RX element position vectors follow the Infineon application note [[22], Eq. (2)]:

$$\overrightarrow{\mathrm{Rx}_1} = \left[\tfrac{\lambda}{2},\, 0,\, 0\right]^\top = [d_x, 0, 0]^\top, \tag{2.5}$$

$$\overrightarrow{\mathrm{Rx}_2} = \left[0,\, \tfrac{\lambda}{2},\, 0\right]^\top = [0, d_y, 0]^\top, \tag{2.6}$$

$$\overrightarrow{\mathrm{Rx}_3} = [0,\, 0,\, 0]^\top. \tag{2.7}$$

Cartesian coordinates (x, y) of a point at polar range r and angles (θ, ϕ) follow ([22] Eq. (3)):

$$x = r\cos\theta\cos\phi, \qquad y = r\sin\phi. \tag{2.8}$$

Let $\vec{k} = \frac{2\pi}{\lambda}\,[\cos\theta\cos\phi,\ \sin\phi]^\top$ denote the wave-number vector along the array's azimuth (x) and elevation (y) axes. Following [[22] Eqs. (4)–(5)], the per-direction complex weight (steering) vector is defined as:

$$\mathbf{w}(\theta,\phi) = \left[e^{j\frac{2\pi}{\lambda}\,d_x\cos\theta\cos\phi},\ e^{j\frac{2\pi}{\lambda}\,d_y\sin\phi},\ 1\right]. \tag{2.9}$$

Here $\mathbf{w}(\theta,\phi) = [w_1(\theta,\phi), w_2(\theta,\phi), w_3(\theta,\phi)]^\top$, and $w_m(\theta,\phi)$ denotes the m-th complex weight applied to receiver m.

Delay–and–sum beamformer and range–angle map Let

$$\mathbf{z}(r,k) = [z_1(r,k), z_2(r,k), z_3(r,k)]^\top \tag{2.10}$$

collect the MTI-filtered Range–Doppler samples for the three receive channels at range bin r in frame k. With the steering vector $\mathbf{w}(\theta,\phi)$ defined in Eq. (2.9), the delay–and–sum DBF output is

$$y_{\mathrm{DBF}}(r,\theta,\phi;k) = \mathbf{w}^H(\theta,\phi)\,\mathbf{z}(r,k) = \sum_{m=1}^{N_{\mathrm{RX}}} w_m(\theta,\phi)\, z_m(r,k), \tag{2.11}$$

and the corresponding beam power is

$$P_{\mathrm{DBF}}(r,\theta,\phi;k) = \left|y_{\mathrm{DBF}}(r,\theta,\phi;k)\right|^2. \tag{2.12}$$

Following Infineon's SDK, the 2D-FFT DBF yields a 4-D spectrum $P_{\mathrm{DBF}}(r,\theta,\phi;k)$ (range r, azimuth θ, elevation ϕ, frame k). For visualization and downstream Cell Averaging Constant False Alarm Rate (CA-CFAR), we obtain the RA map by non-coherent integration across elevation

$$\mathrm{RA}_{\mathrm{DBF}}(r,\theta;k) = \sum_{\phi\in\Phi} P_{\mathrm{DBF}}(r,\theta,\phi;k), \tag{2.13}$$

which matches Infineon's application note and the reference implementation (*range-angle-map.py*). The Infineon's application note [17] remarks that more sophisticated methods, for example, Capon/MVDR, can sharpen the mainlobe at the cost of higher computation and additional assumptions. In this study, we use the Infineon standard DBF as a standard-provided baseline and compare it to our baseline RA processor, which adopts Capon/MVDR (described in Section 2.2.2) that narrows the lobe and the proposed RASSO (described in Section 3.1), which sharpens weak micro-Doppler responses before beamforming. All pipelines share identical MTI and static clutter suppression, and the same downstream CA-CFAR configuration (Section 2.3.1).

2.2.2 Capon/MVDR Beamforming

After Range–Doppler pre–processing and MTI (Section 2.1.1), each frame provides, for every range bin r, a complex RD slice per receive channel, denoted $Y_m(r,d) \in \mathbb{C}$, where $m \in \{1,\ldots,N_{Rx}\}$ indexes Rx channels and d indexes the MTI–filtered Doppler bins. Following the Infineon geometry used in our DBF baseline, we employ the L–shaped three–Rx layout and form a two–element azimuth subarray from the Rx1 and Rx3 arms to estimate azimuth

only to be consistent with our SDK workflow [17,22]. For a fixed range bin r, we collect N_D neighbouring Doppler indices around the MTI passband into a slow-time *snapshot matrix* (one column per Doppler cell, one row per Rx channel):

$$X_r \;=\; \begin{bmatrix} Y_1(r, D_1) & \cdots & Y_1(r, D_{N_D}) \\ \vdots & \ddots & \vdots \\ Y_{N_{Rx}}(r, D_1) & \cdots & Y_{N_{Rx}}(r, D_{N_D}) \end{bmatrix} \;\in\; \mathbb{C}^{N_{Rx} \times N_D}. \tag{2.14}$$

The Capon algorithm or MVDR beamformer is applied to the Range-Doppler data, enabling the generation of precise RA and Range-Elevation (RE) feature maps [23]. The algorithm estimates the AOA by solving an optimization problem that minimizes the output power P_{out} while ensuring a distortionless response in the desired target direction:

$$\min_{\mathbf{w}} \left(P_{\text{out}} = \frac{1}{2} \mathbf{w}^{\mathrm{H}} \mathbf{R}_r \, \mathbf{w} \right) \quad \text{s.t.} \quad \mathbf{w}^{\mathrm{H}} \mathbf{a}(\theta) = 1. \tag{2.15}$$

where R_r is the sample spatial covariance at range r that can estimated as following (2.14):

$$R_r \;=\; \frac{1}{N_D} X_r X_r^H \;\in\; \mathbb{C}^{N_{Rx} \times N_{Rx}}. \tag{2.16}$$

and form its Moore–Penrose pseudoinverse R_r^{-1} (no diagonal loading, $\delta = 0$). With azimuth steering vector:

$$a(\theta) \;=\; \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta} \end{bmatrix}, \tag{2.17}$$

the Capon power spectrum at range r and azimuth θ is computed as

$$P_{\text{Cap}}(r, \theta) \;=\; \frac{1}{a(\theta)^H \, \widehat{R}_r^{-1} \, a(\theta)}. \tag{2.18}$$

which are similar to our previous work on in-vehicle monitoring [20] and fall detection [24]. The only difference in (2.18)'s notation is that we aggregate over all Doppler bins at that range to estimate R_r. We define the Range-Azimuth feature map as $\text{RA}_{\text{Capon}}(r, \theta) = |P_{\text{Capon}}(r, \theta)|$, optionally smoothed over 5 frames and normalized to $[0, 1]$ for visualization and CFAR. In the remainder of this thesis, we refer to this Capon-based RA estimation (2.15)–(2.18) as our **Baseline method** or **Baseline RA** (**Conv RA**). When we include this method as the full signal-processing chain used for baseline results and ablations, we call the end-to-end chain the **Baseline pipeline** as shown in the Figure 2.3.

2.3 Constant False Alarm Rate

2.3.1 Cell-Averaging Constant False Alarm Rate (CA-CFAR)

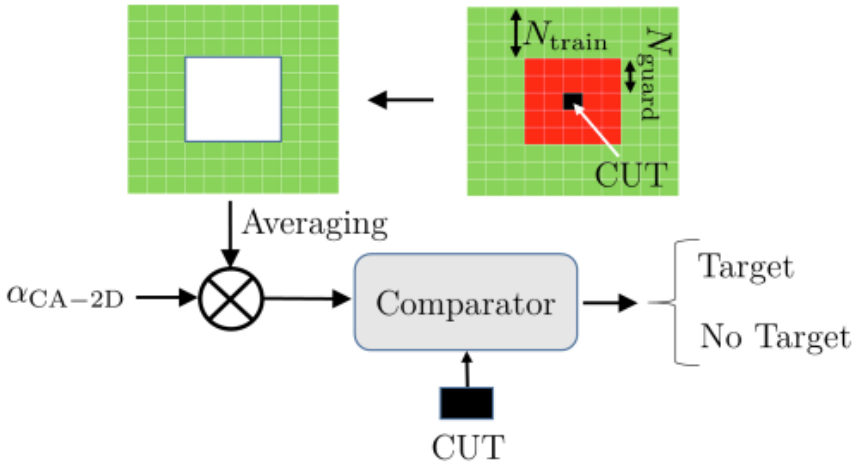

Figure 2.5: 2D CA-CFAR [25].

Let $Y(i,j)$ denote a generic 2-D radar feature map - either a Range-Doppler ($Y(r, f_D)$) or Range-Azimuth ($Y(r,\theta)$) case, we perform detection on these image-like feature maps using the *CA-CFAR* as described in [26, section 7.2-7.3]. For each Cell–Under–Test (CUT), we place a square window of side $2H+1$ cells centred at CUT. A *guard band* of G cells per side protects the CUT and its intermediate neighbours from leaking target energy into the noise estimate. The *training ring* therefore had width T cells per side, so the training region is the outer ring of the $(2H+1)\times(2H+1)$ window after removing the $(2G+1)\times(2G+1)$ guard as visualized in Fig 2.5. Let $\mathcal{T}(r,c)$ be the set of training cells. The sample average estimates the local mean interference power

$$\widehat{\mu}(r,c) = \frac{1}{N_{\text{train}}} \sum_{(i,j)\in\mathcal{T}(r,c)} Y(i,j) \tag{2.19}$$

$$N_{\text{train}} = (2H{+}1)^2 - (2G{+}1)^2 \tag{2.20}$$

A detection is declared at (r,c) when

$$Y(r,c) \;>\; \Gamma(r,c), \qquad \Gamma(r,c) \triangleq k_{\text{CFAR}}\,\widehat{\mu}(r,c), \tag{2.21}$$

where $k_{\mathrm{CFAR}} > 0$ is a sensitivity factor (or α in Richards [26, Eq.(7.11)] that controls the effective false-alarm rate. In preliminary sweeps experiments, we varied $G \in \{2, 4, 6\}$ and $T \in \{6, 10, 14\}$, and found performances stable once $T \geq 10$. Thus, unless specified in below in Section 3.1.4– 3.1.7, we vary k_{CFAR} and use G=4, T=10, which yields H=14 and N_{train}=760 training cells per CUT.

2.3.2 Order-Statistic Constant False Alarm Rate (OS-CFAR)

Unlike CA-CFAR, which estimates the local interference level by averaging all reference cells, Order-Statistic Constant False Alarm Rate (OS-CFAR) replaces the mean with a ranked sample from the reference window. This modification is particularly useful when the reference ring is contaminated by a small number of large returns, such as clutter edges, sidelobe leakage, or neighbouring targets. In such cases, the arithmetic mean can be biased upward, causing missed detections, or can fluctuate strongly from one location to the next. OS-CFAR [27] addresses this by using an order statistic that is less sensitive to outliers. Let the reference-cell powers in the training window around a given CUT be denoted by

$$\{p_1, p_2, ..., p_n\} \tag{2.22}$$

where N is the number of reference cells after removing the guard region. Sort these values in ascending order [21],

$$p_{(1)} \leq p_{(2)} \leq ... \leq p_{(N)} \tag{2.23}$$

Instead of estimating the background by the sample mean, OS-CFAR uses one ranked element $p_{(k)}$ as the local noise-plus-clutter estimate, where k is a design parameter. The detection rule can then be written as

$$I_{\mathrm{det}}(\mathrm{CUT}) = \begin{cases} 1, & |X(\mathrm{CUT})|^2 > \mu p_{(k)}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.24}$$

where μ is a scale factor chosen to obtain the desired false-alarm behaviour.

2.4 Deep Learning

Deep learning is a class of machine-learning methods based on multi-layer neural networks that learn hierarchical representations directly from data. Unlike traditional approaches

that rely on manually engineered features, deep-learning models jointly learn both the representation and the decision function through optimization over a large set of parameters [28,29]. This capability has made deep learning effective for image analysis [30], speech processing [31], sequence modelling [32,33], and many sensing applications [34,35] in which the raw input contains a complex nonlinear structure. A neural network consists of interconnected computational units, or neurons, arranged in layers. Each layer transforms its input into a new representation, and deeper layers progressively capture higher-level abstractions. During training, the network parameters are adjusted by minimizing a loss function over a set of labelled examples, which can be achieved through gradient-based optimization and backpropagation [36]. In this section, the main deep-learning components used later in the thesis are reviewed in increasing order of complexity. We begin with the perceptron as the fundamental building block of neural networks, then introduce activation functions and feed-forward training. These concepts provide the foundation for convolutional neural networks, which are suitable for image-like inputs, and recurrent sequence models such as Recurrent Neural Network (RNN), LSTMs, and bidirectional LSTMs, which are used to model temporal dependencies.

2.4.1 Perceptron

A perceptron is the fundamental single-neuron building block of a neural network. As illustrated in Figure 2.6, a perceptron takes an input vector $\mathbf{x} \in \mathbb{R}^d$, computes a weighted sum of its components, adds a bias term, and then applies a nonlinear activation function. Mathematically, this operation is written as

$$z = \mathbf{w}^\top \mathbf{x} + b, \tag{2.25}$$

followed by

$$y = f(z), \tag{2.26}$$

where $\mathbf{w}$ is the weight vector, b is the bias term, and $f(\cdot)$ is the activation function. The linear part $\mathbf{w}^\top \mathbf{x} + b$ determines how strongly each input contributes to the neuron response, while the activation function introduces nonlinearity into the model.

The perceptron provides the basic computational template for deeper neural architectures. By combining many such neurons into layers, the network can represent increasingly complex nonlinear mappings between the input and the output. Since the perceptron output depends critically on the choice of activation function, it is natural to next discuss the most common activations used in neural networks.

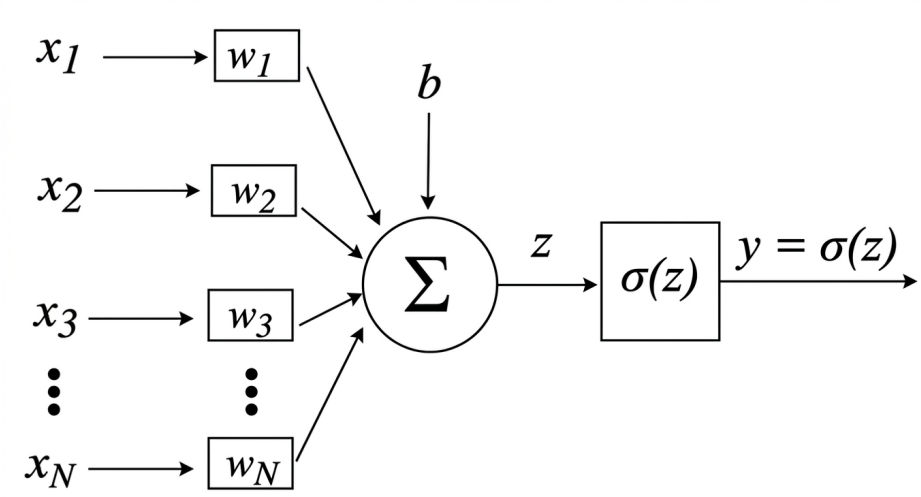

Figure 2.6: Illustration of a perceptron. The inputs $x_1, \ldots, x_N$ are weighted by $w_1, \ldots, w_N$, summed with bias b, and passed through an activation function $\sigma(\cdot)$ to produce the neuron output.

2.4.2 Activation Functions

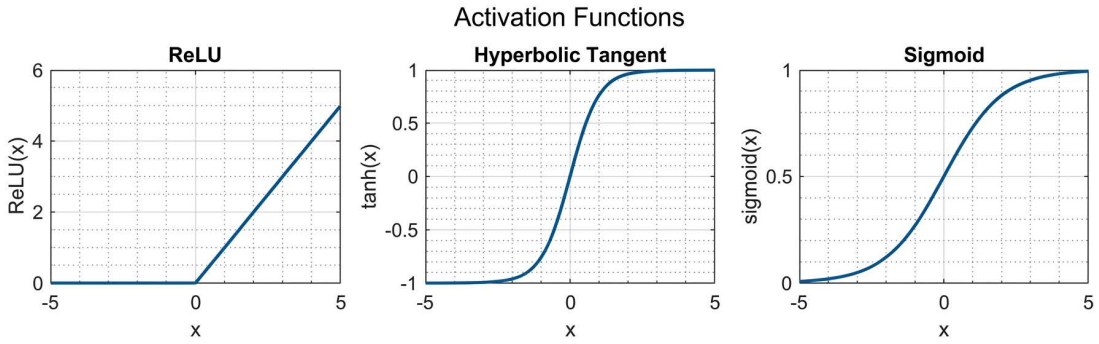

Figure 2.7: Common activation functions used in neural networks, including ReLU, hyperbolic tangent, and sigmoid [37].

Activation functions introduce nonlinearity into neural networks. Nonlinear activations, therefore, allow neural networks to learn more complex decision boundaries and hierarchical feature representations. Figure 2.7 shows three common activation functions used in practice: the sigmoid function, the hyperbolic tangent function, and the Rectified Linear Unit (ReLU). The sigmoid function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2.27}$$

which maps the input to the interval $(0, 1)$. Because of this property, sigmoid is often used when an output needs to be interpreted as a probability. However, sigmoid saturates for large positive or negative inputs, which can slow training due to small gradients.

The hyperbolic tangent function is defined as

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{2.28}$$

which maps the input to $(-1, 1)$. Compared with sigmoid, $\tanh(\cdot)$ is zero-centred, which can improve optimization in some settings, although it still suffers from saturation in the tails.

The ReLU is defined as

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.28}$$

ReLU is widely used in modern deep networks due to its simplicity and optimization behaviour. It preserves positive responses while suppressing negative ones, thereby encouraging sparse intermediate representations. For classification problems, the final layer often uses the softmax function

$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \tag{2.29}$$

which converts a vector of class scores into normalized class probabilities. Thus, the perceptron of Figure 2.6 becomes a useful nonlinear computational unit only after choosing an activation function such as those shown in Figure 2.7. Once such neurons are arranged into layers, they form the basis of feed-forward neural networks.

2.4.3 Feed-Forward Neural Networks and Training

By arranging many perceptrons into layers, a neural network can model increasingly complex nonlinear mappings between input and output. A feed-forward neural network passes

information from the input layer through one or more hidden layers to the output layer without feedback connections. If the network contains L layers, the output of layer ℓ can be written as

$$\mathbf{h}^{(\ell)} = f^{(\ell)}\left(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \tag{2.30}$$

where $\mathbf{h}^{(0)} = \mathbf{x}$ is the input, $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the trainable weights and biases, and $f^{(\ell)}(\cdot)$ is the activation function at layer ℓ. Figure 2.8 illustrates a typical feed-forward

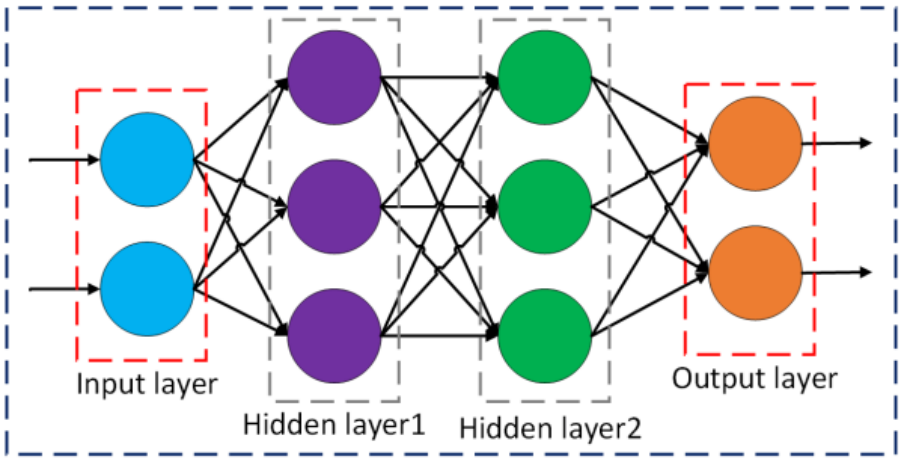

Figure 2.8: Example of a feed-forward neural network. Information flows from the input layer through hidden layers to the output layer without recurrence [38].

network in which information propagates from the input layer through hidden layers to the output layer. During training, the network parameters are adjusted so that the model output matches the target labels as closely as possible over the training set. This is usually achieved by minimizing a loss function using gradient descent or one of its variants. In each training iteration, the input is propagated forward through the network to compute a prediction, the loss is evaluated, and gradients of the loss with respect to the parameters are computed by backpropagation. The parameters are then updated in the direction that reduces the loss. For classification tasks with C classes, a common choice is the cross-entropy loss

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{c=1}^{C} y_c \log \hat{y}_c, \tag{2.31}$$

where y_c is the target indicator and $\hat{y}_c$ is the predicted class probability. If θ denotes all trainable parameters of the network, a basic gradient-descent update takes the form

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}, \tag{2.32}$$

where $\eta > 0$ is the learning rate. Feed-forward networks provide the conceptual basis for more structured architectures. In particular, when the input has spatial organization, such as an image or a two-dimensional radar map, convolutional neural networks are usually more effective than fully connected layers alone.

2.4.4 Convolutional Neural Network (CNN)

A convolutional neural network is a feed-forward neural network designed for grid-structured inputs such as images, spectrograms, and two-dimensional feature maps. CNNs differ from fully connected networks in that they exploit local spatial structure through convolution, parameter sharing, and hierarchical feature extraction. This reduces the number of trainable parameters while preserving important local patterns.

Convolutional Layers

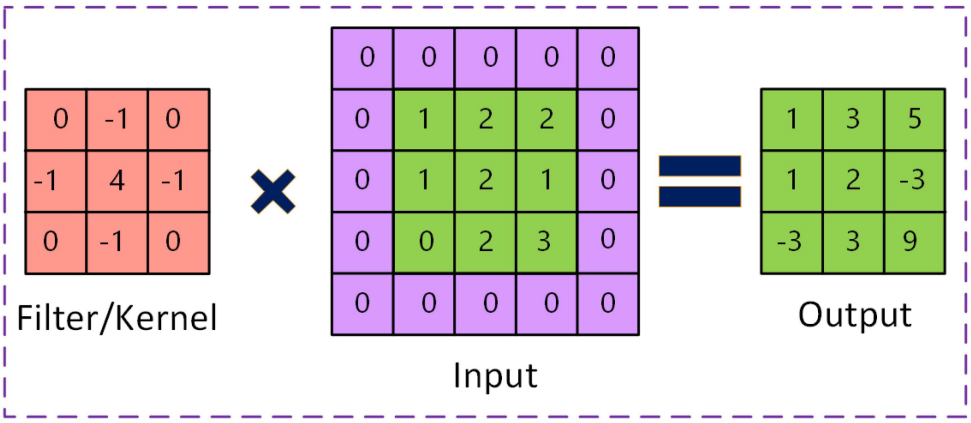

Figure 2.9: Example of the convolution operation in a CNN. A local kernel is applied across the input feature map to generate an output activation map [38].

The convolutional layer is the fundamental building block of a CNN. It consists of a set of learnable kernels, also called filters, that are convolved with the input feature map. Figure 2.9 visualizes this operation. For a two-dimensional input X and kernel K, the output feature map Y at location (i, j) may be written as

$$Y(i,j) = \sum_m \sum_n K(m,n)\, X(i+m, j+n) + b, \tag{2.33}$$

where b is a bias term. Several kernels are learned simultaneously, producing multiple output channels. Each filter responds strongly to certain local structures in the input. In early layers, these responses may correspond to low-level patterns such as edges, corners, or simple blobs, whereas in deeper layers they become more abstract and task-specific. Because the same kernel is reused across the full input, CNNs implement parameter sharing, which greatly reduces model size compared with dense fully connected architectures.

Pooling Layers

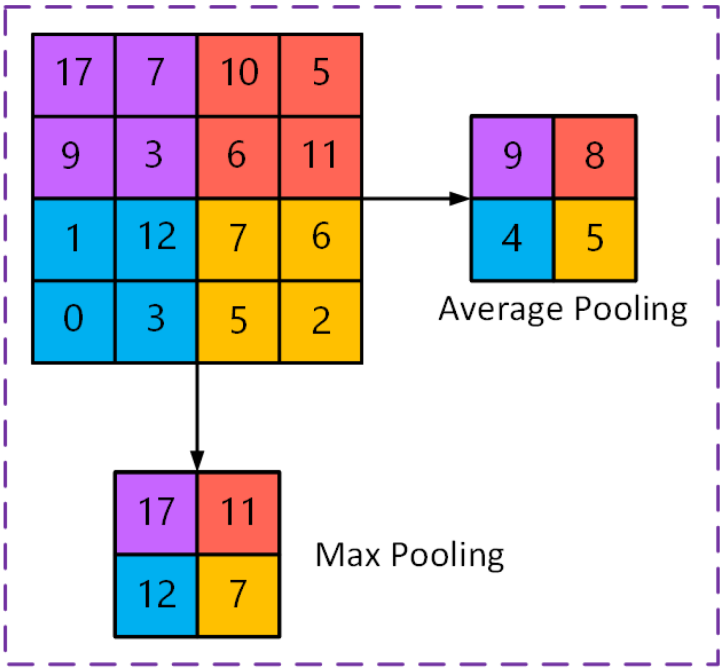

Figure 2.10: Illustration of average pooling and max pooling [38].

Pooling layers are used to reduce the spatial dimensions of intermediate feature maps while retaining the most relevant information. This lowers computational cost, improves robustness to small translations, and can reduce overfitting. Figure 2.10 illustrates two common forms: average pooling and max pooling. For a local pooling window containing values $\{y_1, \ldots, y_N\}$, average pooling computes

$$P_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} y_i, \tag{2.34}$$

while max pooling computes

$$P_{\max} = \max_i y_i. \tag{2.35}$$

Average pooling summarizes the average activation in a region, whereas max pooling preserves only the strongest response. The choice between them depends on the desired invariance properties and the task. In many vision-based settings, max pooling is preferred because it preserves the most salient local activation, while average or adaptive average pooling can be useful when a compact global summary is needed before classification.

Batch Normalization and Dropout

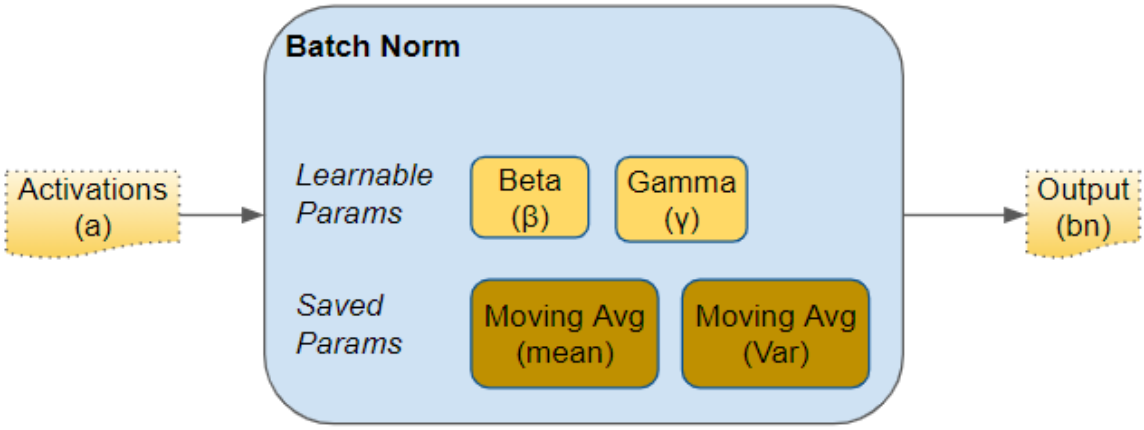

Figure 2.11: Conceptual diagram of batch normalization. Intermediate activations are normalized using the mini-batch mean and variance, then rescaled and shifted by learnable parameters to stabilize network training [39].

Batch normalization is introduced to stabilize training by normalizing intermediate activations within a mini-batch. Figure 2.11 shows the basic idea. Given an activation a, the batch mean μ_B and variance σ_B^2 are first computed, and the activation is normalized as

$$\hat{a} = \frac{a - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{2.36}$$

where ϵ is a small constant for numerical stability. The normalized activation is then rescaled and shifted using learnable parameters γ and β:

$$y = \gamma \hat{a} + \beta. \tag{2.36}$$

Batch normalization often improves convergence, stabilizes optimization, and permits the use of larger learning rates.

Dropout is a regularization technique that randomly suppresses a fraction of activations during training. In its standard form, a binary mask is sampled independently for each activation from a Bernoulli distribution. Let h_i denote a hidden activation and let

$$m_i \sim \text{Bernoulli}(1-p),$$

where p is the dropout probability. In the inverted-dropout form used in common deep-learning libraries, the retained activations are rescaled during training:

$$\tilde{h}_i = \frac{m_i}{1-p}\, h_i. \tag{2.37}$$

This preserves the expected activation magnitude so that, during evaluation, dropout can be removed and the layer acts as the identity mapping. By preventing the network from depending too strongly on any one pathway, dropout reduces overfitting and improves generalization. The standard CNN pipeline, therefore, consists of convolution, nonlinear activation, optional batch normalization, optional pooling, and finally one or more fully connected layers for classification.

2.4.5 Recurrent Neural Networks (RNN)

While CNNs are effective for spatial feature extraction, many learning problems involve sequential data in which temporal order is important. Recurrent neural networks are designed for such problems by maintaining a hidden state that evolves over time and carries information from previous inputs. Figure 2.12 illustrates the basic recurrent structure. Given an input sequence $\{x_t\}$, a standard RNN updates its hidden state as

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{2.38}$$

where x_t is the input at time step t, h_t is the hidden state, W_{xh} and W_{hh} are the input-to-hidden and hidden-to-hidden weight matrices, b_h is a bias vector, and $\phi(\cdot)$ is a nonlinear activation function. The output may then be written as

$$y_t = \psi(W_{hy}h_t + b_y), \tag{2.39}$$

where W_{hy} maps the hidden state to the output and $\psi(\cdot)$ is the output activation. The defining feature of an RNN is recurrence: the hidden state at time t depends on both

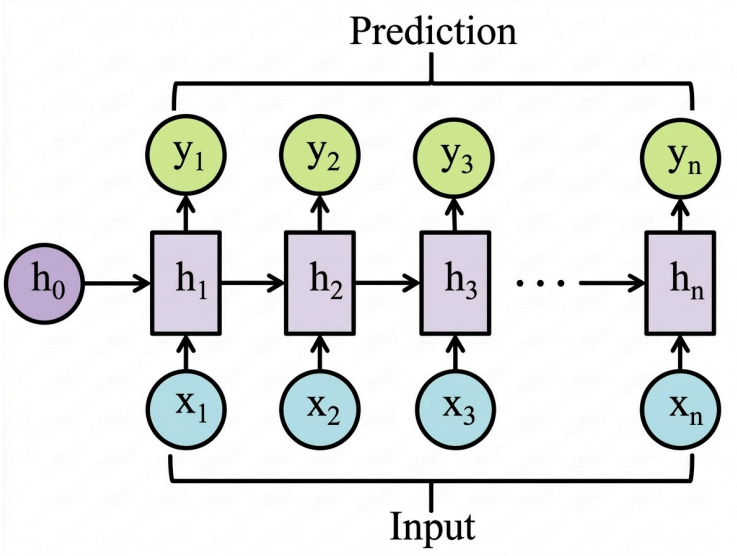

Figure 2.12: Basic RNN architecture [40].

the current input and the previous hidden state. This allows the network to model temporal dependencies and context. However, standard RNNs often suffer from vanishing or exploding gradients when trained over long sequences, which limits their ability to learn long-range dependencies reliably. This motivates the use of gated recurrent architectures such as LSTM.

2.4.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory networks are a gated variant of recurrent neural networks designed to address the vanishing-gradient problem and enable learning over longer temporal contexts. An LSTM maintains both a hidden state h_t and a cell state c_t, where the cell state acts as a memory path that can preserve information across many time steps. An LSTM cell contains three gates: the input gate i_t, forget gate f_t, and output gate o_t. Given an input x_t and previous hidden state h_{t-1}, these gates are computed as

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \tag{2.40}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \tag{2.41}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \tag{2.42}$$

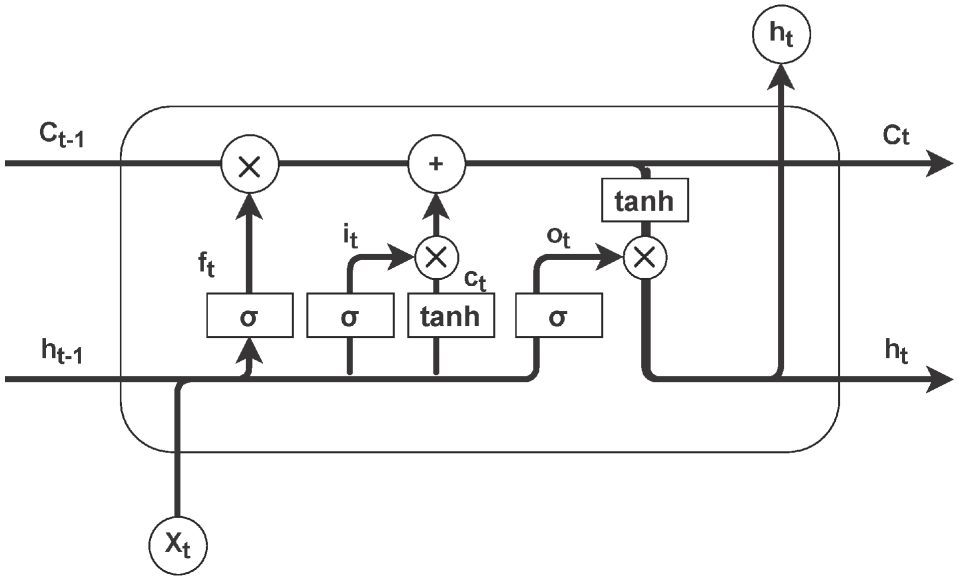

Figure 2.13: Architecture of the LSTM network [40].

together with the candidate cell update

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c). \tag{2.43}$$

The cell state and hidden state are then updated according to

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{2.44}$$

$$h_t = o_t \odot \tanh(c_t), \tag{2.45}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\odot$ denotes element-wise multiplication. The forget gate f_t controls how much previous memory c_{t-1} is retained, the input gate i_t controls how much new information is written into the memory cell, and the output gate o_t controls how much of the updated memory contributes to the hidden state. Through these mechanisms, the LSTM can preserve useful temporal information over longer horizons than a standard RNN.

2.4.7 Bidirectional LSTM

Bidirectional LSTM (BiLSTM) extends the LSTM by processing the sequence in both forward and backward directions. Figure 2.14 illustrates this idea. Let $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote

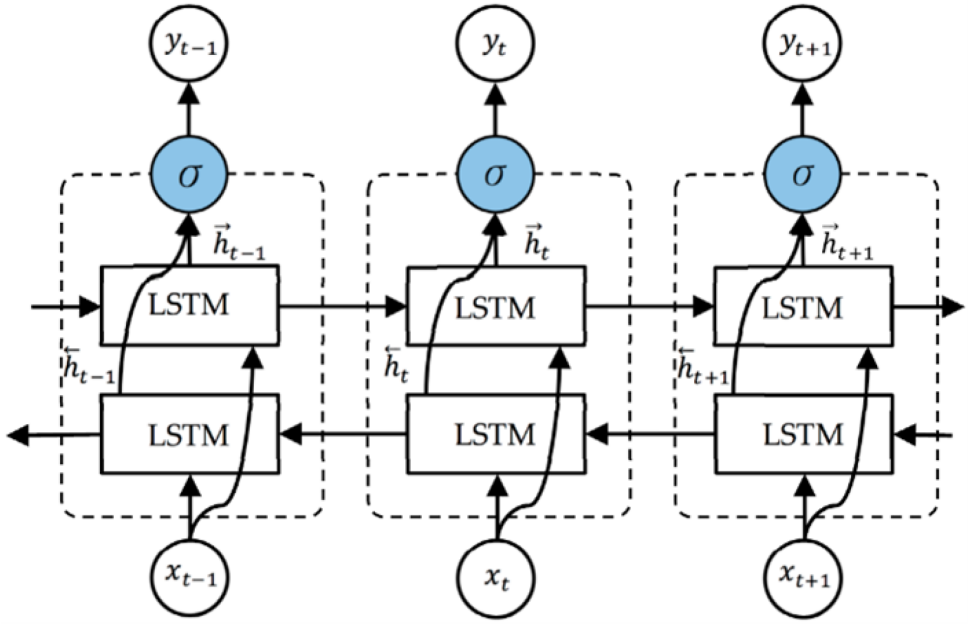

Figure 2.14: Architecture of the biLSTM network [40].

the forward and backward hidden states, respectively. The output may be formed by concatenation as

$$y_t = \psi\left(W_y[\overrightarrow{h_t}; \overleftarrow{h_t}] + b_y\right). \tag{2.46}$$

By using context from both past and future positions in the sequence, a BiLSTM can capture richer temporal structure than a unidirectional recurrent model. As shown in Figure 2.14, the sequence is processed by two parallel LSTM streams: one from left to right and the other from right to left. Their hidden representations are then combined before the final output layer. This architecture is especially useful when the interpretation of one time step benefits from information on both sides of that time step. In practice, LSTM and BiLSTM layers are often combined with CNN feature extractors, where the CNN extracts spatial features from each frame and the recurrent module models how those features evolve over time.

Chapter 3

Quasi-static Activities, Occupancy And Post-fall Detections

3.1 Doppler-Domain Warping and Resampling for Semi-Static Human Occupancy Detection

3.1.1 Related Work

Radar provides a privacy-preserving alternative by measuring motion and reflectivity without capturing identifiable imagery and has been widely studied for indoor localization [41, 42]. A key difficulty, however, is quasi-static occupancy [43]: senior residents may spend long periods lying in bed or sitting quietly, producing only weak micro-motions from postural sway and breathing. These motions concentrate energy near zero Doppler, where severe static clutter from walls, furniture, and the bed itself also appears. Reliable indoor quasi-static occupant detection has long been known to be challenging due to its very low Doppler shift; even recent FMCW systems acknowledge that detecting static people is more difficult than detecting moving targets [44], [45]. Early efforts to detect stationary occupants focused on resolving respiratory motion using Ultra-Wideband (UWB) or continuous wave (CW) radars. For example, Song et al. [46] developed an algorithm to detect a stationary person based on the spectrum of a demodulated respiration signal, specifically targeted breathing frequencies. This approach requires the subject's respiration to be sensitive to distance-null points. Additionally, FMCW radars provide range and angle information, enabling indoor localization of people. A study by Luo et al. [47] proposed a

standard-deviation weighting method to locate a person by emphasizing range bins with high amplitude variance across chirps, but their system still relies on heuristic weighting. Later, Kaiser et al. [48] improved semi-static human localization by combining Range-Angle estimates with clutter suppression and thresholding. In parallel, Li et al. [44] employed gls music processing with Multiple-Input Multiple-Output (MIMO) arrays to localize standing subjects, but their approach requires multiple antennas and is computationally intensive. These methods generally treat stationary detection as a by-product of localization and do not explicitly address the faint micro-Doppler components of semi-static subjects.

Sabri et al. [49] introduced the MCROOD framework, which operates on Range-Doppler (RD) images to detect reconstruction-based out-of-distribution (OOD). In other words, it simultaneously predicts whether a human is present and flags unfamiliar disturbances. Their method relies on E-RESPD, which temporally accumulated a heatmap produced by sliding-window summation/averaging over magnitude RDIs. While promising, E-RESPD's consecutive-frame averaging methods eliminate per-antenna phase, which cannot be used directly for AOA beamforming. In addition, like many previous works, their work mainly reports results from controlled laboratory environments and has not yet been shown to deploy in dynamic scenarios. To ensure compliance, we also strive to reproduce their work and compare it fairly with our methods, as shown later in Section 3.1.6. To resolve these issues and overcome these disadvantages, we introduce RASSO, an invertible Doppler-domain non-linear warping and resampling of the Doppler axis that densifies the sampling grid around near-zero micro-Doppler before spatial processing, thereby preserving per-antenna phase while making faint μ-Doppler signatures more prominent in the subsequent RA maps. Integrated with MVDR/Capon beamforming and 2-D CFAR, our signal-processing pipeline boosts the SNR of quasi-static targets and enables compact CNN/LSTM classifiers to reach $\approx$ 92-99% accuracy on both held-out and generalization tests in realistic living-room scenarios. Unlike previous work, our results emphasize robustness in real-world settings with clutter and semi-static posture changes, supporting dependable deployment beyond the lab.

3.1.2 Dataset Collection

Table 3.1: Radar hardware and waveform configuration (BGT60TR13C).

Parameter	**Value**
Band / Center frequency	58–63.5 GHz (center $\approx$ 60 GHz)
Sweep bandwidth B	$\approx$ 500 MHz
Frame rate	10 Hz
Chirps per frame N_{chirp}	128
Samples per chirp N_{sample}	64
Chirp Duration	64 μs
Range resolution Δr	0.30 m
Max range	9.6 m
Max unambiguous speed	3 m/s

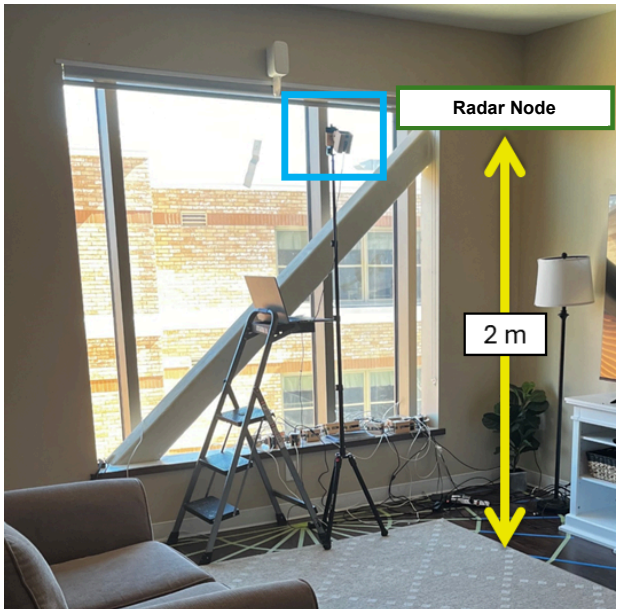

Figure 3.1: Testing setups for data collection in the MIRADA apartment at the Schlegel–UW Research Institute for Aging, operated by the Wireless Sensors and Devices Laboratory. The yellow 2 m arrow refers only to the approximate sensor-to-monitored-floor separation visible in this illustrative figure. The actual mounting height used in all recordings is $\approx$ 2.5 m and tilted 30° downward.



Figure 3.2: Experimental setup and viewpoints. Two wall-mounted radars (TV-side and window-side) observe quasi-static occupancy from different walls. The five annotated floor locations (P1–P5) indicate the positions where the subject lies or sits during each trial, covering near-wall, near-furniture, and open-floor regions.

For this study, we collected a dedicated dataset of semi-static activities in a long-term care facility using an off-the-shelf 60 GHz Millimeter-wave (mmWave) FMCW radar (Infineon XENSIV™ BGT60TR13C) [15] specified above (Section 2.1.1). The hardware and waveform parameters used in all recordings are summarized in Table 3.1. With this setting, the radar can offer a maximum detection range of 9.6 meters, which is enough to cover the measuring areas. Figure 3.1 is included only to illustrate the MIRADA room geometry and a representative setup view. The 2 m arrow shown in Figure 3.1 should therefore be interpreted as an approximate visual cue in the figure setup, not as the mounting height used in the recorded experiments. In this work, two of the radars are attached to the evaluation board as shown in Figure 2.3 and used for the actual recordings. They were wall-mounted at approximately 2.5 m above the floor on different walls (TV-side and window-side) with a slight 30° downward tilt to cover both the sofa and the open floor, while remaining out of reach for residents. This tilting is similar to that used in other works [50, 51]. Two radars are operated simultaneously with identical FMCW parameters from the same control PC. We define the timer and triggering for two radar starts to start simultaneously. However, for this study, these two sensors have been used individually, rather than com-

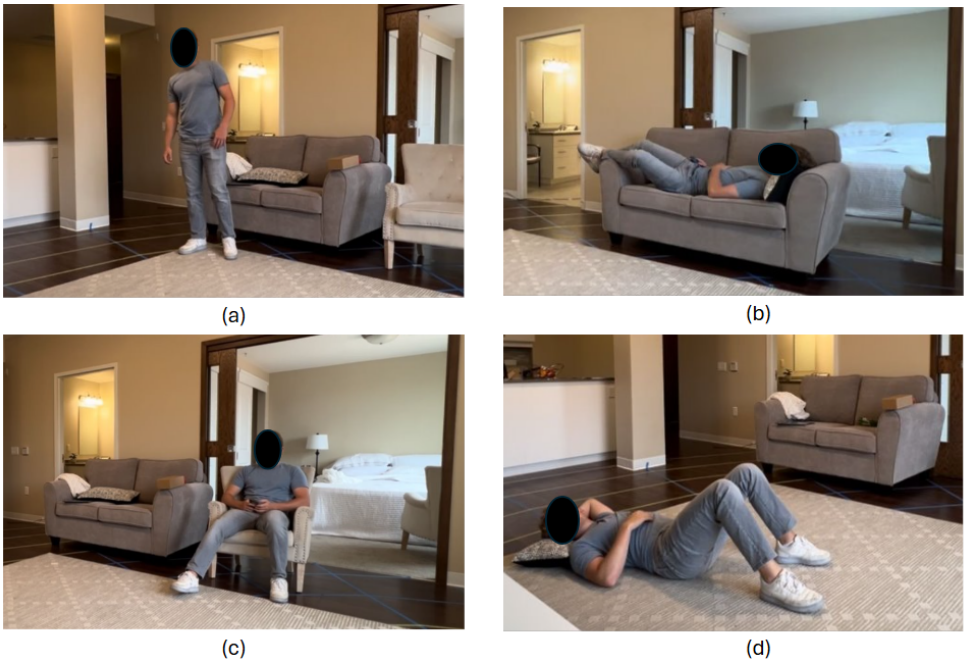

Figure 3.3: Few measurement activities: Standing (a), Lay on Sofa (b), Sitting (c), Lay on Floor (d).

bining them into a general view of the scenario. Thus, no strict synchronization between these two data streams. By doing this, our study will be independent of sensor placement, and both algorithms and models will be processed from various angles and views. The room contains common furniture (sofa, coffee table, TV, windows, and surrounding walls), inducing realistic multipath and occlusions. Unlike many "controlled" lab datasets, ours targets real-life LTC conditions: participants are not strictly constrained, and recordings include idle posture (lying on the floor/sofa, sitting on the floor/sofa, standing) as well as natural micro-movements (posture micro-adjustments while sitting/standing), and brief low-motion transitions (e.g., getting up from a sofa, picking an object from the floor/table) as shown in Figure 3.3. For each viewpoint, the subject is asked to perform in-place quasi-static activities at five floor locations (P1–P5), covering cases near the wall, near furniture edges, and open-floor regions. These positions are annotated as expected ground-truth regions and will be used for evaluation later in Section 3.1.4. For each activity, we recorded 2 minutes at 5 distinct positions and orientations relative to the sensors to vary the aspect angle, range, and background layout. Besides, for each measurement session, we placed this furniture randomly to create angular diversity to mimic different room layouts in the

LTC. This enables our final datasets to be very diverse and reflect real-world setups by residents. The radar frames were acquired continuously at 10 Hz, and the broader data collection has currently expanded to 8 participants, resulting in a total of approximately 384,000 frames. However, the experiments reported in this paper use only the first 4 participants, who were the subjects available when the study protocol and evaluation splits were finalized. Subject 1 is used for development/tuning, while Subjects 2–4 are used for held-out cross-subject evaluation. Participants 5–8 are not used in any results reported in this manuscript.

3.1.3 Proposed RASSO Algorithm

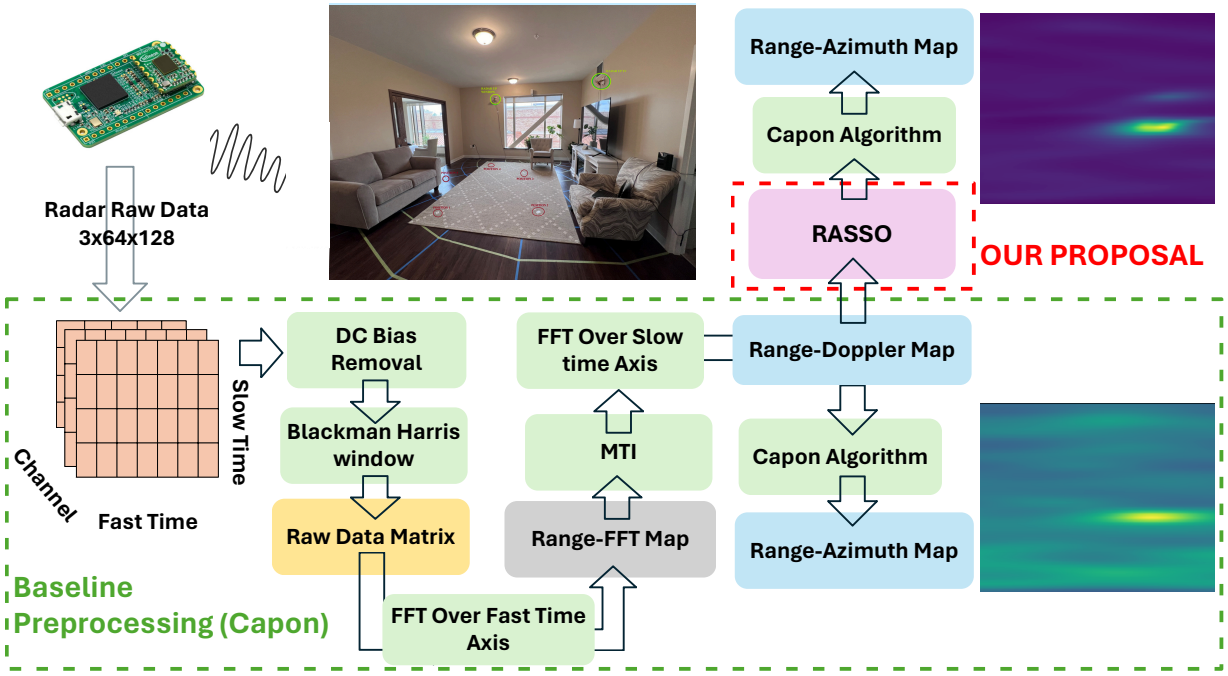

Figure 3.4: Our baseline signal processing pipeline (green-dash) [18], [19], [20], and our proposed RASSO (red-dash box)

Semi-/quasi-static human motions employ subtle postural changes, and micro-motions concentrate Doppler energy in the tight neighbourhood of zero frequency. With a uniform slow-time FFT, the Doppler grid uses the same resolution everywhere, resulting in the bin near $f \approx 0$ being too coarse for the dynamics we are interested in. We propose RASSO, a

wrapper that applies an invertible Doppler-domain non-linear warping and resampling of the Doppler axis before downstream spatial processing (Capon) and CFAR (Figure 3.1.3). Concretely, RASSO re-parameterizes each Doppler slice via the mapping $f \to D$ in (3.2)-(3.3) and re-samples the complex Range-Doppler (RD) data on a uniform grid in D as in (3.4)-(3.6), thereby allocating more samples around near-zero Doppler while preserving per-antenna phase.

Warped signal object After clutter suppression in Section 2.1.1, we obtain, for each frame k and receive antenna m, a MTI-filtered RD cube $Y_{k,m}(r, d)$ on the uniform Doppler-bin grid $d = 0, ..., N_D - 1$. Let f_d denote the Doppler frequency corresponding to bin d after the Doppler FFT. In the following, we drop the explicit frame index k and re-index the Doppler axis by its frequency:

$$Y_m[r, f_l] \triangleq Y_{k,m}(r, d_l), \tag{3.1}$$

where f_l is the Doppler frequency of bin d_l. RASSO operates only along this Doppler axis and leaves the $Y_m[r, f_l]$'s complex amplitude and per-antenna phase unchanged. RASSO does not modify the upstream FFT processing or downstream spatial processing; it instead, inserts an invertible mapping on the Doppler axis by replacing the uniform-f representation $Y_m[r, f_l]$ with a uniform-D representation $\hat{Y}_m[r, D_k]$, where $\{D_k\}$ is user-defined uniform grid in warped coordinate D.

Build the mapping grids Log-like warps are the standard way to densify near-zero and sparsify far frequency sampling. Compared with generic time-frequency re-allocations, we maintain a monotone, invertible mapping, ensuring a one-to-one representation and preserving phase relationships used by subsequent beamforming. We densify sampling in $f \le f_e$ (where the quasi-static energy is distributed), where $f_e > 0$ is the edge frequency, and compress the outer band, which is similar to the principle behind perceptual warping mel/log scales [52]. For even symmetry in magnitude, we add the sign back so that directions are preserved, and we scale by $\frac{1}{log2}$ to normalize the mapping. In addition, the mapping must be DC-safe, meaning there is no singularity at $f = 0$. Let $f \in [-F_{\max}, F_{\max}]$ denote the Doppler frequency of a given (r, m) slice (after FFT shift) on the uniform grid, and let D be the warped coordinate. We define this mapping and its inverse as:

$$D = \operatorname{sgn}(f)\tilde{g}(f, f_e) = \operatorname{sgn}(f)\ \frac{f_e}{\log 2}\ \log\Big(1 + \frac{|f|}{f_e}\Big), \tag{3.2}$$

$$f = \operatorname{sgn}(f)\tilde{g}^{-1}(D, f_e) = \operatorname{sgn}(D)\ f_e\big(2^{|D|/f_e} - 1\big)\,, \tag{3.3}$$

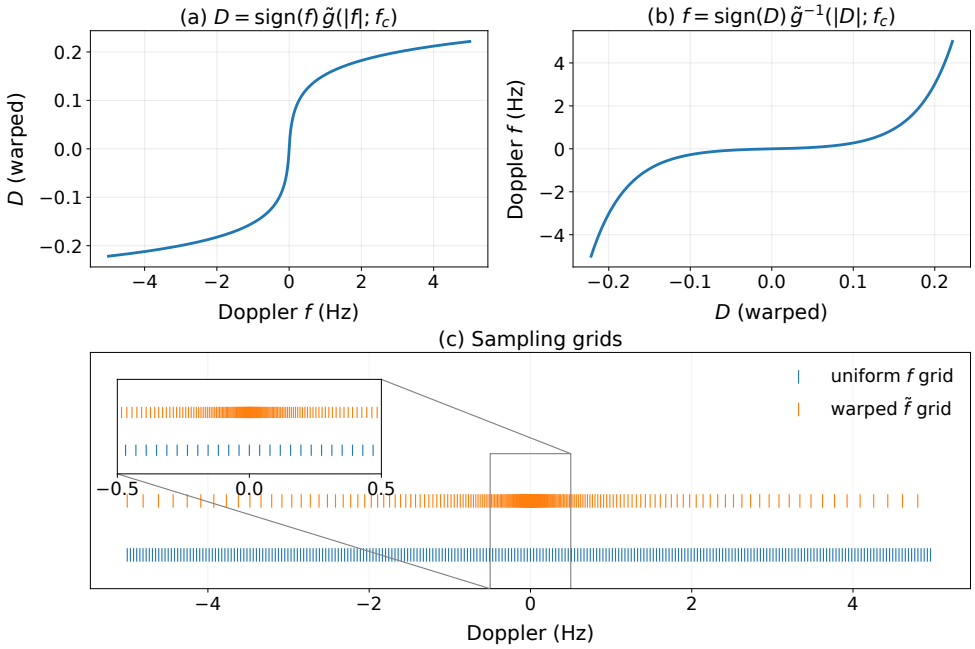

Figure 3.5: RASSO forward and inverse Doppler warp for edge/corner frequency f_c ($\equiv f_e$) corresponding to (3.2)–(3.3). Note that the corner frequency f_c shown is identical to the edge frequency f_e. Top: forward (left) and inverse (right) mapping between linear Doppler f and warped coordinate D. Bottom: uniform sampling in D (blue) induces denser sampling near $f = 0$ (zoomed view).

The forward map $D = \mathrm{sgn}(f)\tilde{g}(f, f_e)$ in (3.2) defines the warped coordinate, and is primarily used to visualize how Doppler is re-parameterized (Figure 3.5 (top left)). In implementation, resampling is driven by the inverse map $f = \mathrm{sgn}(f)\tilde{g}^{-1}(D, f_e)$ in (3.3). For each uniformly spaced D_k, we compute a generally non-uniform Doppler location $f_k = \tilde{g}^{-1}(D_k, f_e)$. In other words, RASSO does not warp and unwarp data, but it evaluates the original uniform-grid spectrum $Y_m[r, f_l]$ at non-uniform Doppler locations $\{f_k\}$ induced by uniform spacing in D, producing $\hat{Y}_m[r, D_k]$. Since the mapping is one-to-one, this prevents ambiguity in the conversion process between a continuous system and its discrete representations. In other words, the discrete–continuous–discrete step is stable and does not yield a significant amplification of errors in the discrete domain. Figure 3.5 (top left and right) visualizes (3.2) and (3.3) as the forward warp and the inverse used for resampling, respectively. For consistency, we use f_e throughout the text; in Figure. 3.5 the same quantity is labeled f_c, i.e., $f_c \equiv f_e$. The grid plot in Figure 3.5 (bottom) shows how uniform

steps in D (in blue) produce a non-uniform grid (in orange) in f, that is much denser near zero (see zoom insert).

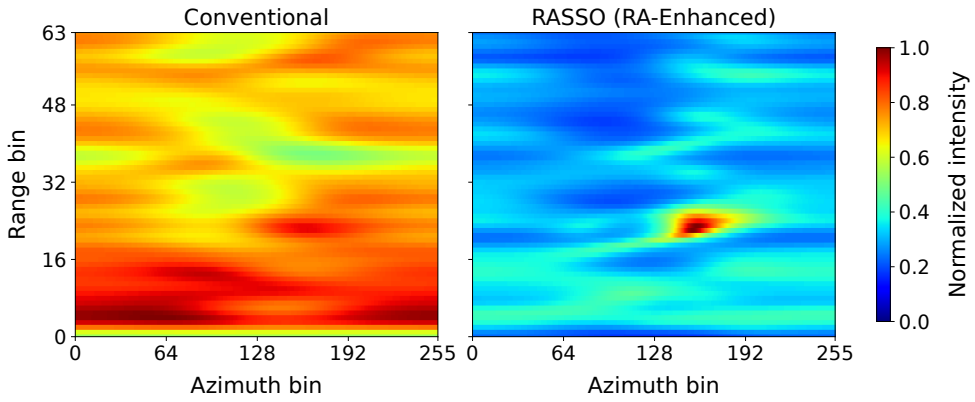

Figure 3.6: Range–Azimuth feature maps (frame 100) obtained from a Doppler-averaged Capon beamformer: (left) Conventional RA, (right) RASSO-enhanced RA. Both plots show power integrated over all MTI-filtered Doppler bins.

Complexed-valued resampling operator Since $Y_m[r, f]$ is only available on discrete FFT frequencies f_l, we approximate the continuous evaluation $Y_m[r, f_k]$ via 1D interpolation along Doppler. Specifically, for each range bin r and antenna m, after computing $f_k = \tilde{g}^{-1}(D_k, f_e)$, we apply vectorized linear interpolation to the real and imaginary parts separately as:

$$\mathcal{R}\{\tilde{Y}_m[r, D_k]\} \approx \text{Interp1D}(\{f_l\}, \mathcal{R}\{Y_m[r, f_l]\}, g^{-1}(D_k; f_e)) \tag{3.4}$$

$$\mathcal{I}\{\tilde{Y}_m[r, D_k]\} \approx \text{Interp1D}(\{f_l\}, \mathcal{I}\{Y_m[r, f_l]\}, g^{-1}(D_k; f_e)) \tag{3.5}$$

and then we compute $\tilde{Y}_m[r, D_k] = \mathcal{R}\{\cdot\} + j\mathcal{I}\{\cdot\}$ with zero-pads outside $[f_0, f_{N_D-1}]$. Since vectorized real/imaginary interpolation and Capon beamforming remain unchanged, the warp's complexity is $O(RN_DN_{RX})$ linear-time per frame. This ensures that RASSO reallocates Doppler sampling density without converting to magnitude, thereby preserving the per-antenna complex phase relationships that are required by subsequent beamforming/covariance estimation. In practice, wall-time overhead is dominated by the Capon covariance inversion. Finally, stacking over m and r yields the RASSO-enhanced RDC $Y[r, \tilde{D}_k, m]$

$$\widetilde{Y}_m[r, D] \;\approx\; Y_m\big[r,\; g^{-1}(D; f_e)\big]. \tag{3.6}$$

In other words, RASSO replaces each Doppler slice $Y_m[r, f]$ defined on the uniform FFT grid $\{f_\ell\}$ with $\tilde{Y}_m[r, D_k] \approx Y_m\big[r, g^{-1}(D_k; f_e)\big]$, i.e., an invertible non-linear change of coordinates on the Doppler axis, implemented via complex-valued interpolation, without modifying the underlying complex field. Figure 3.6 compares Baseline versus RASSO-Enhanced RA feature maps (frame 100) of Subject 3 lying on the floor. Unless otherwise stated, all Range–Azimuth maps in Figure 3.6, Figure 3.8 and Figure 3.14 are obtained from a Doppler-averaged Capon covariance, i.e., we average the MTI-filtered RD cube over all Doppler bins to estimate the 2×2 covariance per range bin before steering over angle. It can be seen from Figure 3.6 that after applying RASSO to the diffused Conventional map, the same scene exhibits a compact, blob-like target with suppressed low-angle smear, consistent with more Doppler samples being devoted to quasi-static before covariance estimation and beamforming.

Relation to standard alternatives RASSO operates at the Doppler re-parameterization level and is complementary to commonly used pre-processing strategies in quasi-static radar sensing.

- **Standard framewise RD processing:** performs detection or beamforming directly on single-frame Range–Doppler maps formed from a uniform slow-time FFT. In this case, Doppler resolution is uniformly distributed across the spectrum. RASSO, on the other hand, modifies near-zero representation by an invertible Doppler-axis re-parameterization before spatial processing.
- **Temporal integration of RDIs (e.g., RESPD / E-RESPD [49, 53]):** RESPD and E-RESPD perform a sliding-window summation of multiple consecutive Range–Doppler images to enhance weak respiration or subtle body motion cues. This constitutes a temporal accumulation pre-processing step applied after RD formation, aggregating energy across frames. In contrast, RASSO operates within each frame by re-allocating Doppler sampling density via an invertible mapping, without aggregating across time and without modifying the observation duration.
- **Zero-padding / spectral interpolation:** Zero-padding increases the number of FFT points and improves display resolution, but does not change how resolution is distributed across Doppler frequencies. RASSO instead performs a coordinate transformation that increases effective sampling density near $f \approx 0$ while compressing the outer band, implemented directly on complex Doppler slices to preserve phase for AOA estimation.

Overall, unlike temporal integration approaches that aggregate information across multiple frames, RASSO performs an invertible within-frame Doppler re-parameterization that preserves complex phase and downstream beamforming structure while specifically targeting quasi-static micro-Doppler content.

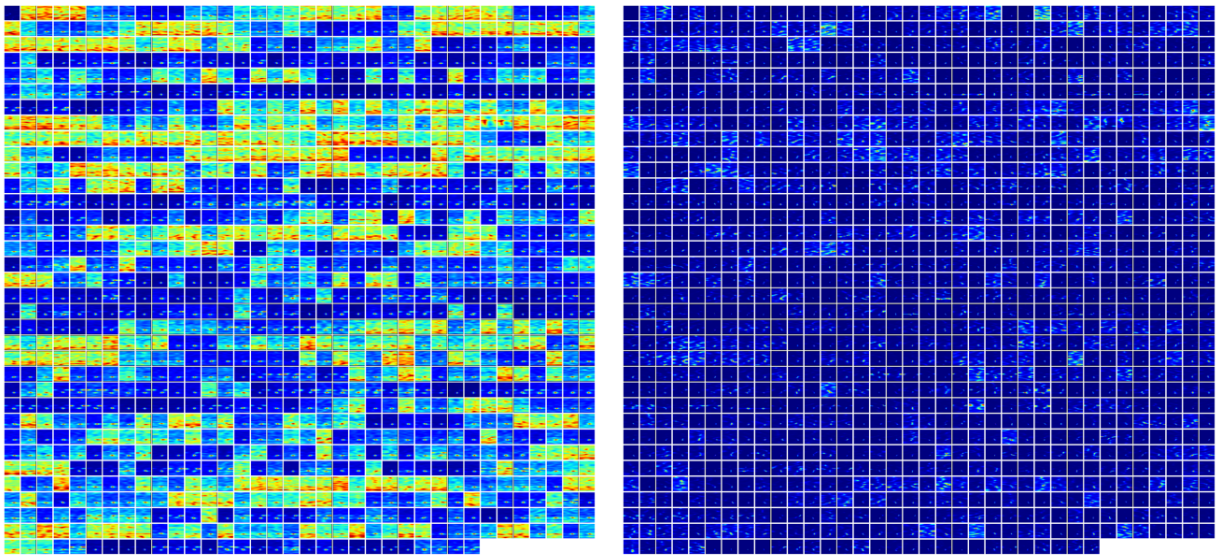

Figure 3.7: Range-Azimuth feature maps of 2-minute data collected when the subject was lying on the floor (a) before applying RASSO and (b) after applying RASSO. The RASSO-processed maps in (b) exhibit noticeably reduced background clutter and enhanced target energy, indicating an improved signal-to-noise ratio that supports more robust recognition.

3.1.4 Signal-to-Noise Ratio Analysis of Baseline and RASSO methods

Figure 3.7 first shows 1253 frames (approximately 2 minutes) of data of subject 1 lying on the floor to illustrate the effects of before and after RASSO qualitatively and demonstrate the shift from clutter-dominated to higher-SNR frames. We apply CA-CFAR and frame-level occupancy detections mentioned in Section 2.3.1 on RA feature maps obtained from 30 seconds (frame 300-600) data for subject 1 in the Lay On Floor scenario. To make qualitative comparisons meaningful, we render all RA maps using a standard colour scale after applying mean smoothing over a 5-frame window and $[0, 1]$ normalization. This is

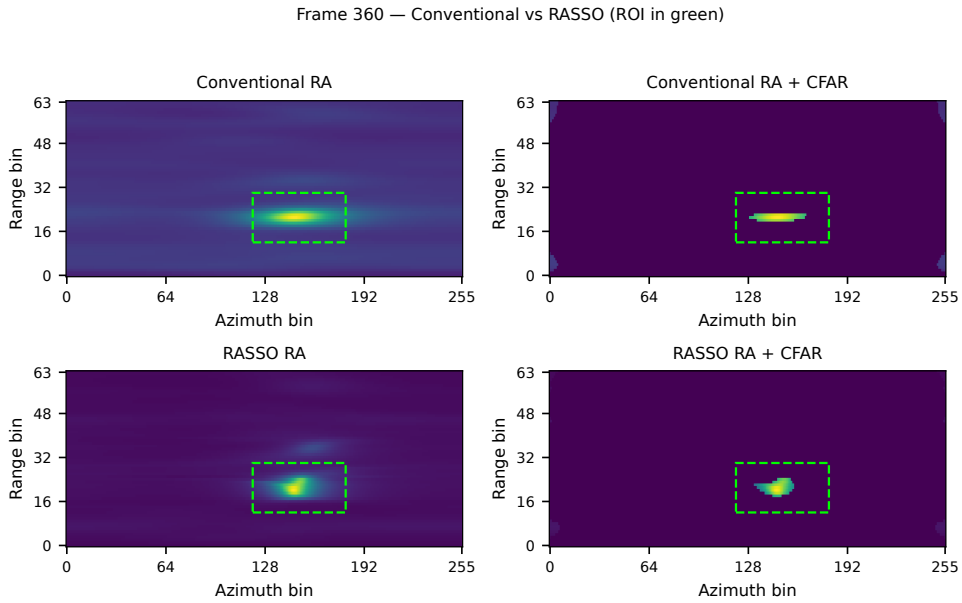

Figure 3.8: Conventional and RASSO-enhanced Range-Azimuth maps (frame 360) for a representative "Lay-On-Floor" frame (Subject 1). **Top row**: baseline RA before (left) and after CA-CFAR (right). **Bottom row**: RASSO RA before (left) and after CA-CFAR (right). All maps use normalized intensity, and the green dashed box marks the evaluation ROI around the lying subject.

the same normalization used in prior work (e.g., Hajar et al. [20]). Figure 3.8 shows the result of representative frame 360, and the subject's approximated position is annotated by a green dashed bounding box in the RA map. Since we record over 5 positions, these annotated ground-truth bounding boxes are changed accordingly in both range and angle bins. The detection is counted if the residual signal is located within the bounding box. We defined the True Positive Rate (TPR) as:

$$TPR = \frac{TP}{TP + FN} \tag{3.7}$$

with True Positives (TP) are frames where at least one CFAR cluster lies inside the annotated bounding box, i.e., the residual signal of the lying subject is correctly detected. False Negatives (FN) are frames where the subject is annotated as lying in the box, but no CFAR cluster is present inside it. The False Alarm Rate (FAR) is computed on empty–room recordings as the fraction of frames that contain at least one valid CFAR cluster anywhere in the RA map. We first tune the MTI forgetting factor α on the preprocessing baseline by

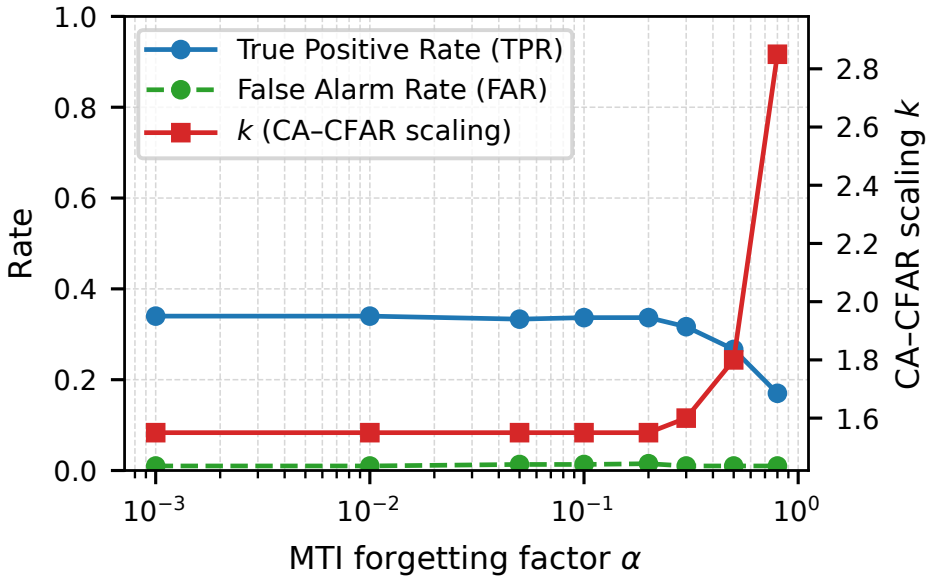

Figure 3.9: Effect of the MTI forgetting factor α on detection performance and CA-CFAR sensitivity on the development set (Subject 1). The curves show the true-positive rate (TPR), false-alarm rate (FAR), and the re-tuned CA-CFAR scaling k_{CFAR} (right axis) as α is swept; k in the legend corresponds to k_{CFAR}. The results show that MTI must be enabled and that performance remains nearly unchanged across a broad range of small nonzero α values, while larger α values reduce detection performance. The main takeaway is that performance is broadly stable for small nonzero α, whereas overly large α degrades performance; accordingly, $\alpha = 0.01$ is selected as a representative operating point. Based on this stable regime, $\alpha = 0.01$ is used in the remainder of the experiments.

sweeping through a very small α to balance between very conservative DC cluster removal, as well as keep our quasi-static signal alive to more common values used in dynamic activity recognition settings (0.001, 0.01, 0.1, 0.2, 0.3, 0.5 and 0.8). We set the cap for FAR to be 0.01 empty-room set ($\mathrm{FAR}_{\mathrm{target}}$=0.01). For each α, we re-tune CFAR sensitivity k_{CFAR} on empty-room data to meet a fixed false-alarm rate $FAR \approx 0.01$, then we report the z-score SNR inside the bounding box and TPR on the data frames. We then measure (i) the TPR on these tuning frames using the bounding-box criterion and (ii) the z-score SNR in Eq. 3.8 inside the bounding box. Figure 3.9 shows the TPR, FAR and k_{CFAR} results as α is swept. The main observation is that detection performance remains broadly stable for small nonzero α, whereas overly large α degrades performance. In contrast, disabling MTI

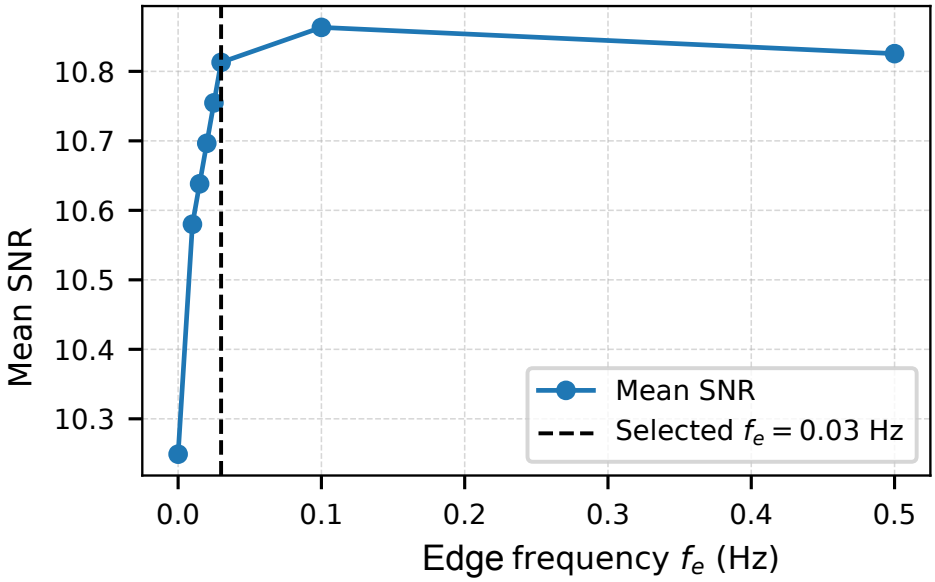

Figure 3.10: Selection of the Doppler edge frequency f_e for the RASSO mapping. The mean z-score SNR inside the bounding box (Eq. (3.8)) is plotted versus f_e; the elbow around $f_e = 0.03$ Hz (dashed line) is chosen as the operating point in all experiments.

($\alpha = 0$) leads to an unrecoverable FAR of 1.0 even after re-tuning k_{CFAR}, confirming that MTI is essential for suppressing static room clutter. Since the small-α regime does not show a sharp optimum, we use $\alpha = 0.01$ in all experiments as a representative operating point. Once we fixed forgetting $\alpha = 0.01$ for the conventional preprocessing pipeline, we perform a one-dimensional sweep to choose the edge frequency f_e in the RASSO mapping. We evaluate a grid of edge/corner frequencies $f_e \in \{0, 0.01, 0.015, 0.02, 0.025, 0.03, 0.1, 0.5\}$ Hz and report the mean z-score SNR inside the bounding box (Eq. (3.8)) in Figure 3.10. The curve exhibits an "elbow" point at around $f_e \approx 0.03$ Hz: increasing f_e from 0 to 0.03 Hz yields a noticeable SNR gain, while larger values only provide marginal improvements and begin to emphasize higher-frequency fluctuations. We therefore fix $f_e = 0.03$ Hz for all experiments as a representative operating point in this plateau region, rather than as a finely tuned optimum. Finally, using the background ring $\mathcal{N}$ (the CFAR training set excluding $\mathcal{B}$), we define

$$\mathrm{SNR}_z = \frac{\overline{Y}_{\mathcal{B}} - \overline{Y}_{\mathcal{N}}}{\sigma_{\mathcal{N}}}, \tag{3.8}$$

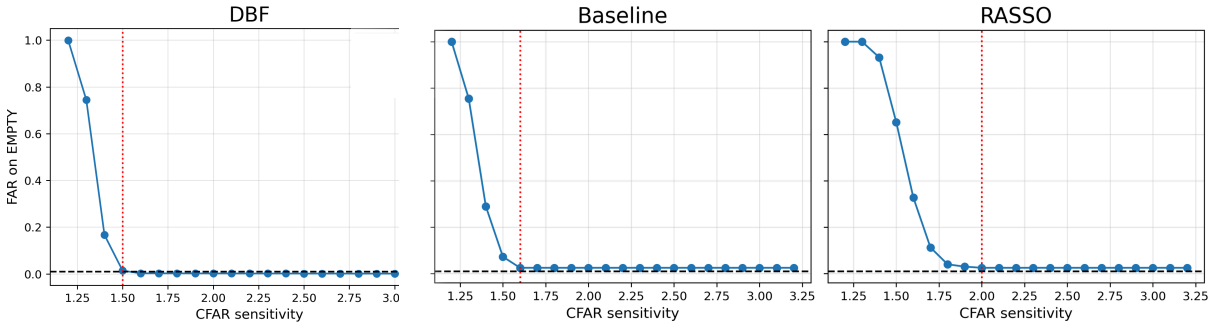

Figure 3.11: False–alarm rate (FAR) on empty-room sequences as a function of the CA–CFAR sensitivity k_{CFAR} for the three processors: (**left**) SDK Digital Beamforming (DBF), (**middle**) our Conventional baseline, and (**right**) our RASSO-enhanced pipeline. Curves are computed on negatives only (no person present), after MTI (α=0.01), five-frame mean smoothing, $[0, 1]$ normalization, and CA–CFAR with guard G=4 and train T=10. Dots indicate individual sequence measurements; the solid line is the mean FAR. The red vertical dashed line marks the operating point chosen for each method (smallest k_{CFAR} at which mean FAR meets the target constraint used throughout the paper).

where $\overline{Y}_{\mathcal{B}}$ is the mean intensity inside the bounding box, $\overline{Y}_{\mathcal{N}}$ and $\sigma_{\mathcal{N}}$ are the mean and standard deviation of the surrounding background ring. This z-score formulation is robust to slow gain drifts and only depends on the contrast between the target region and its local clutter. As shown in Figure 3.12, the RASSO curve is consistently higher than the conventional RA curve, indicating stronger concentration of energy within the bounding box and reduced background variability. Over this 30–60 s interval in Figure 3.12, the average SNR_z increases from 0.62 (conventional RA) to 1.36 (RASSO RA), corresponding to approximately a 2.2× improvement in target-to-background separation.

3.1.5 Digital Beam Forming comparison

We compare our baseline pipeline with and without RASSO against DBF from Infineon's SDK. All processors share the same pre- and post-processing. To keep up with the compatibility with our conventional pipeline used in [18] and compare to the Digital Beamforming pipeline provided (which also uses MTI), we use the same MTI as described in Section 3.1.4. For a fair comparison, each method's CFAR sensitivity k_{CFAR} is tuned only on negative (empty-room) sequences and sweep k_{CFAR} on a grid and choose k_{CFAR} that minimizes $|\mathrm{FAR}(k_{\mathrm{CFAR}}) - \mathrm{FAR}_{\mathrm{target}}|$ on the pooled empty-room set ($\mathrm{FAR}_{\mathrm{target}}$=0.01) for

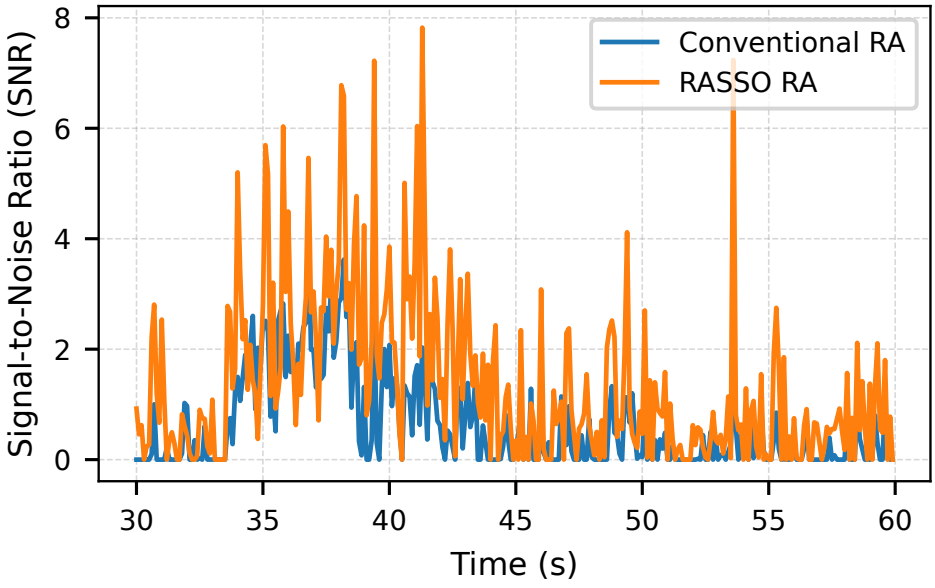

Figure 3.12: Frame-wise z-score SNR for the Lay-On-Floor sequence (Subject 1), comparing conventional RA and RASSO RA over a 30–60 s interval. RASSO RA yields a consistently higher SNR, indicating stronger target concentration and reduced background variance.

all three processors. Balanced Accuracy (BA) is then computed on the evaluation set as $\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$.

Representative frames are shown in Figure 3.14 (left column). In the DBF baseline, the energy blob associated with the subject typically spans a broader set of azimuth bins, indicating a less concentrated angular response. Our baseline processor, however, yields a visibly tighter lobe, and RASSO further concentrates the energy, reducing the angular spread and sidelobe floor. These effects are consistent across activities we inspected and across rooms. Using two empty-room files and very small pilot activity files for demonstrating proof-of-concept, the sensitivity values selected by the above protocol were within the ranges illustrated in Figure 3.14 (right column) and Figure 3.13. We evaluated these matched-FAR operating points on our minimal pilot dataset for proof-of-concept and reported BA rather than raw accuracy. BA is the average of the true-positive rate (occupied) and the true-negative rate (empty). RASSO achieves the highest BA, followed by our baseline processor, and the SDK DBF's accuracy is the lowest. It is consistent with the FAR-sensitivity sweeps (Figure 3.11), where RASSO reaches the low-FAR regime at milder sensitivity than DBF.

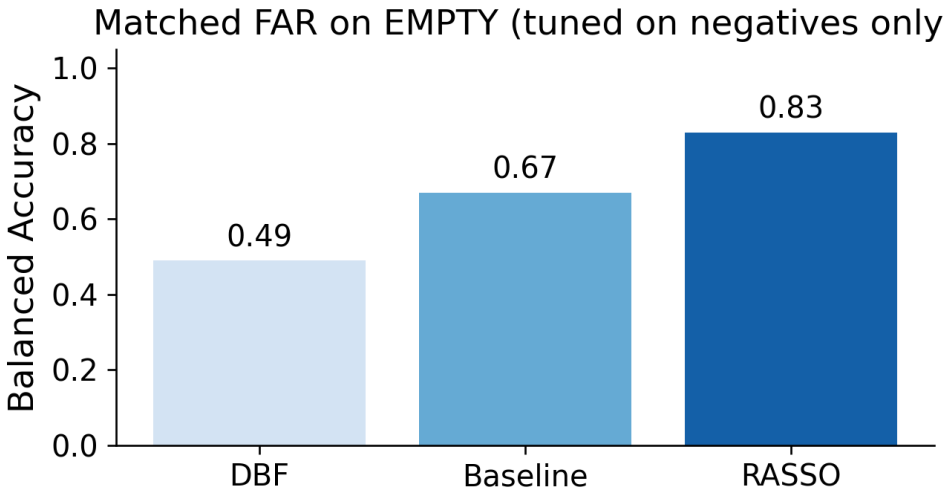

Figure 3.13: Balanced Accuracy (BA) at matched-FAR operating points for SDK DBF, our Conventional baseline, and our RASSO-enhanced processor. For fairness, the CA–CFAR sensitivity k_{CFAR} for each method is tuned on empty-room sequences only to meet the same FAR target as Figure 3.11.

3.1.6 Benchmarking of RASSO against E-RESPD

In this section, we will compare the efficacy of RASSO on our baseline against the E-RESPD method introduced in Section 3.1.1. Fixing a single CA-CFAR threshold k_{CFAR} might not be a fair way to compare different processing chains: each feature map (HOOD E-RESPD Macro/Micro, our Conventional RD/RA, and RASSO-enhanced RA) has different normalization, dynamic-range compression, and residual clutter statistic. In addition, the same k_{CFAR} does not imply the same false-alarm probability $\overline{P}_{\mathrm{FA}}$ [26, Section 7.3]. Specifically, we sweep k_{CFAR} on a dense grid and trace recall versus FAR curves on a per-frame basis. The Area Under the Curve (AUC) plays the same role as AUROC in the MCROOD/HOOD papers [49, 53]. For each k_{CFAR} we count over all frames:

- **Recall (TPR):** fraction of human presence file frames with ≥ 1 valid cluster;
- **FAR**: fraction of empty room files frames with ≥ 1 valid cluster.

Figure 3.16 compares and visualizes Receiver Operating Characteristic (ROC)-like lines and trapezoidal area for the above 6 methods. We also report recall at practical operating

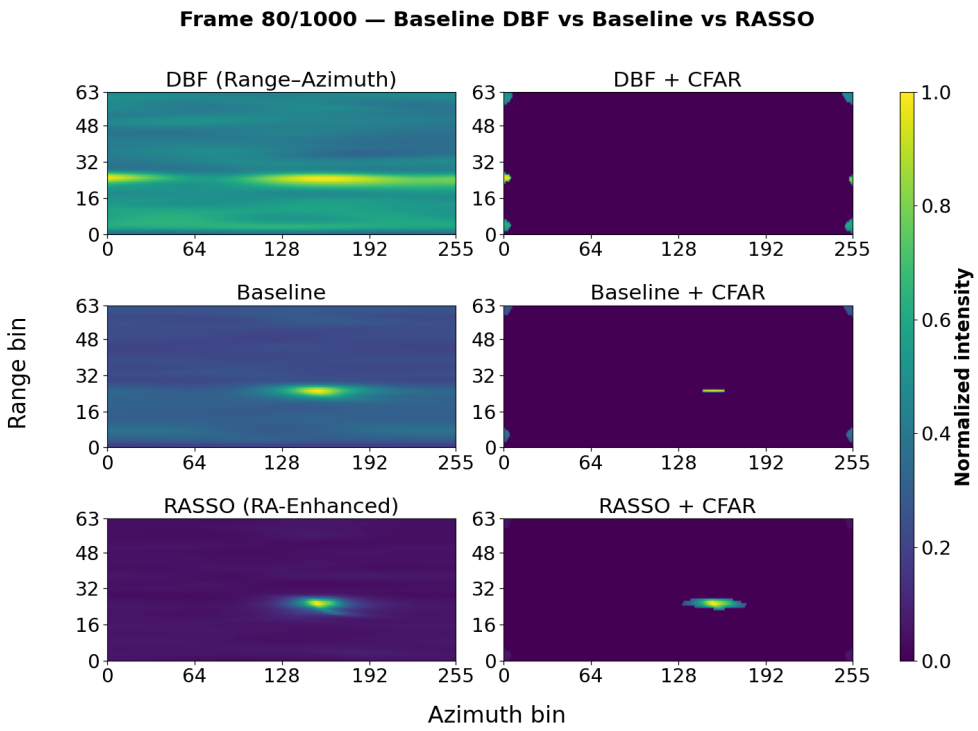

Figure 3.14: (**Left Column**) Range-Azimuth feature map (frame 80) of the subject lying on the floor obtained by (**Top**) Digital Beam Forming (DBF), (**Middle**) Our Conventional, (**Bottom**) RASSO enhanced on Our Conventional. The right column shows their corresponding CFAR detection.

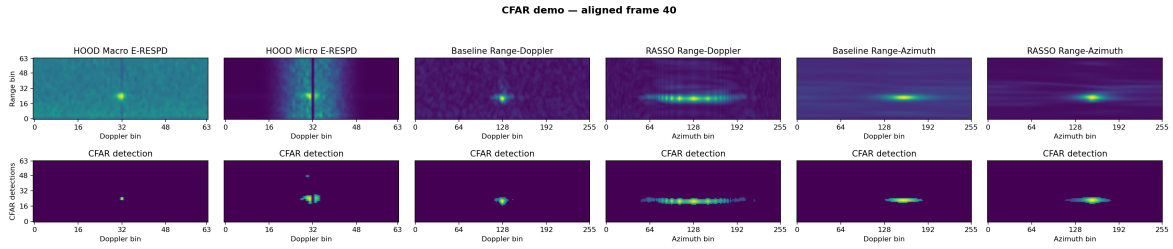

Figure 3.15: Visualization of 6 feature maps on first row and their corresponding CFAR detections on second row (from left to right): Macro E-RESPD, Micro E-RESPD, Baseline Range Doppler, RASSO-enhanced Range-Doppler, Baseline Range-Azimuth, RASSO-enhanced Range-Azimuth.

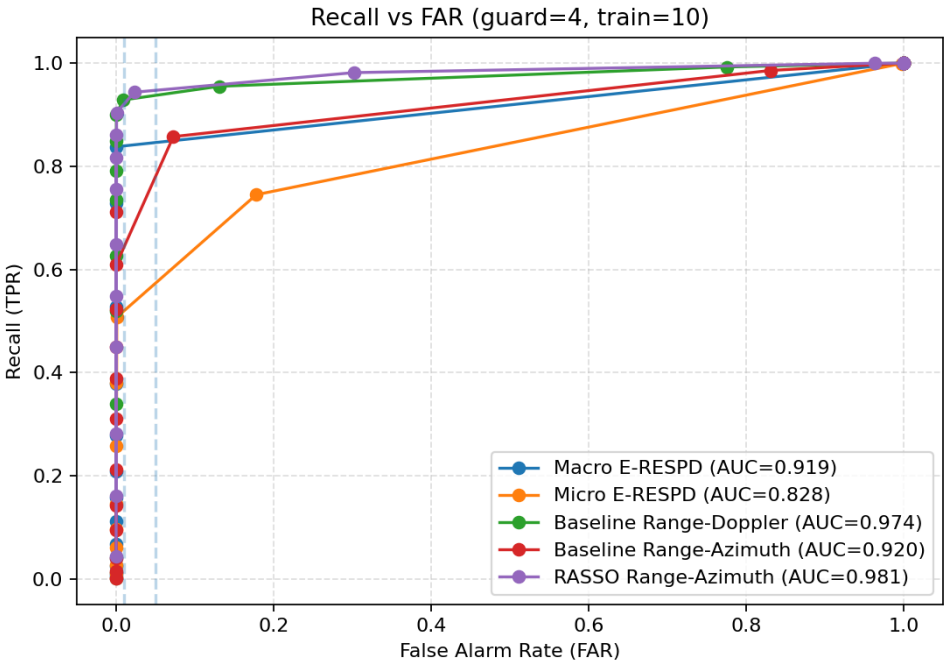

Figure 3.16: Recall–FAR characteristics for five RA/RD feature maps: HOOD Macro E-RESPD, HOOD Micro E-RESPD, Baseline Range–Doppler, Baseline Range–Azimuth, and RASSO Range–Azimuth. For each feature map, we sweep the CA–CFAR sensitivity parameter k_{CFAR} on the development subject and, for every value of k_{CFAR}, compute the corresponding recall and false–alarm rate (FAR). The legend reports the area under the Recall–FAR curve (AUC). Each curve corresponds to a single feature map, and different points along the same curve are obtained for different values of k_{CFAR}. Light blue vertical dashed lines indicate reference operating regions at FAR = 1% and 5% used for pointwise comparisons. The recall values at these FAR levels are reported in Table 3.2.

points (FAR = 1% and 5%) as in Table 3.2. As illustrated in Figure 3.16, sweeping the CA-CFAR parameter k_{CFAR} yields a recall–FAR trade-off curve for each feature map. For a given FAR value, different methods will in general correspond to different k_{CFAR}, since k_{CFAR} is tuned independently per feature map. For completeness with the RD-based E-RESPD literature, Figure 3.16 also includes the Macro and Micro E-RESPD curves together with our conventional RD baseline. As seen in the AUC values, the RD baseline's result is slightly higher than the RA baseline but remains below the RASSO RA curve in the operating FAR region of interest. However, our deployment target is

quasi-static and semi-static occupancy monitoring with a low-cost radar, where angular localization on the floor plane (who and where in the room) is more critical than precise radial velocity. Consequently, all remaining ablations and cross-subject evaluations in this and the following sections are performed on RA maps only, comparing our baseline RA processor against its RASSO-enhanced variant. Using this evaluation, our RASSO-RA front-end enhances the Range-Angle feature map method and overtakes the HOOD baselines and our conventional variants. It achieves the highest AUC (**0.981**) and highest Recall at both FAR targets (**0.920** @1%, **0.947** @5%). This indicates that warping near-zero Doppler to reduce DC before spatial processing (RASSO), followed by adaptive MVDR beamforming, improves separability at low FAR.

Table 3.2: Summary of AUC of the Recall–FAR curve and Recall at two fixed false-alarm operating points (FAR = 1% and 5%) among 5 methods. Best values per column are shown in **bold**.

Method	AUC	Recall@FAR=1%	Recall@FAR=5%
Macro E-RESPD	0.919	0.839	0.845
Micro E-RESPD	0.828	0.520	0.573
Baseline RD (Range–Doppler)	0.974	0.829	0.837
Baseline Range-Azimuth	0.920	0.643	0.780
RASSO Range-Azimuth	**0.981**	**0.920**	**0.947**

Our baseline RD also yields AUC 0.974 with high Recall at 1–5% FAR. In Table 3.2, the row Baseline RD refers to this same RD baseline introduced in Section 2.1.1. The Macro and Micro E-RESPD achieve AUCs of 0.919 and 0.828, respectively. These curves motivate using RASSO-RA as the feature map for the remainder of the paper.

3.1.7 Cross-subject Evaluation

While a k_{CFAR} sweep is suitable for benchmarking, deployment requires sticking to an operating point. We therefore conduct a RASSO ablation study, sweep k_{CFAR} on a development subject (Subject 1) and evaluate that fixed k_{CFAR} on unseen subjects (Subjects 2–4) recorded in the same living room from different vantage points. We select k_{CFAR} to maximize macro-F1 (unweighted mean of class F1 for empty room and 1 person) to balance

the misses and false alarm rate without privileging one class. Similar to our DBF comparison study in the Section 3.1.5, our baseline pipeline induces higher sensitivity $k_{\text{CFAR}} = 1.6$ with RASSO and $k_{\text{CFAR}} = 1.2$ without RASSO. The results are demonstrated in Table 3.3. For completeness, we compute the per-class F1-score from the class precision P and recall R as

$$F_1 = \frac{2PR}{P+R}.$$

Throughout the paper, the reported Macro-F1 is the unweighted mean of the class-wise F1 scores for the two classes (empty and person-present). It can be seen from the Table 3.3

Table 3.3: CA-CFAR Evaluation Across Subjects ($k_{\text{CFAR}} = 1.2$ for Baseline Range-Azimuth and $k_{\text{CFAR}} = 1.6$ for RASSO-Enhanced). Best values for each F1 metric per subject are shown in **bold**.

Subject	**Variant**	**F1 Macro**	**F1 (empty)**	**F1 (1-person)**
Subject 1 (Development Set)	Baseline RA	0.843	0.771	0.915
	RASSO RA	**0.932**	**0.897**	**0.967**
Subject 2 (Generalization Test Set)	Baseline RA	0.811	0.653	0.970
	RASSO RA	**0.924**	**0.857**	**0.990**
Subject 3 (Generalization Test Set)	Baseline RA	0.842	0.720	0.964
	RASSO RA	**0.948**	**0.907**	**0.990**
Subject 4 (Generalization Test Set)	Baseline RA	0.801	0.652	0.950
	RASSO RA	**0.863**	**0.863**	**0.985**

that RASSO-RA consistently lifts performance for every subject and for both classes, resulting in an improvement of F1 Macro of $\approx 6\%$ to 10%. Crucially, $F1_{empty}$ increases markedly, confirming that RASSO suppresses quasi-static clutter and clarifies the spatial covariance used by CFAR. With RASSO-enabled, across subjects, $F1_{1-person}$ remains very high (typically ≥ 0.97), but $F1_{empty}$ are slightly lower, yielding macro-F1 range of 0.85–0.9 for empty room class. In privacy-preserving monitoring for nursing homes and long-term-care facilities, both missed detections (false negatives) and false alarms incur high costs. Missed detections can delay assistance, excessive false alarms drive alarm fatigue and affect trust in the system. A deployable solution, therefore, needs simultaneously high recall and low FAR with real-time latency and generalizes well for different residents.

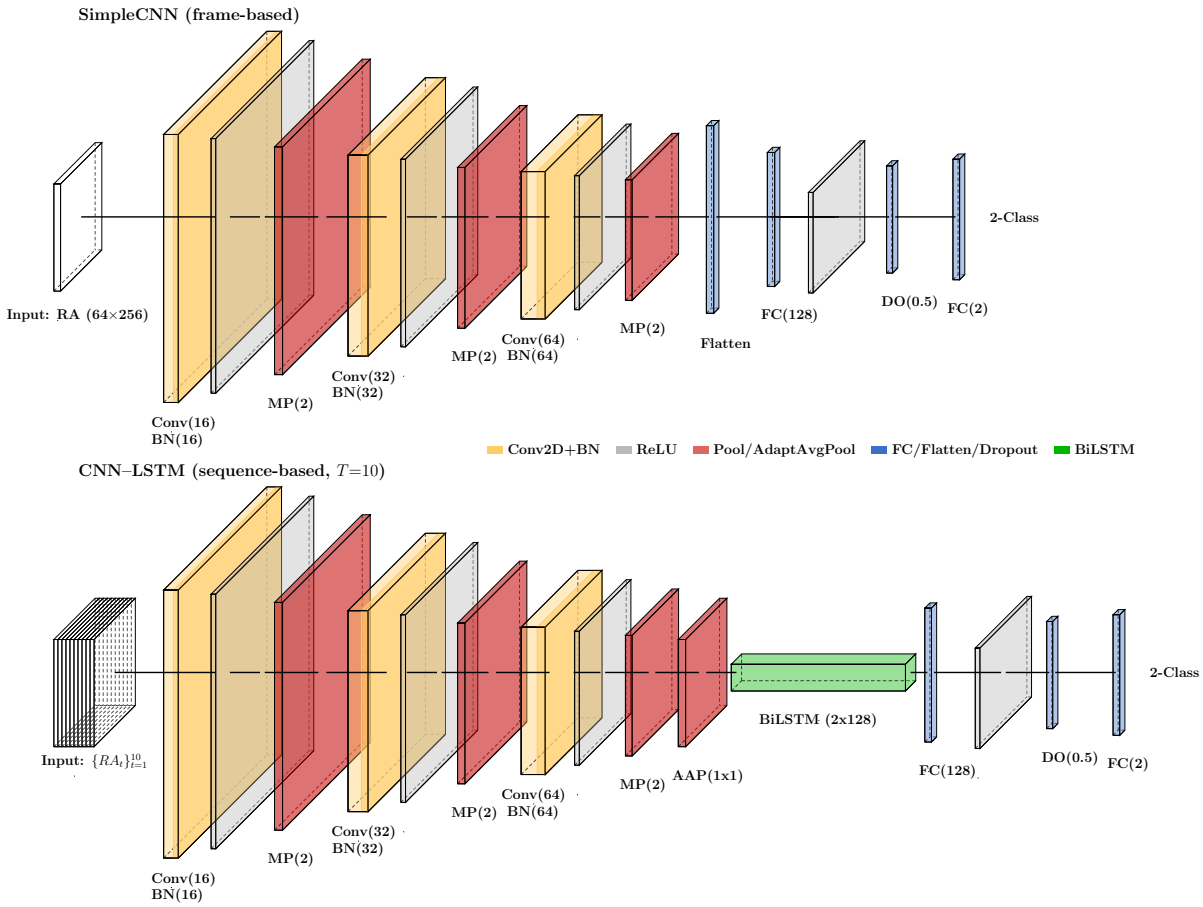

Figure 3.17: Architecture of models (top): SimpleCNN and (bottom) CNN-LSTM.

3.1.8 Data-Driven Classification

To reduce these nuisance detections while retaining high sensitivity to human micro-motion, we complement the model-driven pipeline with lightweight, data-driven classifiers trained directly on the same RA maps. RASSO reshapes the Doppler axis and suppresses near-static clutter, producing Range-Azimuth maps that are sparser with higher output SNR as shown in Figure 3.12. These properties are ideal for learning: the discriminative signal is concentrated, while the nuisance structure is reduced. Thus, we design our own 2 models that reveal the impact of RASSO on learnability and generalization.

SimpleCNN (frame-based) model The input is a single normalized RA heatmap (64×256). We first implemented a CNN based model called **SimpleCNN** for each frame classification. Its architecture is visualized as Figure 3.17 (top). The model comprises a very shallow depth of three 3×3 convolutional blocks with channel width $[16, 32, 64]$, each followed by batch normalization layers, ReLU activation functions and 2×2 max-pooling. We trained SimpleCNN using Adam optimizer, early stopping on validation loss, a 128-unit fully connected layer (ReLU, dropout $p = 0.5$ to reduce overfitting) precedes 2-way softmax. For the training procedure, subject 1's dataset is split 80 %/10 %/10 % into training/validation/test for held-out evaluation, while all subjects 2-4 data are used for cross-subject generalization. The label space is binary $empty_room, person_present$ and performance is reported as overall accuracy and per-class F1 (harmonic mean of precision and recall), which balances false alarms and misses. Table 3.4 summarizes the frame-based results. Across all subjects, RASSO-based RA maps consistently surpass the Baseline RA in terms of accuracy and F1, with the largest gains observed on the challenging cross-subject setting, where RASSO improves F1 (person-present) while maintaining high specificity on empty frames. These results indicate that the proposed Doppler warping not only sharpens classical CFAR-based detection but also provides more discriminative inputs for lightweight CNN classifiers without changing the network architecture or training protocol.

Table 3.4: RASSO Results – Frame Based (SimpleCNN)

Subject	**Variant**	**Acc. (%)**	**F1 (empty)**	**F1 (1-person)**
Subject 1 (Held-out Test Set)	Baseline RA	0.9100	0.9221	0.8934
	RASSO RA	**0.9536**	**0.9582**	**0.9252**
Subject 2 (Generalization Test Set)	Baseline RA	0.9323	0.8509	0.9562
	RASSO RA	**0.9687**	**0.9252**	**0.9802**
Subject 3 (Generalization Test Set)	Baseline RA	0.9250	0.8814	0.9451
	RASSO RA	**0.9678**	**0.9456**	**0.9771**
Subject 4 (Generalization Test Set)	Baseline RA	0.8939	0.8394	0.9208
	RASSO RA	**0.9942**	**0.9899**	**0.9960**

CNN-LSTM (sequence-based) model Semi-static occupancy often manifests as very low-Doppler micro-motions (postural sway, subtle limb readjustments, respiration) that unfold across multiple frames. Thus, classification based on individual frames may miss

Table 3.5: RASSO Results – Sequence Based (CNN-LSTM)

Subject	Variant	Acc. (%)	F1 (empty)	F1 (1-person)
Subject 1 (Held-out Test Set)	Baseline RA	0.9561	0.9607	0.9502
	RASSO RA	**0.9845**	**0.9857**	**0.9831**
Subject 2 (Generalization Test Set)	Baseline RA	0.9772	0.9449	0.9856
	RASSO RA	**0.9948**	**0.9868**	**0.9967**
Subject 3 (Generalization Test Set)	Baseline RA	0.9717	0.9522	0.9799
	RASSO RA	**0.9942**	**0.9899**	**0.9960**
Subject 4 (Generalization Test Set)	Baseline RA	0.9652	0.9415	0.9752
	RASSO RA	**0.9956**	**0.9921**	**0.9969**

these cues. LSTM [54] are expressly designed to learn long/short-term dependencies in sequences. For a sequence-based experiment, we use CNN as a feature extractor that learns spatial patterns in the Range–Azimuth feature maps; its output is temporally aggregated by a bidirectional Long Short-Term Memory head, followed by a linear classifier. We stack 10 consecutive frames, which corresponds to 1 second of data, as a chunk with a stride of 1 as visualized in Figure 3.17 (bottom). Table 3.5 demonstrates that RASSO improves the accuracy of all datasets, with a narrow high-performance band of 99.4–99.6%, and both F1 scores ≥ 0.983.

3.1.9 Uncertainty Quantification (CNN-LSTM, Subject 2-4)

For each generalization subject (2–4) and each model (Baseline and RASSO-Enhanced, both trained on Subject 1; see Section 3.1.8), we draw $B = 1000$ bootstrap replicates. For each replicate b, we compute Accuracy, the per-class F1 scores (empty_room, person_present), and the macro-averaged F1. We report the point estimate from the full (non-resampled) test set, and a 95% Confidence Interval (CI) given by the $[2.5, 97.5]$ empirical percentiles of the B bootstrap values. The Figure 3.18 analyzes the uncertainty of Macro-F1 for these subjects; each point indicates a session-level bootstrap replicate, and the circles and whiskers show the full-set estimate and 95% CI. RASSO-Enhanced (orange) swarms consistently outperform Conventional (blue) ones across Subjects 2-4, with tighter CIs near unity. As a result, RASSO shifts Macro-F1 upward by $2.6-3.6$ percentage points of the median on generalization subjects, and its 95% interval is higher than 98% Macro-F1.

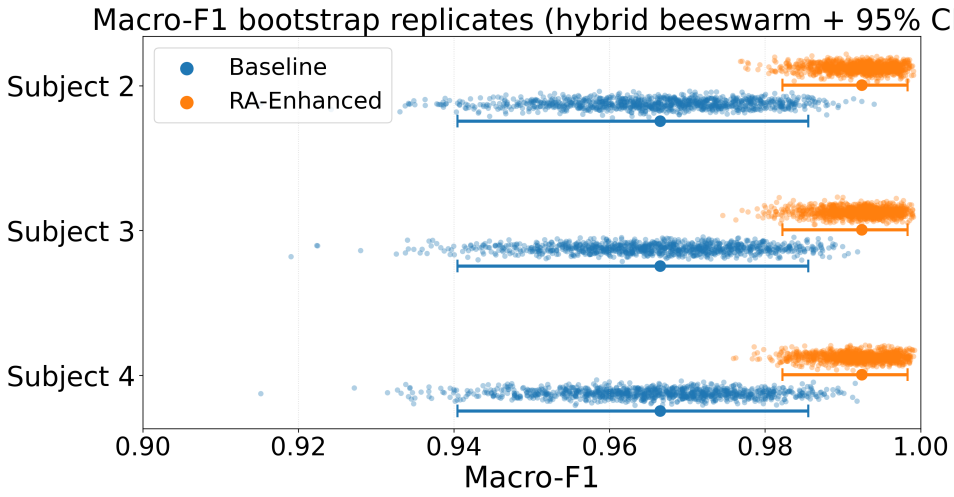

Figure 3.18: Macro-F1 bootstrap replicates (hybrid beeswarm) with 95% CIs for Subjects 2–4. Each dot is one session-level bootstrap replicate. The solid dot and whiskers are the full-set estimate and its 95% CI. Orange (RASSO-Enhanced) consistently exceeds blue (Baseline).

Figure 3.19 presents the per-class behaviour under bootstrap joint scatter of F1 (empty room) versus F1 (person present) for Subjects 2-4. RASSO-Enhanced (orange) concentrates on the upper-right frontier, indicating simultaneous gains in both classes. This suggests that RASSO helps reduce both missed detections in person-present cases and false alarms in empty-room cases. Finally, we compute the paired bootstrap replicate differences for the same sessions for each replicate $r = 1...1000$ in the same set, as visualized in Figure 3.18.

$$\Delta_r = F_{1,macro}^{\text{RASSO}} - F_{1,macro}^{\text{Conv}}. \tag{3.9}$$

The Figure 3.20 presents the distribution of paired differences Δ = Macro-F1$_{\text{RASSO}}$ − Macro-F1$_{\text{Conv}}$ for each subject. It can be seen from the figure that for all three subjects, their 95% CI lies above zero, confirming statistically significant improvements (empirical $p \leq 0.001$ with 1000 resamples). The median and low/high points of the Macro-F1 differences are 0.026 [0.012, 0.043], 0.026 [0.011, 0.044], and 0.036 [0.020, 0.055] for subjects 2, 3, and 4, respectively.

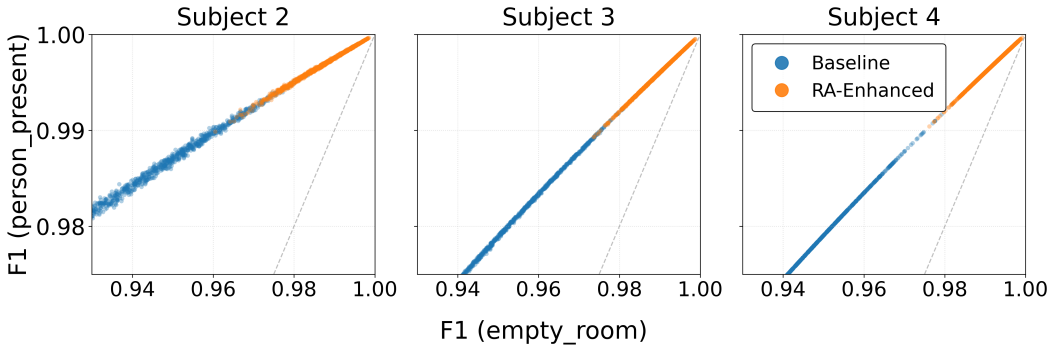

Figure 3.19: Per-class uncertainty. Bootstrap join scatter of $F1_{\text{empty}}$(x) vs. $F1_{\text{person}}$(y). The grey diagonal marks equal per-class performance.

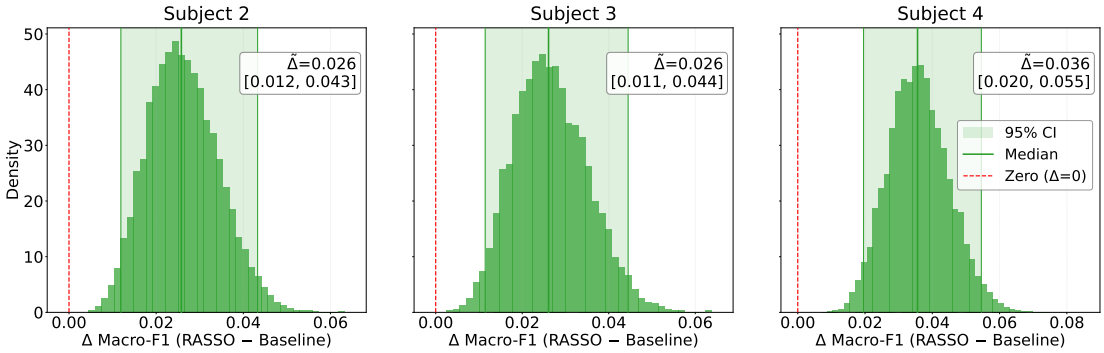

Figure 3.20: Paired improvement distributions. Histograms of Δ = Macro-$F1_{\text{RASSO}}$ − Macro-$F1_{\text{Baseline}}$ with median (solid green), 95% CI (shaded band), and a red dashed line at zero.

3.2 Post-Fall Floor-Occupancy Detection

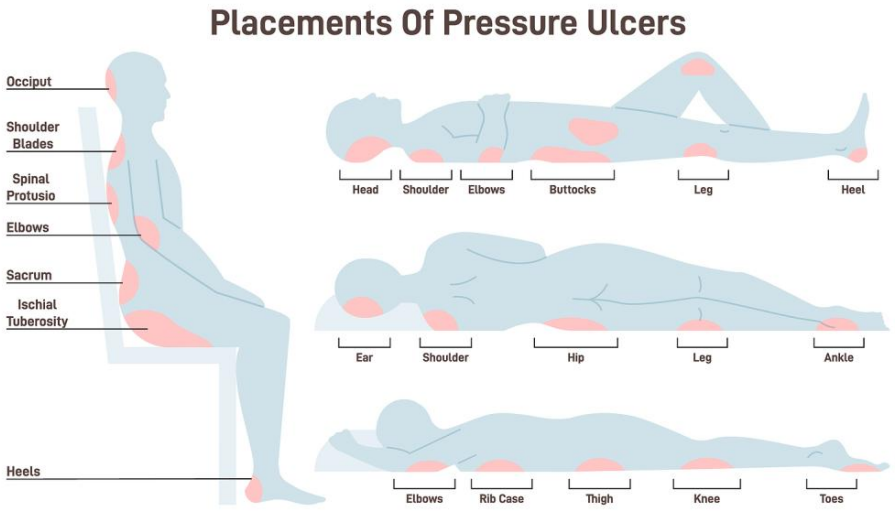

Figure 3.21: Placements of pressure ulcers [55].

As the population ages rapidly, LTC facilities across North America face growing pressure to monitor residents safely while keeping staff workload manageable. Fall incidents remain a major cause of injury and loss of independence among older adults, leading to emergency visits and long-term care admissions [56]. In clinical settings, the term "long-lie" is often referred to a situation in which an individual has fallen and remains on the floor for one hour or longer [57–60]. Long-lies can cause hypothermia, dehydration, pressure ulcers, muscle damage, infections, and even loss of consciousness [60]. Figure 3.21 is included as clinical motivation: prolonged pressure in a few contact regions can cause tissue damage, which illustrates why timely detection of sustained floor occupancy matters. Most of the current fall-detection literature focuses on event detection (classifying a fall at the moment it occurs) using wearable inertial sensors, vision systems, or ambient sensors [61]. A study by Omar et al. [62] reviews various advanced fall detection technologies for older adults. However, practical deployment in real homes and care facilities can remain a challenge. Wearables can be forgotten, removed, broken and malfunction after a fall; camera-based approaches raise privacy concerns and can fail under occlusion, lighting changes, or clutter [63]. As a result, these risks pose an urgent need for post-fall detection systems that can detect floor occupancy quickly and continuously and initiate an appropriate response.

3.2.1 Experiment Setup



Figure 3.22: Few example post-fall floor postures from the dataset. Subjects lie on the floor in multiple quasi-static poses (a) supine with arms extended, (b) side-prone, and (c) supine with knees bent. Furniture is intentionally placed near the subject to create multipath and partial occlusions.

For this study, we collected a dedicated dataset of quasi-static post-fall postures in a LTC facility using an off-the-shelf 60 GHz mmWave FMCW radar (Infineon XENSIV™ BGT60TR13C) [15] shown in Figure 2.1. The selected radar has one transmitter antenna (Tx) and three receiver antennas (Rx). With this alignment, it is possible to obtain both the azimuth and elevation angle of arrival from the reflected signals. The radar configuration used in all recordings is summarized in Table 3.1. With this setting, the radar can offer a maximum detection range of 9.6 meters, which is enough to cover the measuring areas. To reflect real indoor assisted-living conditions, data were collected in a furnished room with common clutter such as a sofa bed, armchairs, a coffee table, a standing lamp, shelves, etc. The furniture layout is intentionally kept close to the subject to create realistic multipath and partial occlusions. Two radar viewpoints are used: a TV-side view and a Window-side view. Each sensor is wall-mounted and tilted downward, approximately 2.5 m above the floor (Figure 3.2). To mimic post-fall conditions safely, subjects are asked to lie on the floor in several representative postures (e.g., facing up, side-lying, and knees-bent) as shown in Figure 3.22. For each viewpoint, the subject is placed at five floor locations (P1–P5), covering cases near the wall, near furniture edges, and open-floor regions. These positions are annotated as expected ground-truth regions and will be used for evaluation later in Section 4.3.5. For each subject, we recorded 2 minutes at 5 distinct positions and orientations relative to the sensors to vary the aspect angle, range, and background layout. Besides, for each measurement session, we placed this furniture randomly to create angular diversity to mimic different room layouts in the LTC. This enables our final datasets to be very diverse and reflect real-world setups by residents. Radar frames were acquired continuously at a 10 Hz rate, resulting in a total of approximately 43,000 frames.

For detection, each frame is first converted into a Range–Azimuth map using either the vendor DBF baseline or our proposed preprocessing pipeline. A two-dimensional CA-CFAR detector is then applied to identify spatially localized energy that corresponds to floor-level occupancy inside the Region of Interest (ROI). Following the background discussion in Section 2.3.1, the detector compares the power of each cell under test against an adaptive threshold derived from neighbouring training cells while excluding a guard region around the CUT. In this study, the CA-CFAR guard and training widths are fixed according to the settings established in the background chapter, and the main tuned parameter is the sensitivity scale factor k_{CFAR}, which controls the trade-off between missed detections and false alarms. We tune k_{CFAR} on Subject 1 only, which serves as the development subject for this post-fall study. For each candidate value of k_{CFAR}, we treat every frame as a binary decision: a positive frame indicates that the subject is present on the floor within the ROI, while a negative frame corresponds to an empty-room frame. We then compute the frame-level False Positive Rate (FPR), also referred to as FAR in Figure 3.23 –as the fraction of negative frames that produce at least one CFAR detection inside the ROI,

$$FPR = \frac{FP}{(FP + TN)}, \tag{3.10}$$

where False Positives (FP) counts empty-room frames. The selected operating point is the value of k_{CFAR} that satisfies a practical frame-level false-alarm cap while maximizing Macro-F1 on the tuning subset. This procedure yields a detector that is conservative enough for deployment, yet still sensitive to weak quasi-static returns from floor-level postures. It is important to emphasize that this frame-level FPR constraint is used only for detector tuning and fair comparison. In deployment, an alarm would not be triggered from a single positive frame. Instead, one would apply a temporal persistence rule, for example, ROI is detected in a large fraction of frames over a short window, such as 20 seconds. We choose an FPR cap of 0.1 ($\mathrm{FPR} \leq 0.1$), and we select k_{CFAR} by maximizing Macro-F1, which balances both detection (TPR) and missed frames (False Negative Rate (FNR)) on the tuning subset. This produces a working operating point that balances both missed detections caused by weak quasi-static signatures and false alarms caused by clutter leakage.

Figure 3.23 shows the sweep for both the vendor DBF baseline and our proposed preprocessing. It can be seen from the figure that increasing k reduces false alarms but can also suppress weak quasi-static returns, which affects both Macro-F1 and FPR. Under the same FPR cap, the selected operating points differ: the DBF baseline reaches its chosen point around $k_{\mathrm{CFAR}} = 1.4$, while our proposed preprocessing operates at a higher value $k_{\mathrm{CFAR}} = 2.4$. Importantly, when we evaluate empty-room data across all subjects and

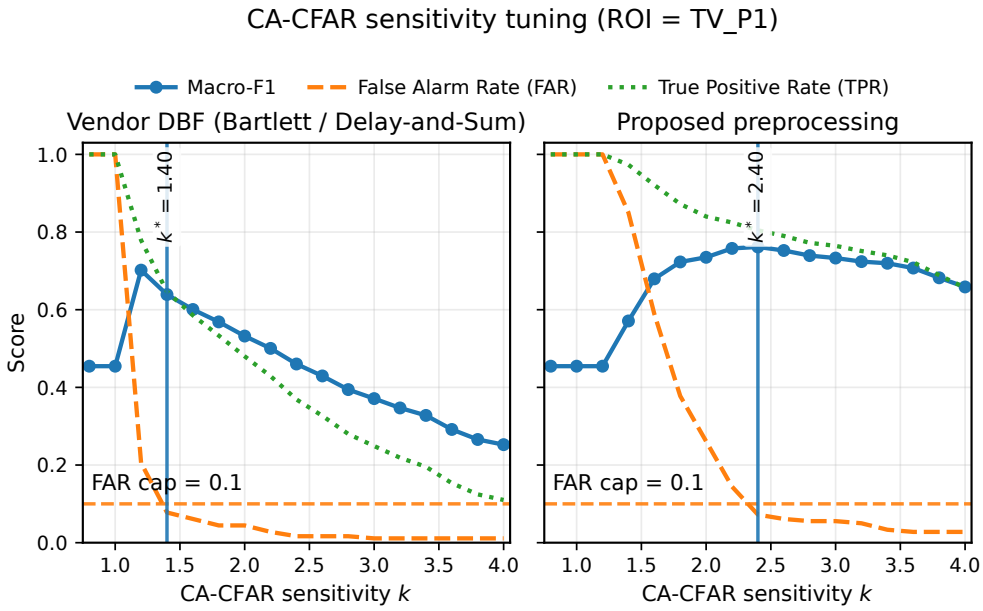

Figure 3.23: CA-CFAR sensitivity tuning on Subject 1's dataset. Macro-F1, frame-level false positive rate (FPR), and true positive rate (TPR) are shown as the CA-CFAR scale factor k_{CFAR} is swept. The selected operating points k satisfy the FPR cap (0.1) while maximizing Macro-F1 for each method.

views, the DBF pipeline achieves about 99% TNR, while the proposed pipeline achieves 100% TNR, corresponding to approximately 1% and 0% FPR, respectively.

Finally, this post-fall study also motivates the broader simulation work in Section 5.3 in Chapter 5. In the present chapter, the focus is deliberately on persistent floor occupancy in measured real data. In Section 5.3, this is extended to a digital-twin setting in which multiple fall types and fall trajectories can be simulated systematically. That later study complements the present one by moving from post-fall presence detection toward broader fall-related representation learning under controlled scene variation.

3.2.2 Results

After fixing k_{CFAR} for each pipeline, we keep it unchanged for all remaining subjects, viewpoints, and locations. Figures 3.24 and 3.25 qualitatively compare the detection results between the vendor DBF and the proposed detector on two representative frames of subject 1 lying on the floor at position 1. In both figures, the subject's approximated post-fall

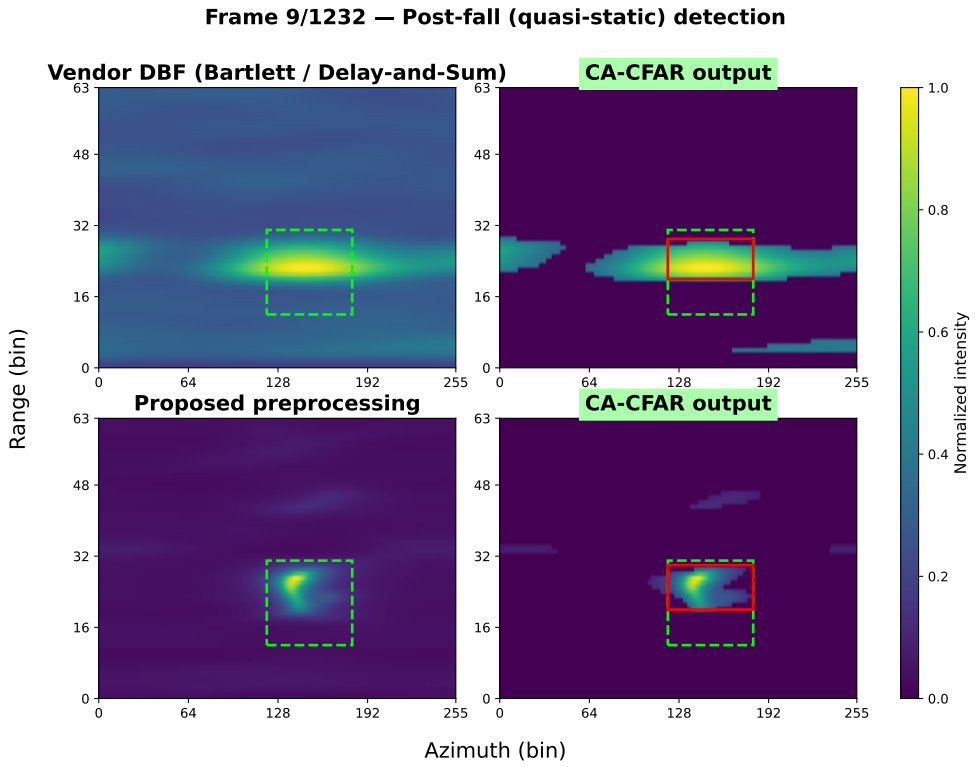

Figure 3.24: Qualitative comparison of DBF and our proposed detectors on frame 9 for subject 1 lying on the floor at position 1 (window view), which both of them produce detections within the ground-truth bounding box.

location is annotated by a green dashed bounding box in the RA map. For each location (for example, TV view and position 1), we determine a center point (r_0, θ_0) from the known floor placement during data collection and a one-time calibration of the radar coordinate system. We then use a fixed-size box around this center, $[r_0 \pm \Delta r,\ \theta_0 \pm \Delta\theta]$ with Δr and $\Delta\theta$ kept constant across frames and subjects for a given viewpoint/location to ensure consistent scoring. The chosen values are wide enough to capture the expected spread of the subject's return across nearby range and azimuth bins (due to posture variation, small placement shifts, and limited angular resolution), but still narrow enough to avoid counting unrelated clutter regions as true detections. We mark a frame as a hit if one remaining CA-CFAR detection (after thresholding and suppression) overlaps the annotated ground–truth bounding box. A green highlight indicates a successful detection, while a red highlight signifies a miss. Figure 3.24 shows a typical case where both methods successfully localize

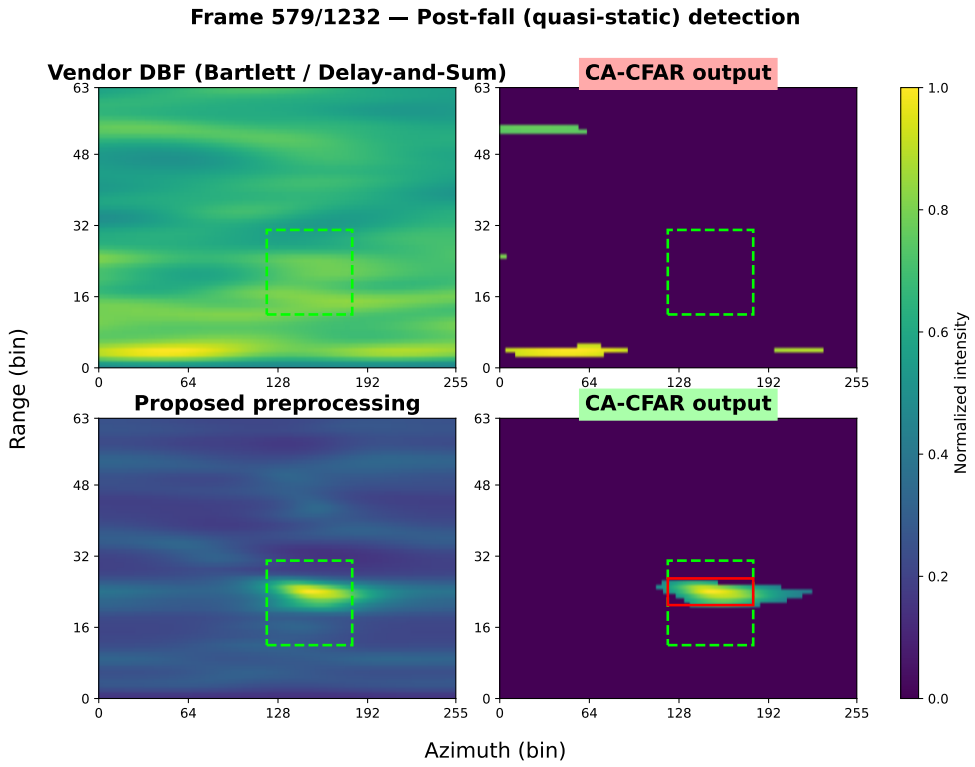

Figure 3.25: Qualitative comparison of DBF and our proposed detectors on frame 579 for subject 1 lying on the floor at position 1 (window view). Only our proposed method can detect within the ground-truth bounding box.

the subject within the ground-truth region, resulting in both results being counted, and the labels of the CA-CFAR results are highlighted in green. However, Figure 3.25 shows a more challenging frame where strong background clutter can affect DBF's result, CA-CFAR fails to localize the target and leaves with only clutter. Our proposed method, on the other hand, demonstrates superior stability under severe noise and succeeds in detecting the subject within the ground-truth bounds. We quantify performance using the frame-positive rate (fraction of frames in a trial that are counted as hits).

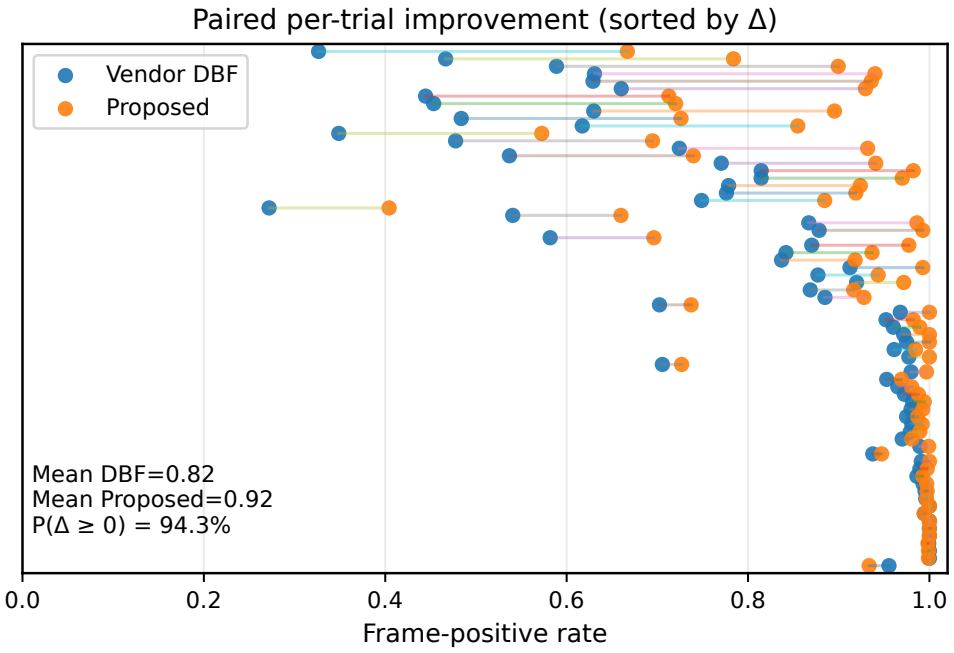

Figure 3.26: Paired per-trial comparison of frame-positive rate. Each point corresponds to one trial (subject, view, and location). Trials are sorted by improvement Δ.

Table 3.6: Participant-level frame-positive rate by viewpoint and location (higher is better). Best values per view and location for each subject are shown in bold.

View	Loc.	Subject 1		Subject 2		Subject 3		Subject 4		Subject 5		Subject 6		Subject 7	
		DBF	Prop.	DBF	Prop.	DBF	Prop.	DBF	Prop.	DBF	Prop.	DBF	Prop.	DBF	Prop.
TV	P1	0.35	**0.57**	0.78	**0.92**	0.72	**0.93**	0.59	**0.90**	0.75	**0.88**	0.88	**0.93**	0.54	**0.66**
	P2	0.98	**0.99**	0.98	**0.99**	0.70	**0.72**	0.98	**0.99**	0.58	**0.70**	0.97	**0.98**	0.97	**0.99**
	P3	0.97	**0.99**	0.99	**1.00**	0.77	**0.91**	1.00	1.00	1.00	1.00	0.99	**1.00**	0.96	**0.99**
	P4	0.27	**0.40**	0.77	**0.94**	0.99	**1.00**	0.98	**1.00**	0.97	**1.00**	0.97	**1.00**	0.99	0.99
	P5	0.63	**0.94**	0.97	**1.00**	0.33	**0.67**	0.88	**0.94**	0.47	**0.78**	0.87	**0.98**	0.97	**0.98**
	Mean	0.64	**0.78**	0.90	**0.97**	0.70	**0.85**	0.89	**0.97**	0.75	**0.87**	0.94	**0.98**	0.89	**0.92**
Window	P1	0.63	**0.90**	0.91	**0.99**	0.88	**0.99**	0.81	**0.97**	0.81	**0.98**	0.92	**0.97**	0.63	**0.94**
	P2	0.48	**0.73**	0.87	**0.99**	0.48	**0.69**	0.62	**0.85**	0.44	**0.71**	0.84	**0.92**	0.54	**0.74**
	P3	0.99	0.99	1.00	1.00	0.98	**0.99**	0.87	**0.92**	0.96	**0.98**	0.95	**0.97**	0.95	**0.98**
	P4	1.00	1.00	1.00	1.00	0.98	**1.00**	**0.96**	0.93	1.00	1.00	0.99	0.99	0.94	**0.95**
	P5	0.70	**0.74**	1.00	1.00	0.88	**0.94**	1.00	1.00	1.00	1.00	0.97	**0.98**	0.99	**1.00**
	Mean	0.76	**0.87**	0.96	**1.00**	0.84	**0.92**	0.85	**0.93**	0.84	**0.93**	0.93	**0.97**	0.81	**0.92**
Overall		0.70	**0.83**	0.93	**0.98**	0.77	**0.88**	0.87	**0.95**	0.80	**0.90**	0.94	**0.97**	0.85	**0.92**

Table 3.6 breaks down the accuracy detection by viewpoint (TV-side vs Window-side) and floor locations (P1-P5) for seven subjects. Many trials are already near 100% accuracy for both methods, but the largest gains appear in harder cases where DBF struggles. For example, Subject 1's accuracy at TV–P1 increases from 0.35 to 0.57, and Subject 1's accuracy at TV–P5 increases from 0.63 to 0.94. The worst DBF case (0.27 at TV–P4 for subject 1) also improves to 0.40. These improvements suggest that the proposed method is especially helpful when the post-fall signature is weak, partially blocked, or mixed with static clutter. We note that even with the improvement, TV-P4 for Subject 1 remains a challenging condition, where many frames are missed due to strong clutter and geometry (e.g., reflections from nearby furniture or partial blockage). As mentioned above in 2.3.1, in a real-life safety-critical deployment workflow, post-fall detection is not a one-frame decision, but temporal integration can turn a moderate per-frame hit rate into a reliable alarm over a short window. For example, if at least 80% of frames are detected over 20 seconds, an alert is triggered.

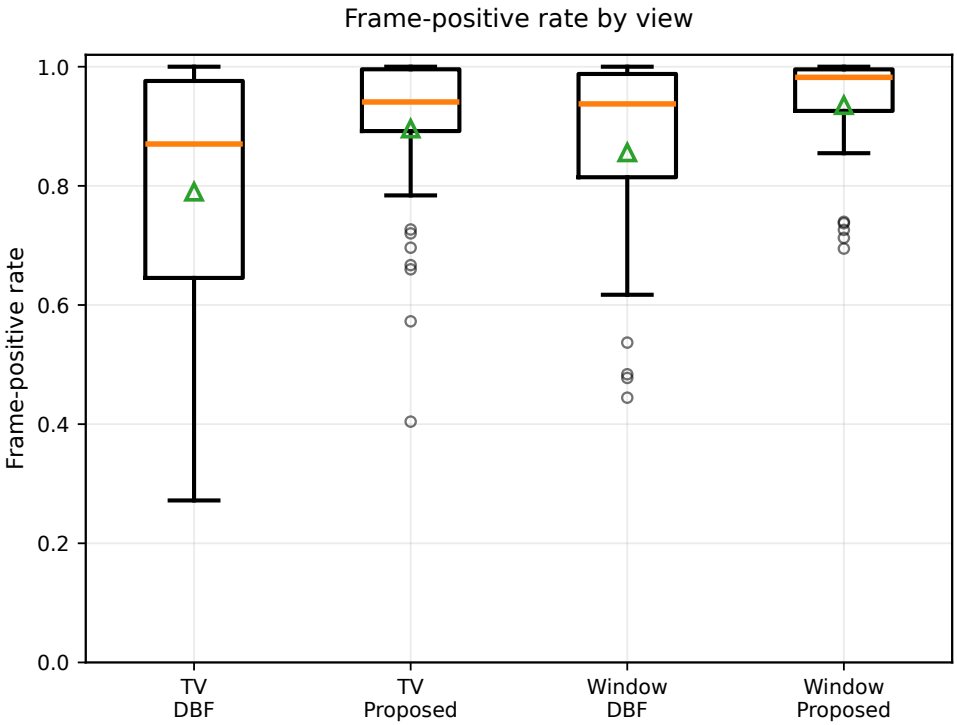

Figure 3.27: Frame-positive rate by viewpoint. Boxplots compare the vendor DBF and proposed method for TV-side and Window-side views.

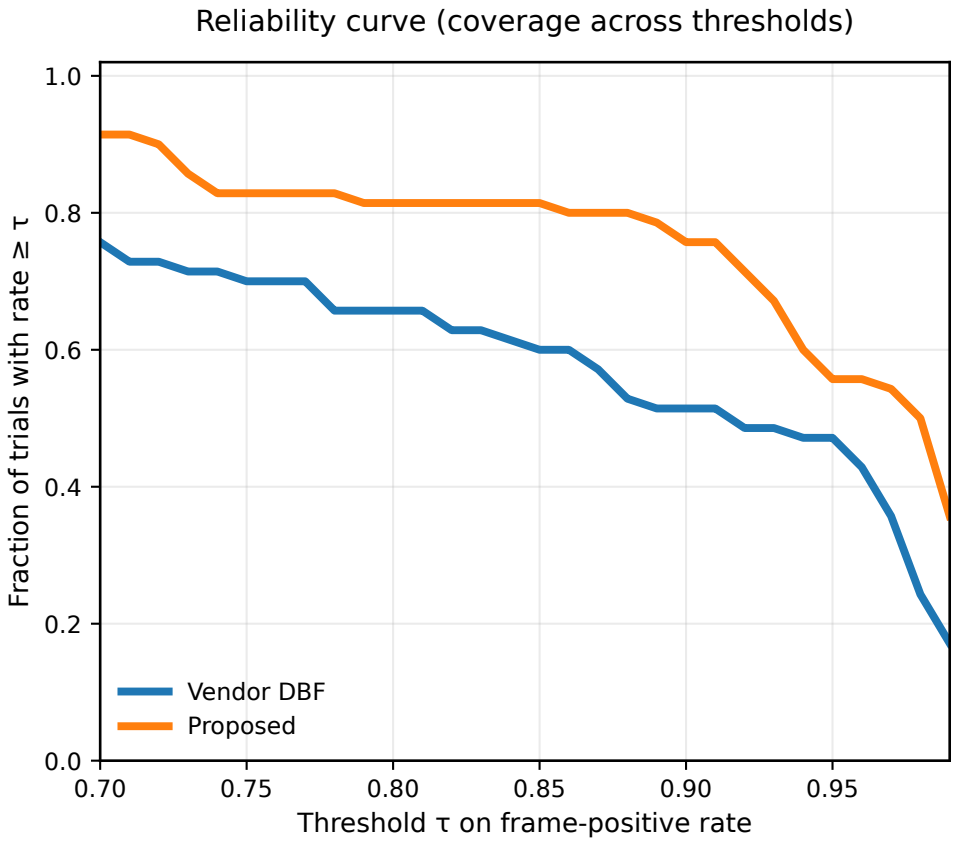

Figure 3.28: Reliability curve (coverage across thresholds). The y-axis shows the fraction of trials with frame-positive rate $\geq \tau$.

Figure 3.26 plots paired trial-level frame-positive rates for DBF and the proposed method, sorted by the improvement Δ. Across all paired trials, the proposed method increases the mean frame-positive rate from 0.82 (DBF) to 0.92 (Proposed), and improves or matches DBF in 94.3% of trials. This indicates that the gain is not limited to a small subset of cases, but is consistent across subjects, viewpoints, and post-fall locations. To summarize the distribution shift, Figure 3.27 reports boxplots of frame-positive rates grouped by viewpoint and method. For both TV-side and Window-side views, the proposed method shifts the distribution upward compared to the DBF baseline. This outlier improvement matters in a clinical monitoring context. In a clinical monitoring context, reducing these low-performing trials is crucial since those are the situations most likely to produce missed post-fall detection. And finally, to better quantify the robustness at different accuracy re-

quirements, we report the reliability curve in Figure 3.28. We measure the fraction of trials whose frame-positive rate is at least τ. This is crucial in a clinical setting since the system is expected to remain reliable and sustained rather than occasional correct detection. As shown in Figure 3.28, the proposed method outperforms the DBF baseline across various thresholds, meaning it maintains higher coverage even as the success criterion becomes stricter. This suggests our method not only improves the mean score, but also in the worst-case reliability, which is crucial for reducing missed post-fall detections in difficult layouts and subject positions.

3.3 Conclusion

This chapter addressed two closely related radar sensing problems in long-term care settings: semi-static occupancy detection and post-fall floor-occupancy detection. In the first part of the chapter, we proposed RASSO, a Doppler-domain warping and resampling method designed to improve sensitivity to weak near-zero Doppler signatures. By reallocating sampling density around the quasi-static region while preserving the complex structure required for downstream beamforming, RASSO produced more compact and better concentrated range–azimuth feature maps than the conventional pipeline. Across classical CFAR-based evaluation, matched-FAR benchmarking, and data-driven classification with both SimpleCNN and CNN–LSTM models, the RASSO-enhanced representations consistently improved the separation between empty-room and low-motion human presence.

The second part of the chapter focused on post-fall floor-occupancy detection in a furnished LTC-like environment. Here, the task was not to detect the fall event itself, but to determine whether a person remained on the floor under realistic clutter, furniture reflections, and viewpoint variation. The study showed that floor-level occupancy can be detected reliably using low-cost radar when the processing chain is tuned to the characteristics of weak, quasi-static returns. In particular, the proposed method yielded more reliable detections than the vendor DBF baseline under the same frame-level false-alarm constraint and across multiple subject positions and viewpoints.

Taken together, the two studies in this chapter support the same overall conclusion: low-cost 60 GHz FMCW radar can provide meaningful room-level awareness in LTC settings even when the target motion is weak or absent, provided that the signal representation and downstream detection stages are designed specifically for quasi-static human states. These results motivate the imaging-based processing methods of Chapter 4 and the simulation-based extensions of Chapter 5.

Chapter 4

Imaging-Based Radar Representation Processing for Occupancy Detection at the Edge

4.1 Chapter Motivation and Scope

Real-world deployment of radar-based human sensing in LTC facilities must operate under practical constraints that are often underemphasized under control laboratory studies. These involve cluttered furniture layouts, wall and floor reflections, multipath, non-ideal subject poses, and strict requirements on privacy, latency, energy efficiency, and cost. Under these conditions, the challenges lie not only in detecting and classifying human activities, but also in ensuring that the radar representation itself is sufficiently reliable for downstream processing. In this chapter, we investigate whether image-based processing can be used to improve the reliability, alignment and computational efficiency of radar-based occupancy sensing using low-cost FMCW radar.

A central premise of this chapter is that several commonly used radar representations—such as micro-Doppler (mD) spectrograms and Range-Angle feature maps—can be treated as structured images. This enables the use of image-processing methods as principled mechanisms for evaluating representation quality, suppressing clutter, enforcing temporal consistency, and designing lightweight detection pipelines. We thus study how image-based operations can provide interpretable improvements that are suitable for embedded deployment in real LTC rooms. The chapter is organized around three research questions. First, for dynamic activities represented by mD spectrograms, how should

denoising and preprocessing methods be evaluated, especially under severe noise? Second, for quasi-static activities represented by Range-Angle feature maps, how can map fidelity be quantified when no clean reference is available? Third, can these insights be translated into a lightweight radar detection method that is accurate, fast, and practical for edge devices? More specifically, this chapter makes three contributions. First, it benchmarks recent radar-spectrogram preprocessing and denoising methods and shows that conventional error-based metrics do not always reflect perceptual fidelity in low-SNR conditions. Second, it introduces no-reference quality scoring and temporal tracking methods for Range-Angle feature maps in quasi-static sensing scenarios. Third, it presents a lightweight image-based detector that operates on Range-Angle maps and achieves strong detection accuracy while meeting real-time embedded constraints. Collectively, these contributions show that image-based radar processing can serve as a practical path toward scalable, privacy-preserving sensing in LTC environments.

4.2 Radar Image Preprocessing Evaluation

Radar spectrograms containing human activities are often degraded by noise and clutter, which can obscure the micro-Doppler structures needed for reliable interpretation. In this section, we evaluate several preprocessing and denoising methods to determine how well they preserve activity-relevant structure in noisy spectrograms. Our goal is not only to reproduce prior pipelines, but also to examine whether commonly used quantitative metrics adequately reflect representation quality under low-SNR conditions. To this end, we reimplemented three recent radar spectrogram preprocessing methods and evaluated their outputs using both conventional error-based metrics and perceptual image-quality measures. This analysis provides the first step of the chapter's broader argument: before proposing new image-based radar algorithms, it is necessary to understand how radar representations should be assessed and which metrics remain informative when the signal is heavily corrupted.

4.2.1 Adaptive Preprocessing

An Adaptive Resolution Preprocessing (APr) technique has been suggested by Park et al. [64] to improve mD spectrogram-based Human Activity Recognition (HAR) using FMCW radar signals. Conventional spectrograms employ a fixed time-frequency resolution, which frequently falls short of capturing the characteristics of the entire range of motion, particularly for low-frequency or subtle movements. The authors address this by

introducing a resolution-adaptive spectrogram, which uses the energy distribution of the mD signatures to non-linearly adjust frequency resolution. In particular, the technique locates each spectrogram's active frequency range and uses logarithmic mean-square scaling to improve resolution in lower-frequency bands, which are usually where fine motion details show up. Compared to traditional representations, this adaptive method generates more discriminative spectrograms, which enhances the performance of the deep learning model.

4.2.2 Adaptive-Thresholding

An Adaptive Thresholding Algorithm (ATh) proposed by Li et al. [65] offers a low-computation, high-accuracy solution for radar-based human activity recognition, making it well-suited for multi-domain mD signatures and resource-constrained systems. The method consists of three stages: signal preprocessing, adaptive thresholding, and classification. In preprocessing, a Hamming-windowed FFT generates a range-time map from time-domain radar pulses, followed by a 4th order Butterworth filter [66] to suppress noise. A Short-Time Fourier Transform (STFT), using a 0.2 second Hamming window with 95% overlap, produces the mD spectrogram. Adaptive thresholding converts the spectrogram to grayscale and iteratively updates a threshold based on average pixel intensity until convergence (typically at $V = 0.1$). The resulting binary mask is used to generate masked spectrograms and phase-related images. The final classification is performed using a Support Vector Machine (SVM) algorithm.

4.2.3 Entropy-Based Denoising

The third reimplemented algorithm, proposed by Nguyen et al. [67], applies a two-step Entropy-Based Denoising (EBD) approach for spectrogram enhancement and classification. The first step identifies the optimal range-bin interval containing the most motion-relevant signal energy. This is achieved by computing the FFT of each range-bin across the slow-time axis, identifying the most frequently occurring peak location, and evaluating surrounding intervals via STFT. The interval with the lowest average Shannon entropy, H_{avg}, is selected as optimal.

$$r_{opt} = \arg \min_{1 \leq q \leq Q} H_{avg}(range_{r_q}) \tag{4.1}$$

In the second step, adaptive spectrogram thresholding is applied within the selected interval. A Gaussian-windowed STFT is computed frame-by-frame, and a dynamic cut-

threshold is generated based on local average energy. This masking function reduces background noise while preserving key mD signatures, ultimately improving the clarity of the spectrogram for downstream classification tasks.

4.2.4 Experiment Setup

After implementing the algorithms discussed above, we evaluated their performance using the dataset provided in [67]. The raw signal dataset comprises 19,800 samples, generated by combining 11 distinct activities, 5 aspect angles, and 60 iterations. The first step in our processing pipeline involves generating spectrograms for each radar signal file. Following this, WGN is added at 5 different SNR levels: -10 dB, -5 dB, 0 dB, and 10 dB. The resulting spectrograms are illustrated in Figure 4.1. These noisy spectrograms are then processed using three reimplemented denoising methods, as well as our combined APr and ATh method (APr+ATh). The Structural Similarity Index Measure (SSIM) [68],

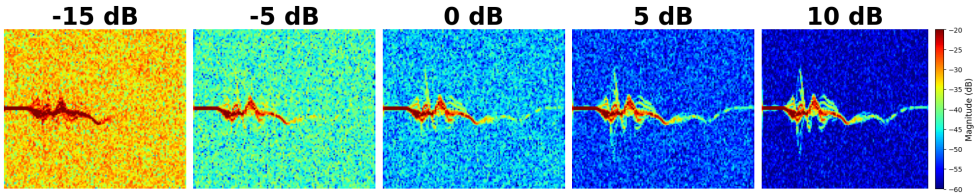

Figure 4.1: Spectrogram after adding 5 different White Gaussian noise levels.

a perceptual metric that evaluates image quality based on structural fidelity, is used to evaluate the denoised spectrogram outputs, which have been done quite rarely before [69]. The similarity between image structure and contrast is the main focus of SSIM. We used several common quantitative evaluation metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), and the Pearson Correlation Coefficient (PCC).

4.2.5 Results

The above findings suggest that representation quality in radar should not be evaluated by pixel-wise agreement, especially in noisy conditions where visually meaningful structure may be more important than absolute error. This motivates the next part of the chapter, where we consider quasi-static sensing and develop no-reference methods for assessing the reliability of range-angle feature maps.

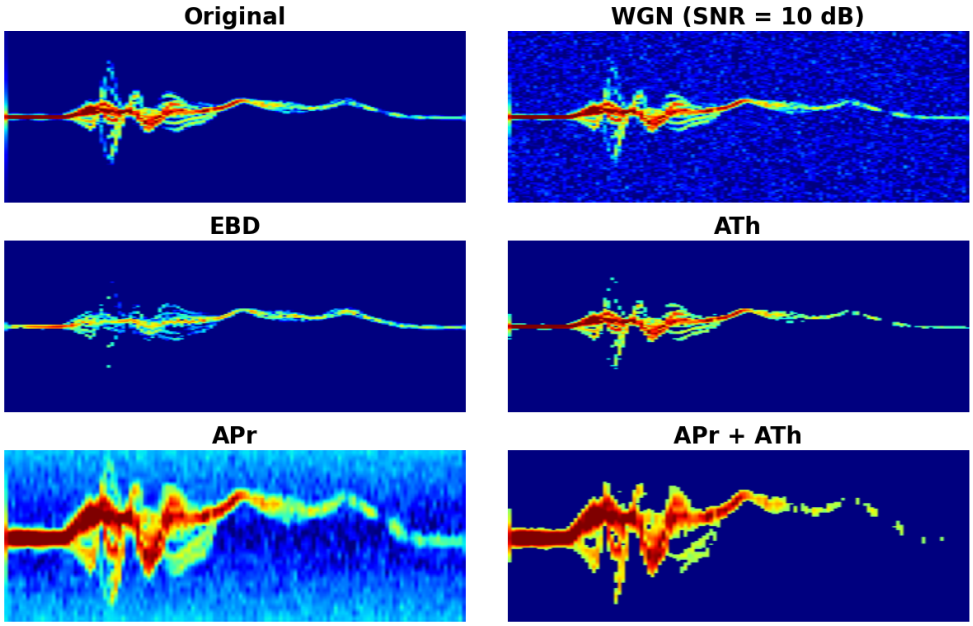

Figure 4.2: Spectrogram results for Walking (W): original spectrogram (top-left), noisy spectrogram (top-right) and 4 methods under WGN (SNR = 10 dB).

Figure 4.2 shows the result of 4 methods along with the original noise-free and noisy spectrograms. Its corresponding evaluation metrics are shown in Table 4.1. It can be inferred from the table that the ATh algorithm removes noise selectively and preserves most of the detail in case SNR = 5 dB. EBD algorithm, however, preserves large-scale intensity distribution with high correlation but degrades to local structures. The APr algorithm effectively adjusted the frequency resolution, enabling mD signatures to span a broader frequency range. This ensures optimal representation by dynamically expanding the active frequency, though it also has the unintended effect of scaling noise within the calculated range. Applying ATh after APr removes these noises for a cleaner representation, but can also remove fine main signal details in this case.

Under severe noisy (–5 dB) conditions (Figure 4.3, Table 4.2), EBD shows a better ability to denoise, although there are some speckles left in the background, which may explain lower metrics compared to its corresponding values in Table 4.1. ATh is quite prone to very high noise levels, whose SNRs are negative. However, its pixel error (MSE, RMSE, PSNR) metrics are lower than those of EBD. SSIM, on the other hand, shows a more reasonable alignment in both cases. Thus, SSIM can be used as a cost function when training deep learning models for fidelity-based applications such as Generative Adversarial

Table 4.1: Evaluation metrics result for walking (W) under White Gaussian Noise (SNR = 10 dB).

METRICS	NS	EBD	ATh	APr	APr + ATh
MSE	5053.2	12953.8	**4785.7**	6254.1	5043.3
MAE	63.82	107.48	**62.15**	73.05	65.58
RMSE	71.08	113.9	**69.17**	79.08	71.01
PSNR (dB)	-19.69	-23.78	**-19.46**	-20.62	-19.68
Pearson Corr	0.572	**0.6168**	0.6006	0.4462	0.5561
SSIM	0.365	0.3639	**0.4109**	0.2881	0.3227

Bold indicates the best (optimal) value in each row

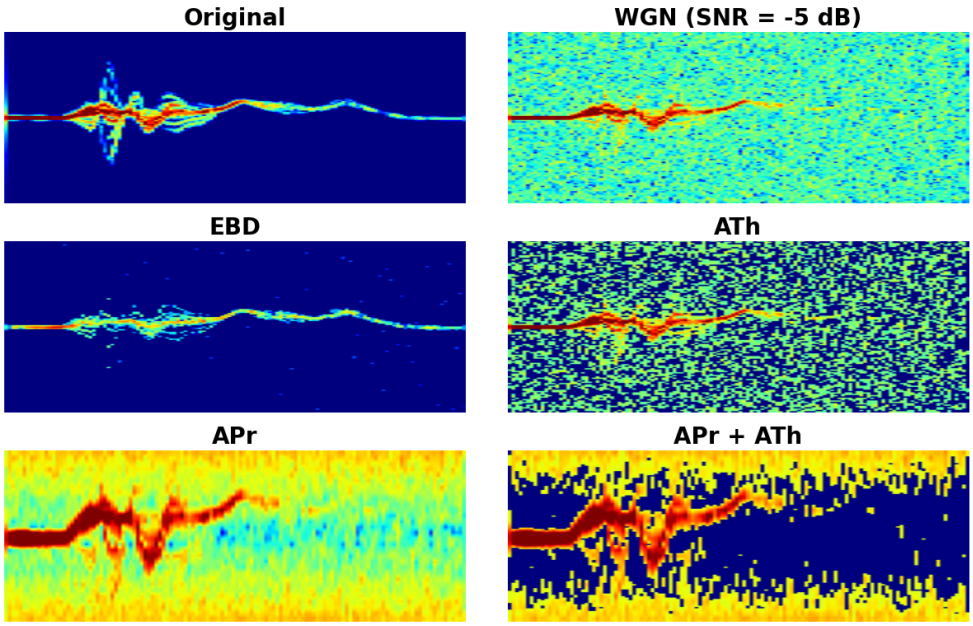

Figure 4.3: Spectrogram results for Walking (W): original spectrogram (top-left), noisy spectrogram (top-right) and 4 methods under WGN (SNR = -5 dB).

Network (GAN)-based [70] reconstruction or synthetic spectrogram generators.

According to our study, depending only on error-based metrics can be deceptive, especially when there is a low SNR, as perceptual fidelity rather than pixel-level similarity is needed to preserve subtle mD signatures, which are essential for precise human activity recognition. Moreover, we find that the SSIM metric does not always match the results of the APr preprocessing method, indicating that structural similarity might not adequately

Table 4.2: Evaluation metrics result for walking (W) under White Gaussian Noise (SNR = -5 dB).

METRICS	NS	EBD	ATh	APr	APr + ATh
MSE	7184.1	13501.9	**6152.6**	8365.6	6935.6
MAE	77.81	110.18	**71.01**	84.86	75.32
RMSE	84.76	116.19	**78.43**	91.46	83.28
PSNR (dB)	-21.22	-23.96	**-20.55**	-21.8864	-21.07
Pearson Corr	0.329	**0.533**	0.225	0.073	-0.058
SSIM	0.261	**0.305**	0.175	0.227	0.177

Bold indicates the best (optimal) value in each row

convey the perceptual quality enhancements brought about by adaptive range-bin selection. This motivates a need for better-tailored metrics to specifically evaluate the quality of spectrograms and also align better with the model's perception when training a deep-learning-based denoiser. A good spectrogram quality metric should take into consideration important characteristics like background noise suppression and the clarity and continuity of ridge patterns that represent mD signatures.

The above findings have two implications for the remainder of this chapter. First, they show that the quality of a radar representation cannot always be judged adequately through pixel-wise agreement alone; in low-SNR conditions, structural fidelity and the preservation of semantically meaningful patterns are often more important. Second, they motivate extending this idea beyond dynamic mD spectrograms toward spatial radar representations used for quasi-static activities. In such settings, clean references are rarely available, and the notion of representation quality must instead be inferred from intrinsic spatial characteristics such as compactness, dominance, and the absence of clutter-induced ghost responses. This observation motivates the proposed no-reference quality scoring and temporal tracking methods introduced in the next sections.

4.3 Proposed Imaging-Based Algorithms

The evaluation in Section 4.2 suggests that useful radar representations should not be assessed solely by pixel-level similarity, but by how well they preserve interpretable and task-relevant structure. We now extend this principle to quasi-static human sensing, where the challenge differs fundamentally from dynamic micro-Doppler analysis. Static and slowly varying activities often produce weak Doppler signatures; thus, spectrogram representation might not contain many useful features. In contrast, RA maps retain spatial information

that can localize dominant reflectors even when motion is limited. However, these maps are themselves highly variable in quality: some frames contain a clean, compact subject target, whereas others are corrupted by background ridges, multiple competing peaks, or diffuse clutter, as can be seen in Figure 3.7 in Chapter 3.

4.3.1 Experiment Setup

We used a subset of the dataset mentioned in Section 3.1.2, similar to Section 3.1. This sub-dataset contains the first 4 subjects. For each activity (standing, sitting on the sofa, sitting on the floor, lying on the sofa, lying on the floor), four subjects were recorded for two minutes each at multiple aspect angles and ranges at a 10 Hz rate. All experiments for designing the three algorithms below in this Section 4.3 use Subject 1's data as the held-out test set to tune their hyperparameters, while Subjects 2, 3, and 4's data are used as the generalization test sets to be evaluated. However, the proposed quality scoring algorithm in Subsection 4.3.2 and the temporal tracking algorithm in Subsection 4.3.3 use test-sets for qualitative evaluation only. The final lightweight algorithm for quasi-static detection in Subsection 4.3.4 evaluates test-sets qualitatively and quantitatively.

4.3.2 Proposed Quality Scoring Algorithm

In quasi-static sensing, the usefulness of a RA feature map depends strongly on its spatial reliability. A high-quality RA map is one in which the subject response appears as a dominant and spatially coherent "blob-shape" target, with limited competing peaks and minimal background clutter. For brevity, these spectral peaks will be denoted as **'blobs'** throughout this study. In contrast, a low-quality map contains multiple disconnected high-intensity regions, strong ghost responses, broad clutter ridges, or ambiguous energy spread that makes the subject difficult to distinguish from the environment. Different from reference-based assessment as in Section 4.2, to quantify this property without requiring a clean reference map, we propose a no-reference RA-map quality score that produces a frame-level score $S \in [0, 1]$ as Algorithm 1. The score combines three intuitive characteristics: (i) energy concentration in high-intensity blobs, (ii) purity of each blob via sub-peaks, and (iii) global peak prominence. High-intensity blob is often, but not necessarily, true scatter, while multiple scatters and subpeaks will result in a penalty. These will capture cases of bad-quality feature maps that contain different blobs or a true blob that is a burden in the background. In this formulation, the blob-based term captures spatial concentration and clutter leakage, while the peak-prominence term captures whether a sufficiently dominant reflector exists relative to the map background. Their combination yields a compact

scalar proxy for frame reliability. In this way, the score is not intended to measure generic visual quality, but rather to estimate whether the RA representation is sufficiently reliable for downstream quasi-static occupancy sensing. A high score, therefore, indicates that the map contains a compact and dominant subject response, whereas a low score indicates that the response is weak, cluttered, or spatially ambiguous.

Algorithm 1 Calculate frame score based on blob + Peak Scores

blobs $\leftarrow$ BLOBSFROMFRAME(X, *pct*) {find connected blobs above percentile} {**detect sub-peak inside blobs**}
subpeakCount $\leftarrow 0$
maxVal $\leftarrow \max(subArray)$
absThresh $\leftarrow \gamma \times maxVal$
for all blob L in *validblobs* **do**
    coords $\leftarrow$ PEAKLOCALMAX($subArray, d_{\min}, absThresh$)
    subpeakCount $\leftarrow subpeakCount + \left|coords\right|$
end for
blobsPenalty $\leftarrow \alpha \times (subpeakCount - 1)$
$leftoverRatio \leftarrow \dfrac{E_{\text{total}} - E_{\text{main}}}{E_{\text{main}}}$
$blobsScore \leftarrow \dfrac{1}{1 + leftoverRatio \ + \ blobsPenalty}$
$blobsScore \leftarrow \max\big(0,\ \min(blobsScore, 1)\big)$
$p \leftarrow \max(X)$ {global peak}
$\mu \leftarrow mean(X)$
$\sigma \leftarrow std(X) + 10^{-9}$
$peakness \leftarrow \max\{0,\ (p - \mu)/\sigma\}$
$peakScore \leftarrow 1 - \exp(-\phi \times peakness)$
$peakScore \leftarrow \max\big(0,\ \min(peakScore, 1)\big)$
$w1 \leftarrow 0.2, \quad w2 \leftarrow 0.8$ {weights}
$S \leftarrow (blobsScore)^{w1} + (peakScore)^{w2}$
$S \leftarrow \max\big(0,\ \min(S, 1)\big)$
return S

As we mentioned above in Subsection 4.3.1, the score hyperparameters were selected qualitatively on Subject 1's held-out test set and then inspected on the remaining subjects to ensure the score remained interpretable across participants. In our implementation, we used a percentile threshold of $pct = 90$, a subpeak penalty $\alpha = 3$ for the single-

peak case (and $\alpha = 0.1$ when multiple peaks are present), and fusion weights $w_1 = 0.25$ and $w_2 = 0.75$. Unlike the spectrogram evaluation in Section 4.3, where reference-based metrics can be computed by comparing a processed output to a known input representation, the present problem does not admit a clean reference RA map for direct comparison in real scenes. We therefore interpret the score S as a *no-reference* reliability indicator: it summarizes whether the current RA map exhibits the spatial characteristics expected of a confident subject response (dominant, compact, and minimally cluttered), rather than measuring fidelity to an external ground truth.

To make the score interpretable in downstream processing, we partition S into three qualitative regimes based on visual inspection of RA maps and typical clutter patterns observed in the dataset. Scores below 0.8 generally correspond to frames where the subject response is weak or spatially ambiguous, often co-occurring with clutter ridges, ghost responses, or multiple competing blobs. Scores in the range $[0.8, 0.9)$ typically correspond to moderately reliable frames, where the dominant blob becomes visible but residual clutter or secondary responses may still be present. Scores above 0.9 correspond to frames in which the dominant subject blob is clearly localized and well separated from the background. Representative examples of these regimes are shown in Figure 4.4. In addition, we performed a qualitative temporal consistency check by averaging scores over short windows (e.g., 30 s) during segments that visually exhibited stable and confident subject responses. We observed that segments with consistently high-quality RA maps tended to yield average scores in the higher regime (typically ≥ 0.8), whereas visually cluttered or ambiguous segments produced lower averages. This analysis does not constitute a formal quantitative validation, but it supports the intended use of S as a practical reliability cue for subsequent temporal association and lightweight detection in Sections 4.3.3–4.3.4.

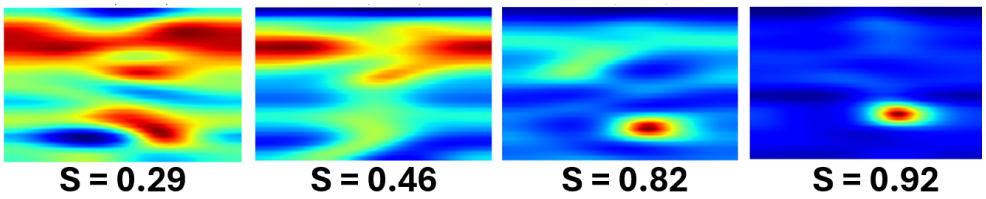

Figure 4.4: Scores calculated from our proposed algorithm corresponding to the feature map.

4.3.3 Temporal Tracking Algorithm

While the quality score provides a frame-level estimate of RA-map reliability, practical sensing systems must also remain stable over time. In quasi-static sensing, the subject reflector is expected to vary smoothly across adjacent frames, whereas clutter-induced responses may appear intermittently or at inconsistent locations. For this reason, the score is used not only as a descriptive measure but also as a confidence signal that governs temporal association. At each frame t, we first extract candidate components $\mathcal{C}_t$ from the RA map X_t using the same percentile-gated connected-component procedure described in Section 4.3 (proposal mask, morphological cleanup, and connected components). We then compute a *frame-level* quality score $S_t \in [0, 1]$ using Algorithm 1 on the full RA map. If $S_t \geq \tau_S$, the frame is considered reliable and the dominant candidate is accepted directly as the tracked target. Otherwise, we avoid forced selection and instead perform a nearest-centroid association: we compare the centroid of the previously accepted target to the centroids of the current candidates, and select the nearest candidate within a distance threshold d. If no candidate satisfies the distance gate, we hold the last valid target for a short temporal window of length W to smooth transient dropouts; if the dropout persists beyond W, no update is issued. In our implementation, we use $\tau_S = 0.8$, $d = 30$ pixels, and

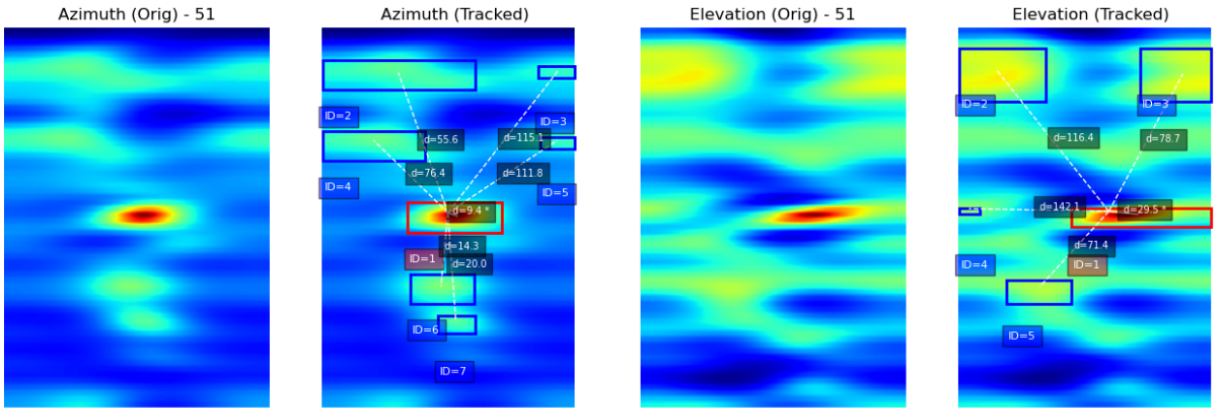

Figure 4.5: Range-Azimuth and Range-Elevation feature maps of frame 51 (class 2) and their tracking results (columns 2 and 4).

$W = 5$ frames, corresponding to 0.5 s at the 10 Hz radar frame rate. Figure 4.5 illustrates a representative example, where the selected component is shown in red and competing candidates are shown in blue. Since no dense ground truth trajectory is available for every frame in real scenes, we evaluate this tracker qualitatively by verifying that the selected component remains spatially consistent with the visually dominant subject response and

that spurious switches to clutter are reduced during low-quality intervals. The purpose of this tracking step is therefore to provide temporal stability and reliable region proposals for the subsequent lightweight detector, rather than to claim optimal tracking performance under a fully supervised benchmark.

4.3.4 Lightweight Algorithm for Quasi-Static Detection

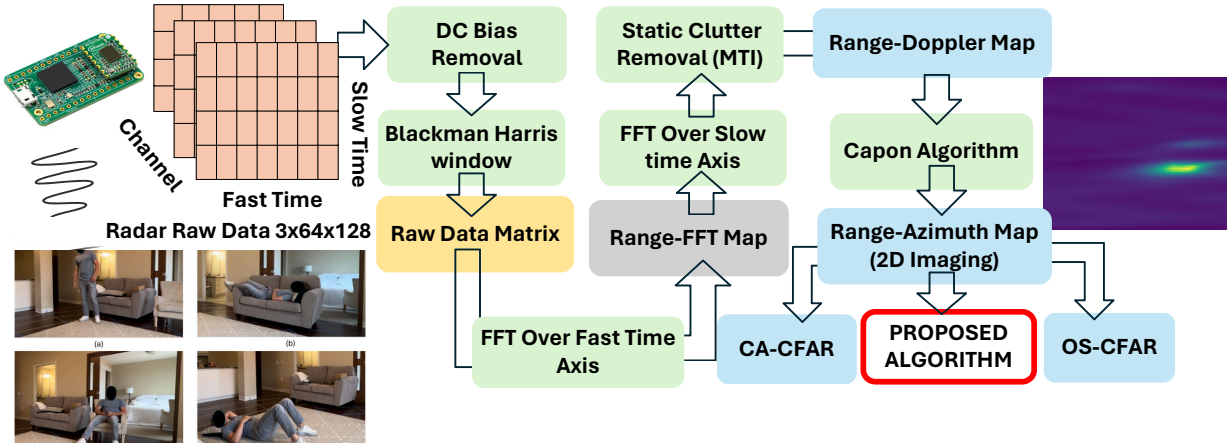

Figure 4.6: End-to-end radar's imaging processing pipeline. Raw complex IQ frames from the 60 GHz FMCW radar are converted to range–Doppler maps via range and Doppler FFTs with DC removal, windowing, and MTI clutter suppression. Capon beamforming over two receive channels produces a normalized range–azimuth (RA) intensity map, which is then processed by three detectors: 2-D CA–CFAR, 2-D OS–CFAR, and the proposed percentile-gated lump detector.

The quality-scoring and tracking results indicate that, for quasi-static activities, the most informative cue in the RA map is often the presence of a dominant connected reflector. This motivates a lightweight detector that operates directly on the normalized RA image. The proposed method first forms a sparse proposal mask by retaining pixels whose intensity exceeds a global percentile threshold, then enforces spatial coherence through morphological closing and small-component removal. We apply a small 2×2 structuring element for binary closing to connect nearby bright pixels along range and azimuth, and we remove any blobs whose area is below $A_{\min}$. A two–pass connected–components algorithm labels the remaining blobs, and we keep only the largest component as the final detection blob. A key design requirement is that the detector should not force a positive decision in every frame. In practical LTC environments, empty-room scenes, clutter

ridges, or transient interference may still produce residual bright regions after percentile gating. Therefore, the percentile threshold is used only as a proposal stage. The final decision is made by verifying whether the dominant blob is sufficiently reliable; otherwise, the frame is rejected, and no detection is returned. Algorithm 2 summarizes this procedure. We compared our method against CA-CFAR and OS-CFAR as a baseline. For fairness, all three detectors operate on the same normalized RA image produced by the upstream radar imaging chain shown in Figure 4.6. That chain follows the same FMCW preprocessing principles introduced earlier in Chapter 2 and the same Capon-based spatial imaging used in Chapter 3; however, in this chapter, the comparison is intentionally restricted to the *detector stage* in image space. For each frame, a positive detection is counted when the detector returns a dominant blob that lies within the annotated subject region, whereas detections outside the annotated region are treated as false positives. The subject-wise results reported later in Section 4.3.5 summarize frame-level detection accuracy across the evaluation set, while the runtime analysis on the Raspberry Pi 4B measures the average per-frame execution cost of each detector. In this way, the reported results isolate the practical trade-off between detection reliability and computational efficiency for edge deployment, rather than re-evaluating the upstream signal representation itself. This separation is crucial since the purpose of the proposed method is not to replace the radar imaging pipeline developed earlier, but to provide a lightweight alternative to conventional CFAR-based detectors once an image-like radar map has already been formed.

Here, textscLargestComponent($\mathcal{C}$) denotes the connected component with the largest area, and ConfidenceOK($c^\star, X, \mathcal{C}$) denotes a lightweight component-level verification step that rejects blobs with weak contrast, poor concentration, or insufficient dominance over competing components. Algorithm 2 remains lightweight, since it consists only of percentile thresholding, simple morphology, connected-components labelling, and a small component-level verification step. In contrast, 2-D CA-CFAR and OS-CFAR require sliding-window processing around each cell, leading to higher computational cost and runtime. The comparison is therefore meaningful at the task level: all methods operate on the same RA maps and produce frame-level localization decisions, but they rely on different decision principles. Figure 4.7 illustrates that different percentile intensity p affects the detection mask on a representative RA frame for the Lying on sofa activity of Subject 1. At the 90th and 95th percentiles, the mask still includes a wide horizontal band along the dominant clutter ridge, causing the detected blob to spread across the entire sofa region. In contrast, the 99th percentile isolates a smaller, well-localized blob that aligns with the annotated human activity detection.

4.3.5 Results

Figures 4.8 and 4.9 compare the detection results on representative frames for Subject 1 lying on the sofa. In these visualizations, the subject's actual position is annotated by a green dashed bounding box. We further encode the decision outcome in the method head-

Algorithm 2 Confidence-Gated Range–Angle blob Detector

Require: Normalized RA map $X \in [0,1]^{H \times W}$, percentile p, minimum blob area $A_{\min}$
Ensure: Binary mask M and bounding box B (or $\emptyset$)

$\tau \leftarrow \textsc{Percentile}(X, p)$ {p-th percentile threshold}
$M \leftarrow \mathbb{1}\{X \geq \tau\}$ {proposal mask}
$M \leftarrow \textsc{Close}(M)$
$M \leftarrow \textsc{RemoveSmall}(M, A_{\min})$
$\mathcal{C} \leftarrow \textsc{ConnectedComponents}(M)$
if $|\mathcal{C}| = 0$ **then**
  return $0_{H \times W}, \emptyset$
end if
$c^\star \leftarrow \textsc{LargestComponent}(\mathcal{C})$
if not $\textsc{ConfidenceOK}(c^\star, X, \mathcal{C})$ **then**
  return $0_{H \times W}, \emptyset$
end if
$M \leftarrow \mathbb{1}\{\text{pixel} \in c^\star\}$
$B \leftarrow \textsc{BBox}(c^\star)$

return M, B

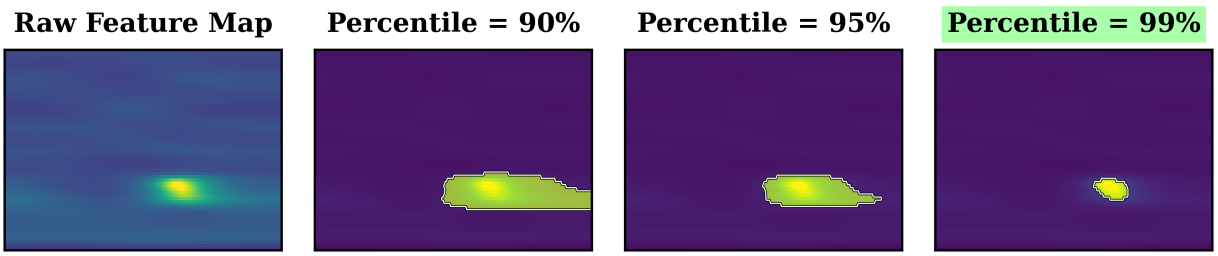

Figure 4.7: Percentile ablation on a representative Range–Azimuth frame (*lying on sofa*). From left to right: normalized RA image and detection masks obtained by keeping only the top 90%, 95%, and 99% of pixel intensities.

Table 4.3: Detection accuracy (%) per Subject. Best values for each metric per Subject are shown in **bold**.

	Subject 1			Subject 2			Subject 3		
Activity	CA	OS	**Ours**	CA	OS	**Ours**	CA	OS	**Ours**
Standing	79.0	89.9	**93.5**	45.0	65.7	**91.7**	80.0	91.2	**97.3**
Sit on Sofa	75.7	88.2	**97.0**	31.4	51.6	**86.9**	57.0	72.2	**92.7**
Sit on Floor	83.3	91.5	**97.2**	54.6	68.4	**94.1**	74.7	82.9	**97.4**
Lay on Floor	70.6	80.5	**91.9**	63.9	77.0	**94.7**	78.4	86.4	**93.7**
Lay on Sofa	33.0	43.9	**86.6**	61.8	79.0	**94.1**	48.5	67.0	**93.0**
Average	68.3	78.8	**93.24**	51.3	68.3	**92.3**	57.72	69.94	**94.82**

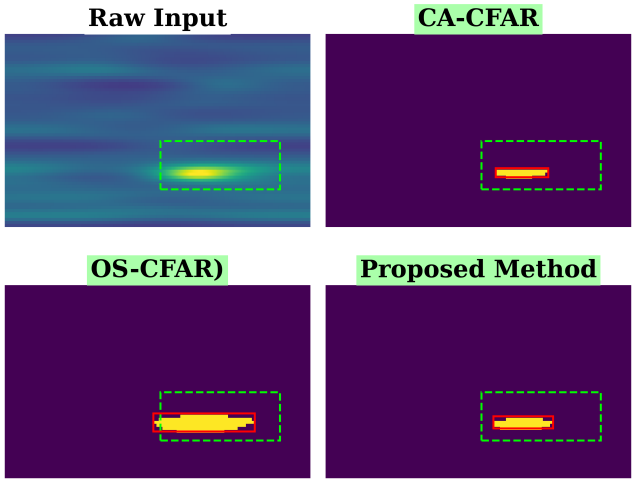

Figure 4.8: Qualitative comparison of all three detectors on frame 21 for subject 1 lying on the floor, which all of them produce detections within the ground-truth bounding box.

ers: we count a frame as a hit if one detected blob overlaps the annotated ground–truth bounding box. A green highlight indicates a successful detection, while a red highlight signifies a miss. While all methods successfully localize the subject within the ground-truth region during the high signal-to-noise ratio (SNR) scenario (Frame 21, Figure 4.8), the proposed algorithm demonstrates superior stability under severe noise. As shown in Frame 106 (Figure 4.9), strong background clutter causes both CA-CFAR and OS-CFAR to fail, whereas our proposed method succeeds in detecting the subject within the ground-truth bounds. Table 4.3 reports per–activity detection accuracy for three subjects. On average, the proposed method improves detection accuracy to 93.24% (Subject 1), 92.3% (Subject

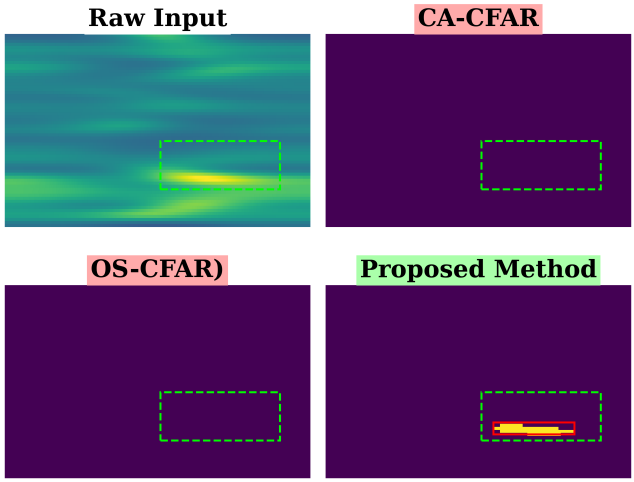

Figure 4.9: Qualitative comparison of all three detectors on frame 106 for subject 1 lying on the floor. Only our proposed method can detect within the ground-truth bounding box.

2), and 94.82% (Subject 3), significantly outperforming CA-CFAR (68.3%/51.3%/57.72%) and OS-CFAR (78.8%/68.3%/69.94%). The accuracy in the Lay on Sofa case increases

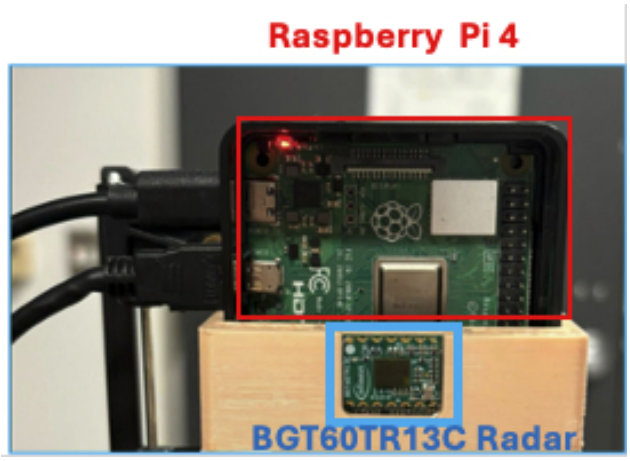

Figure 4.10: Raspberry Pi 4B Board with BGT60TR13C radar.

from 33.0% (CA) and 43.9% (OS) to 86.6% for Subject 1, and 48.5% (CA),67.0%(OS) to 93% for Subject 3. For Subject 2, the increase is from 31.4% and 51.6% to 86.9% for Subject 2 in Sit on Sofa. Notably, sofa-based activities remain the most difficult, representing the lowest detection performance for Subject 1 (lying) and Subject 2 (sitting),

Table 4.4: Average runtime analysis on Raspberry Pi 4 for 3 methods. Best values for each metric are shown in **bold**.

Method	Latency (ms)	FPS	RAM (MB)	Speedup (vs. OS)
CA-CFAR	91.2	11.0	299.4	6.6×
OS-CFAR	601.1	1.7	299.5	1.0×
Ours	**8.2**	**122.4**	299.5	**73.6**×

respectively. Finally, to assess real-world feasibility and performance, we benchmark all three detectors across all activities using a streaming implementation on a Raspberry Pi 4B [71] as Figure 4.10, and the results are shown in Table 4.4. Range-Azimuth feature maps are precomputed using the shared pipeline and then fed to each detector. Compared to CA-CFAR (91.2 ms) and OS-CFAR (601.1 ms), our proposed method runs on average of just 8.2 ms per frame, achieving a throughput of over 122 FPS. This represents a 74× speed-up over OS-CFAR and ensures the system easily meets the 10 Hz radar frame-rate constraint without deadline misses. These results support the chapter's central argument that image-based processing delivers both robust detection accuracy and real-time computational efficiency, useful for real-world radar sensing. The proposed detector outperforms CA-CFAR and OS-CFAR across all three subjects, with particularly large gains in difficult sofa-related cases, while simultaneously reducing computation to a level suitable for embedded real-time execution. This is important in LTC settings, where scalability across many rooms depends not only on sensing accuracy but also on low latency, low computational overhead, and the ability to run continuously on inexpensive edge hardware.

4.4 Conclusion

This chapter examined how image-based processing can improve radar-based occupancy sensing in cluttered LTC environment. We first showed that the effectiveness of preprocessing methods cannot be understood adequately through error-based metrics alone. In low-SNR conditions, perceptual and structural criteria provide a more meaningful view of whether task-relevant motion signatures are preserved, motivating a broader representation-quality perspective for radar sensing. Building on this insight, we consider quasi-static sensing, where a spatial range-angle feature map provides more appropriate information. For these representations, we introduced a no-reference quality scoring method that quanti-

fies whether an RA map contains a dominant clean and interpretable subject response, and a temporal tracking algorithm that stabilizes detections when individual frames are unreliable. Finally, we translated these representation-level insights into a lightweight detection algorithm for quasi-static human presence.

Chapter 5

Digital-Twin Simulation and Calibrated Sim-to-Real Radar Sensing for Occupancy and Fall Analysis

5.1 Chapter Motivation and Scope

The previous chapters focused primarily on methods developed and validated using measured radar data collected in realistic indoor environments. While such measurements are essential, they also reveal a practical limitation of radar research in assisted-living settings: large-scale data collection is costly, time-consuming, and often difficult to control [72, 73]. Small changes in room layout, furniture placement, wall material, subject trajectory, or radar placement can alter the observed multipath and clutter structure [74, 75]. These challenges are especially pronounced when studying rare or safety-critical events such as falls, where collecting large, diverse, and clinically representative datasets from vulnerable populations is ethically and operationally difficult [76, 77]. Electromagnetic (EM) digital twins provide a complementary way to study these problems before or alongside real deployment. By explicitly modelling room geometry, material properties, radar hardware, and human motion, a digital twin can generate synthetic radar measurements under controlled and reproducible conditions. This makes it possible to vary one factor at a time, such as bandwidth, viewpoint, clutter configuration, or human trajectory, while holding the rest fixed. As a result, digital twins support tasks that are difficult to perform with

measured data alone, including pre-deployment system qualification, data augmentation under sparse labels, representation-level ablation, and controlled sim-to-real analysis.

This chapter studies two complementary uses of digital twins for radar sensing. Section 5.2 investigates a calibrated sim-to-real framework for occupancy estimation, where simulated corridor measurements are brought closer to real data through calibrated domain randomization and noise-floor matching. Section 5.3 then turns to fall-related sensing and uses a digital twin as a controlled testbed for studying how different radar representations behave in a fall-versus-non-fall setting. Together, these two studies broaden the thesis from measured-data processing toward simulation-assisted radar design, showing how digital twins can improve scalability, reduce data-collection burden, and support more systematic evaluation of sensing pipelines before real-world deployment.

5.2 Digital Twin Framework for Radar Occupancy Estimation

A physics-informed digital twin of the monitored corridor was developed using a EM simulation platform that combines Shooting and Bouncing Rays (SBR) and Physical Optics (PO) methods [73, 78]. The corridor environment, including walls, ceiling, and floor, was reconstructed with measured dielectric properties to capture realistic multipath propagation and clutter effects. A virtual 60 GHz FMCW radar modelled after the Infineon BGT60TR13C sensor [15] replicated the antenna configuration, polarization, and chirp parameters of the physical hardware. More specifically, the simulated radar was configured to match the deployed hardware in carrier frequency, bandwidth, chirp timing, receive configuration, and mounting pose. This matching is important because our main goal is to investigate whether data generated from a physically consistent geometric twin can be transferred to the deployed sensing system with minimal real supervision. The simulator, therefore, will be a controllable source to generate labelled RD data under scenarios that would be expensive, repetitive or tedious to collect. In this experiment, the radar was emulated with a carrier frequency of 60 GHz and a bandwidth of 882.35 MHz. The sampling rate and chirp duration were set to 1 MHz and 0.256 ms, respectively, yielding 256 samples per chirp and a chirp slope of approximately 3.44×10^{12} Hz/s. These parameters correspond to a theoretical range resolution of 0.17 m and a maximum unambiguous range of 21.7 m, consistent with the operating characteristics of the real device.

Human motion dynamics were introduced using animated 3D humanoid meshes derived from Mixamo and CMU Motion Capture datasets [79, 80]. These animated models

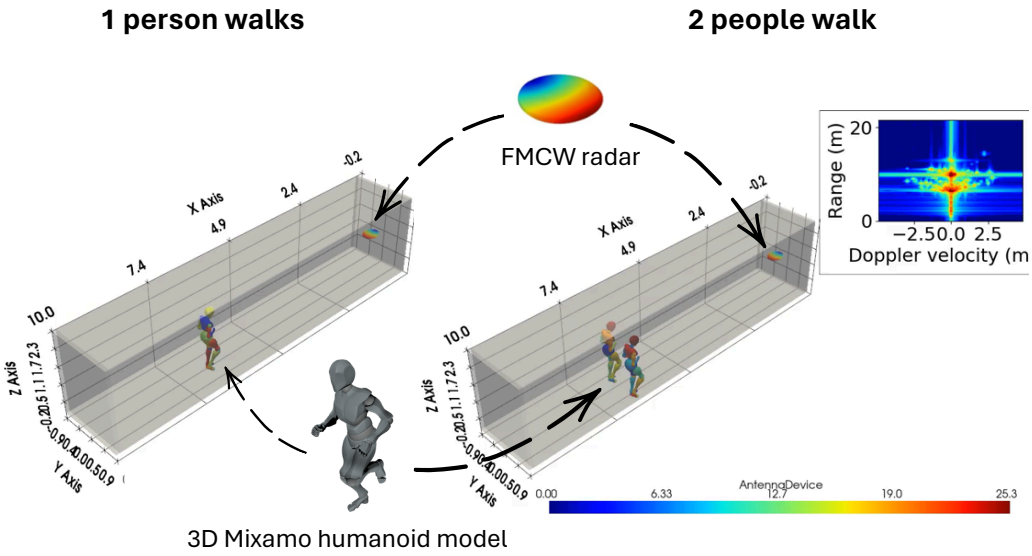

Figure 5.1: Simulation setup showing the created radar scenes. A 3D Mixamo humanoid model is animated to represent walking motion and imported into the virtual corridor environment. The FMCW radar observes different scenarios—(left) a single walking subject and (right) two subjects walking at different speeds, producing corresponding range–Doppler signatures.

provided time-varying radar cross sections and realistic trajectories for walking and interaction behaviours. These trajectories were selected to reflect the same behaviours later used in the real measurement scenarios: an empty corridor, one subject walking toward the radar, and two subjects walking together. This design makes the simulation and measurement domains semantically consistent while still preserving the practical domain gap caused by clutter, sensor noise, and environmental mismatch. The EM solver iteratively computed scattered fields through multiple ray interactions until the residual reflected power dropped below a predefined threshold. The simulated radar outputs were organized as complex data cubes containing amplitude and phase information across fast time, slow time, and frequency dimensions. The complex radar cubes were then post-processed into RD representations as Figure 5.1 for downstream learning. In the simulation branch, these outputs were exported as floating-point arrays and treated as the labelled source domain for training. Throughout the remainder of this thesis, we refer to this source generation process as the **Ray-Tracing (RT) baseline** or **RT simulation**. This emphasizes that the simulator already captures meaningful physics, scene geometry, and produces RD maps

whose global background statistics differ from those of the real corridor. This observation motivates the calibrated sim-to-real alignment strategy introduced later in this Section 5.2.

5.2.1 Experiment Setup

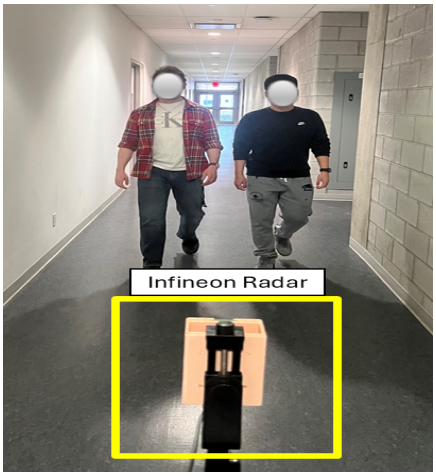

Figure 5.2: Experiment setup for 2 people walking scenario.

A real-world FMCW radar dataset is collected in the corridor using the same Infineon BGT60TR13C radar [15] and configuration specified above. We record three occupancy states: empty room, one person walking, and two people walking, as seen in Figure 5.2. An additional short unlabeled calibration sequence (10 seconds or 100 frames) of an empty corridor is acquired once and used only to estimate the real noise-floor statistics for Calibrated Domain Randomization (CDR) (Section 5.2.2). This calibration split is created deliberately to separate *environment calibration* from *task supervision*. In other words, the empty-room sequence is not labelled for learning occupancy classes and is never used to train the classifier on real target examples. Instead, it is used only to summarize the background clutter statistics of the deployment environment. This distinction is important because the proposed method aims to reduce the amount of real data labelled required for deployment. Both simulated and real data are applied with a 2D FFT applied across fast-time and slow-time dimensions to obtain a complex Range-Doppler map, and their magnitudes are converted to dB [81]. For the real radar data, the preprocessing pipeline starts from JavaScript Object Notation (JSON)-encoded complex radar cubes. For each frame, the in-phase and quadrature components are reconstructed for all receive channels. Per-chirp DC removal is first applied, followed by Blackman–Harris windowing in range

and Doppler, zero-padding, and a 2-D FFT similar to preprocessing we demonstrated in Section 2.1.1. The resulting RD map is FFT-shifted along Doppler so that zero Doppler lies at the center of the spectrum. To improve robustness against isolated artifacts, we further apply a small median filter and Gaussian smoothing before storing the RD magnitudes in dB. The real data processing stage does not go through any randomization and is used exclusively for testing purposes. They are then clipped to a fixed dB range $[v_{\min}, v_{\max}]$ estimated from simulation, normalized to $[0, 1]$, and mapped via a perceptually uniform colormap (viridis) to form 3-channel images for the input of the neural network model (ResNet18 with random initialization and a task-specific classifier head). The final fully connected layer is replaced with a task-specific classifier: 2 units for binary occupancy or 3 units for people counting applications.

In the source-domain branch, the same clip–normalize–colormap pipeline is applied to simulated RD maps so that both synthetic and real inputs are projected into the same image representation before entering the CNN. Importantly, the clipping range is estimated from the simulated training set only and then reused for all domains and all methods. This ensures that any performance improvement cannot be attributed to re-scaling the real test set differently, but instead arises from better alignment of the underlying RD statistics before normalization. Training is performed on simulated RD maps: either (i) RT simulation, (ii) RT simulation with Random Domain Randomization (DR) (Random DR), or (iii) RT simulation with the proposed CDR. The real labelled test set is held out entirely for sim-to-real evaluation. Real RD maps at test time are passed through the same clip-and-normalize pipeline as the simulated inputs (Figure 5.3). To further enforce a clean evaluation protocol, the processed real dataset is split into two parts. First, 100 empty-room frames are sampled once to form the calibration subset used to estimate the CDR statistics. Second, the remaining empty-room frames are combined with all one-person and two-person measurements to form the final held-out test set. This split guarantees that real occupancy examples are never used for training or calibration, and that the final results reflect genuine sim-to-real transfer.

5.2.2 Noise-Floor Domain Randomization and CDR

Figure 5.3 shows an overview of the proposed sim-to-real pipeline. The yellow/green branch corresponds to the simulated data flow: the geometric twin generates synthetic radar scenes, these scenes are converted to range–Doppler maps, and the resulting RD maps are optionally modified by a noise-floor randomization block before clipping, normalization, colour mapping, and CNN training. The purple branch corresponds to the calibration flow: a small, unlabeled set of real empty-room radar frames is converted to RD maps

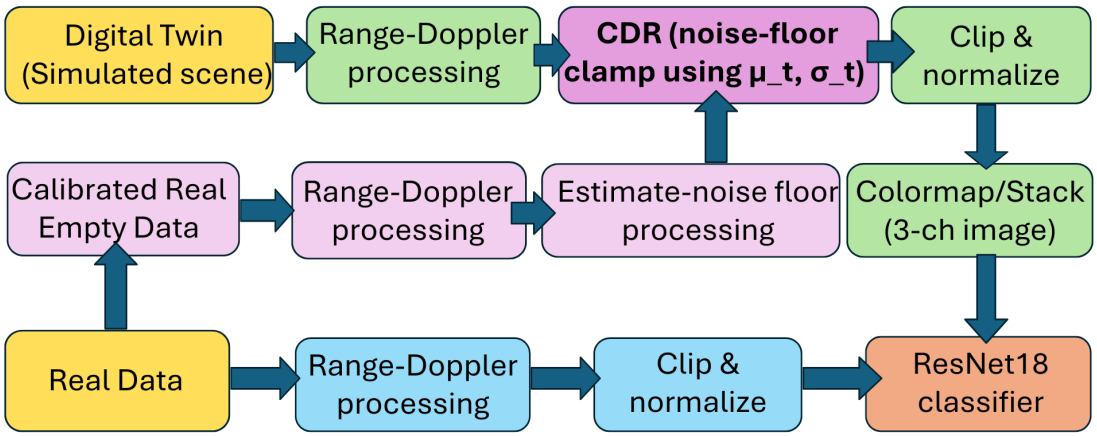

Figure 5.3: Overview of the proposed sim-to-real pipeline. **Yellow blocks:** Start of simulated data and real data preprocessing **Green path**: simulated radar frames from the digital twin are converted to range–Doppler (RD) maps and passed through the noise-floor CDR block, then clipped, normalized, and colour-mapped for network training. **Purple path**: CDR uses 10 seconds of unlabeled real empty-room data to estimate the target noise-floor statistics that drive the calibration. **Blue path**: real radar frames for evaluation are converted to RD maps and only clipped and normalized, without any randomization or CDR modification.

and used only to estimate the target background statistics that drive the calibrated randomization. The blue branch corresponds to deployment and evaluation: real RD maps are processed with the same RD conversion, clipping, and normalization, but are not altered by any randomization at test time. This figure is central to the method because it makes clear where the proposed contribution lies. The simulator, RD processing, clipping, normalization, colormap conversion, and CNN architecture are shared across methods. The only component that changes between the Naive baseline, Random DR, and CDR is the noise-floor randomization block inserted in the simulated branch before normalization. This isolation is important because it allows the effect of calibrated noise-floor alignment to be studied independently of architectural or training-schedule changes. Inspired by DR in computer vision and robotics [82–84], our approach introduces synthetic noise to bridge this gap. Both our baseline Random DR and a proposed CDR method apply a per-frame, per-cell clamp in the dB domain, before clipping and normalization:

$$\tilde{S}(i,j) = \max\big(S(i,j),\, N(i,j)\big). \tag{5.1}$$

where $S(i, j)$ denotes the simulated RD magnitude at range bin i and Doppler bin j, and $N(i, j)$ is a sampled noise value. This formulation is intentionally simple. If a simulated RD cell already contains a strong target return, then the maximum operation leaves it unchanged. However, if the simulated cell is unrealistically dark compared to the clutter observed by the deployed radar, the clamp raises it to a more plausible background level. Thus, the transform primarily modifies low-power cells associated with clutter, static background, and noise floor, while preserving stronger human-related returns. Since the transform is applied directly in DR space before image conversion, the resulting changes propagate through the entire downstream learning pipeline. The two methods differ only in how they sample the noise map $N(i, j)$. For the Random DR baseline, we emulate generic domain randomization by sampling:

$$N(i, j) \sim \mathcal{N}(m, s^2) \tag{5.2}$$

with mean m from $[-130, -100]$ dB and a standard deviation s from $[1, 4]$ dB. These ranges approximate typical background levels and variability observed in our quantitative indoor FMCW experiment. This baseline is deliberately strong: it introduces stochastic clutter over a broad range of possible backgrounds without relying on any real data. In this sense, Random DR captures a common sim-to-real strategy in which one broadens the source distribution in the hope that the target domain will fall inside it. However, the choice of m and s remains heuristic, and there is no guarantee that the resulting synthetic clutter distribution matches the actual deployment environment. CDR keeps the same clamping form (5.1) but replaces heuristic sampling with a one-shot, data-driven calibration from the unlabeled real empty-room calibration set by aggregating all RD magnitudes and computing their empirical mean m_t and standard deviation s_t. CDR then samples

$$N_t(i, j) \sim \mathcal{N}(m_t, s_t^2) \tag{5.3}$$

i.i.d. over (i, j) and applies the same clamp $\tilde{S}(i, j) = \max(S(i, j), N_t(i, j))$. Here, m_t and s_t are obtained once from the unlabeled empty-room calibration frames and represent the measured average background level and variability of the deployed radar. CDR, therefore, differs from Random DR only in how the noise map is generated: instead of using heuristic ranges, it uses a small real calibration set to anchor the randomization to the target environment. No real occupancy labels are required, and no architecture changes are introduced.

Figure 5.4 shows the global histogram of RD magnitudes for naive simulation, real empty-room measurements, and CDR-processed simulation. The x-axis is RD magnitude in dB, and the y-axis is shown on a logarithmic density scale. We use the logarithmic

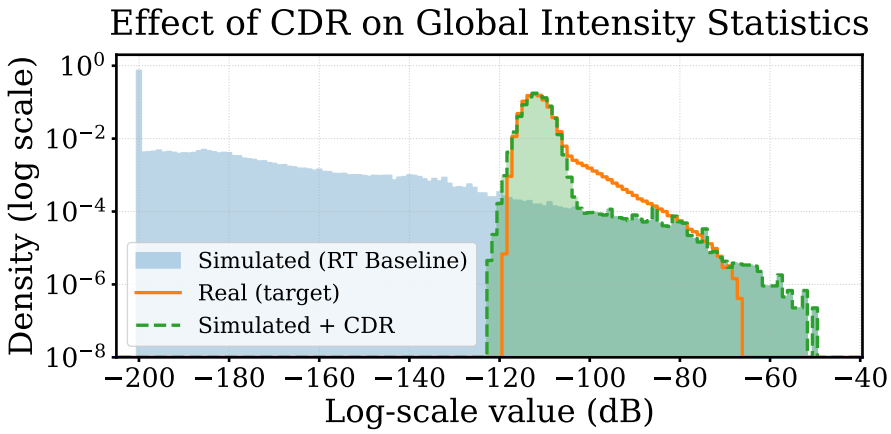

Figure 5.4: Effect of CDR on global RD magnitude statistics. The histogram compares naive simulation, real empty-room measurements, and CDR-processed simulation. The horizontal axis shows RD magnitude in dB, while the vertical axis uses logarithmic density to reveal both the dominant low-noise peak and lower-density clutter tails. Naive simulation is concentrated at unrealistically low background levels, whereas CDR shifts the simulated distribution toward the real empty-room statistics.

vertical axis because the distributions are highly imbalanced: naive simulation contains a very large concentration of samples near an extremely low noise floor, while the informative differences between clutter tails and mid-density regions would be visually compressed on a linear scale. In the figure, the naive simulated distribution is sharply concentrated at very low dB values, indicating an unrealistically clean background. After CDR, the simulated histogram shifts toward the real empty-room distribution and exhibits a broader tail structure more consistent with CDR, which corrects the noise-floor mismatch that motivates the method.

Figure 5.5 complements this global view with representative frame-level RD maps. The first column shows baseline simulated RD maps for the empty-corridor and one-person walk toward radar cases; the second column shows the same simulated frames after CDR; and the third column shows corresponding real measurements. The purpose of this figure is to demonstrate that CDR fills in the unnaturally dark and artifact-free simulated backgrounds with clutter-like texture while preserving the dominant target structure. In the empty corridor case, CDR creates background fluctuations that better resemble the deployed corridor. In the one-person case, CDR preserves the localized person-related RD response while embedding it in a more realistic background field. This visual evidence supports the idea that the clamp is acting on the correct part of the distribution: low-power background

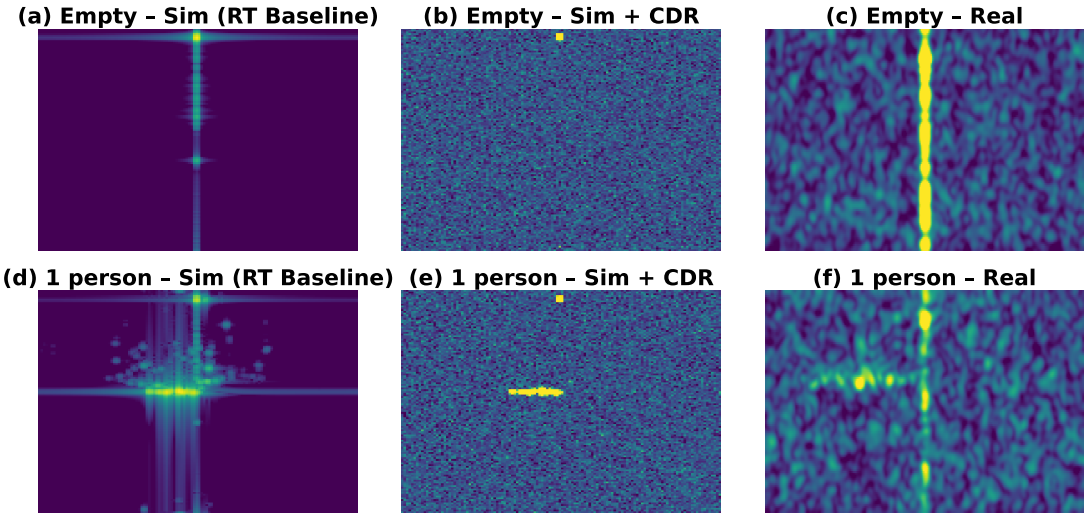

Figure 5.5: Range-Doppler maps for (a,d) baseline simulation, (b,e) simulation after CDR, and (c,f) real measurements for the empty room and one-person walking.

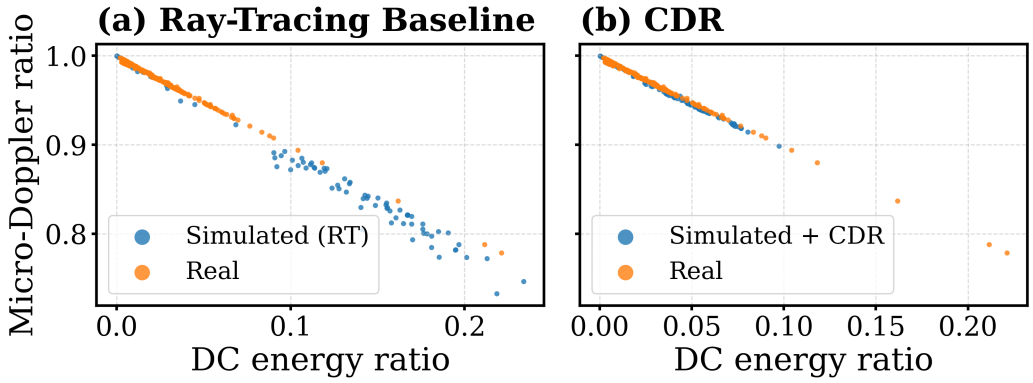

Figure 5.6: Static-to-total energy ratio in RD space. Each point is one frame. CDR moves simulated points toward the real cluster, indicating that the global clutter level has been calibrated.

cells are adjusted, whereas strong target responses remain visible.

Figure 5.6 provides a compact summary of this alignment using energy statistics derived from each RD frame. For every frame, we compute the fraction of total RD energy located in the zero- and near-zero-Doppler bins (static / clutter energy) and compare it with the complementary micro-Doppler energy fraction. Each point in the scatter plot

represents one frame. In the naive baseline, simulated frames occupy a different region of this space from real frames, reflecting that the static clutter level of the simulator is too weak. After CDR, the simulated points move toward the real cluster, indicating that the balance between static clutter and dynamic human-related energy is more realistic. This is crucial since occupancy detection and people counting also depend on the surrounding clutter context seen by the classifier.

5.2.3 Results

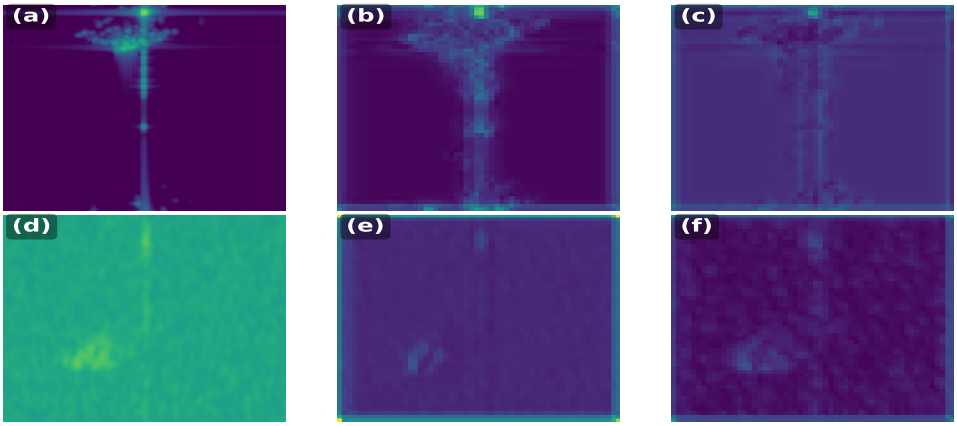

Figure 5.7: Qualitative visualization of early feature maps from the first residual block of ResNet-18 for representative simulated and real radar range-Doppler inputs. (a,d) inputs; (b,e) features from the baseline RT model trained only on unmodified simulation; (c,f) features from the CDR-trained model.

Figure 5.7 investigates how these statistical changes affect the learned representation. We visualize channel-averaged activations from the first residual block of ResNet-18 for representative simulated and real RD inputs. The inputs themselves are shown in (a,d), the activations of the baseline RT model trained on unclamped simulation are shown in (b,e), and the activations of the CDR-trained model are shown in (c,f). The baseline model exhibits diffuse responses and is sensitive to simulator-specific patterns, whereas the CDR-trained model produces more compact, human-centred activations while suppressing background artifacts. This figure supports the interpretation that CDR does not merely improve global statistics; it also changes the internal feature representation in a way that is more stable across simulation and reality.

On the real corridor test set, the baseline RT model trained solely on unmodified simulation struggles to generalize. Figure 5.8 shows that it misclassifies many occupied frames as empty due to mismatched clutter and noise, resulting in a balanced accuracy close to 50% (chance-level for a binary task). This failure mode is consistent with the earlier distribution plots. Since the naive simulator produces backgrounds that are too clean, the network learns a decision boundary that is overly sensitive to isolated target peaks and insufficiently exposed to realistic corridor clutter. When deployed on measured radar data, the background statistics shift substantially, and the model misclassifies real occupied scenes with the empty class. The introduction of uncalibrated Random DR improves the result: the model becomes less sensitive to background mismatch and achieves balanced accuracy around 83%. However, this improvement comes with a trade-off. Because Random DR injects clutter sampled from a broad heuristic range, the resulting simulated RD maps may not always resemble physically plausible corridor observations. In practice, some training samples become overly noisy or structurally distorted, limiting how much the classifier can benefit from the added diversity. Random DR, therefore, demonstrates that noise-floor variation matters and uncontrolled variation is not sufficient for optimal transfer. CDR provides a further and substantial improvement. By aligning simulated noise floors to real statistics while preserving micro-Doppler content, the CDR-trained model achieves balanced accuracy $\approx 97\%$ and high overall accuracy. The confusion matrices (Figure 5.8) show that CDR dramatically reduces both false negatives and false positives compared to the baselines. The confusion matrix analysis in Figure 5.8 is particularly informative for the binary task. The RT baseline predicts nearly all frames as occupied, showing that it has failed to learn a robust distinction between empty-room clutter and real target presence. Random DR recovers much of this distinction, but still produces noticeable errors. Finally, CDR yields the cleanest separation, with most empty-room frames correctly classified and most one-person frames preserved. The gain is due to better calibration of the simulated source domain.

The people counting task is more challenging due to the overlap between one- and two-person signatures and limited data. Simulated training remains feasible by using the same pipeline, while the real test set is used exclusively for evaluation. As shown in Figure 5.9, the baseline RT model again fails to transfer, achieving balanced accuracy near 33% (random guessing) Random DR improves performance to balanced accuracy $\approx 61\%$, indicating that generic noise-floor perturbation offers some regularization but does not fully align domains, and heavily confuses among 3 classes. CDR yields the best performance with balanced accuracy $\approx 72\%$ and improved confusion matrices, particularly for distinguishing one vs. two-person cases. Misclassifications mainly occur in ambiguous scenes with partial overlap or occlusion in Doppler. The 3-class confusion matrices in Figure 5.9 show that the

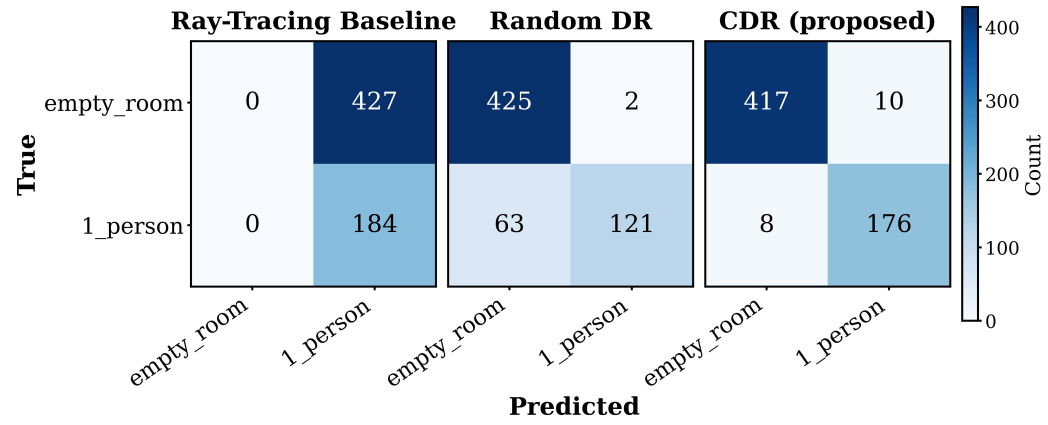

Figure 5.8: Confusion matrices for real binary occupancy detection across 3 methods.

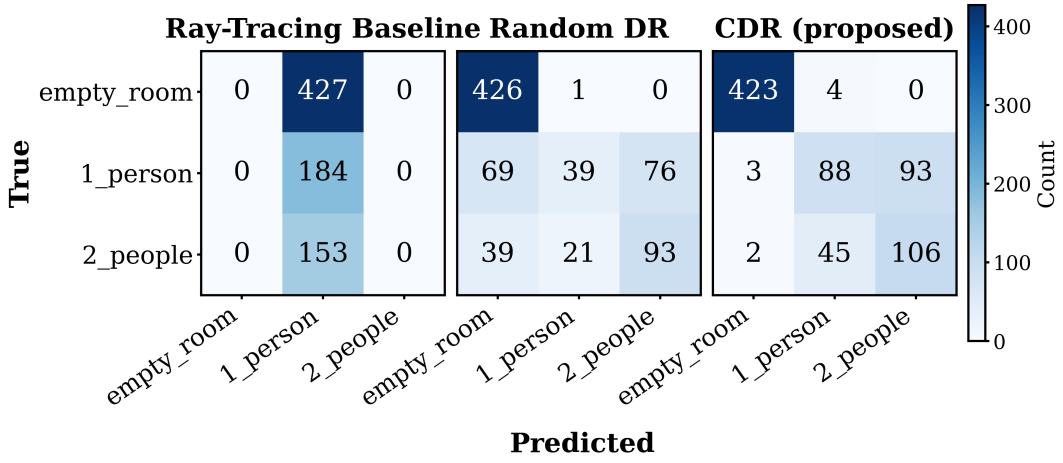

Figure 5.9: Confusion matrices for real people counting across 3 methods.

main challenge is separation between one-person and two-person walking. These classes share overlapping Doppler extents, similar bulk motion, and only limited differences in return strength when subjects partially occlude one another. Even under this more difficult setting, CDR consistently improves the diagonal structure of the confusion matrix, reducing confusion between adjacent occupancy levels. This suggests that calibrated clutter alignment not only helps the model detect the presence of a target but also stabilizes the representation enough to preserve finer-grained occupancy information.

5.2.4 Conclusion

Taken together, the results from Figures 5.4–5.9 support a coherent interpretation. Figure 5.4 establishes that CDR corrects the global magnitude distribution. Figure 5.5 shows that this correction produces visually more realistic RD maps. Figure 5.6 confirms that the correction also aligns the static-to-dynamic energy structure. Figure 5.7 demonstrates that the learned early features become more transferable. Finally, Figures 5.8 and 5.9 show that these improvements translate into better real-world occupancy detection and people counting. This progression from global statistics, to frame-level structure, to feature representation, and finally to classification performance is central evidence for the effectiveness of CDR.

5.3 Digital-Twin Representation for Fall Detection

Section 3.2 examined post-fall floor-occupancy detection using measured data from a furnished LTC-like environment. That study asked whether a radar system could reliably detect that a person remained quasi-statically on the floor after a fall-like event, under realistic clutter and viewpoint variation. The present section addresses a complementary question: before collecting large measured fall datasets, can a digital twin be used as a controlled testbed for studying how different radar representations respond to fall and non-fall motion patterns? In Section 3.2, the focus was on persistent floor-level occupancy after the event, using real measurements and deployment-oriented detection criteria. Here, the focus is on controlled representation analysis in a synthetic setting, where full fall and non-fall trajectories can be generated repeatedly without exposing human subjects to risk. By varying motion patterns systematically while fixing the network architecture, preprocessing pipeline, and train/test split, the digital twin allows representation-level effects to be studied in isolation from many of the uncontrolled factors that arise in measured-data comparisons.

EM Digital Twins (DT) offer a way to probe such design choices before large measurement campaigns. By combining human motion libraries, 3D room geometries, and radar front-end models, DTs generate synthetic radar measurements under controlled, reproducible conditions [72, 85–89]. A radar DT can simulate fall and non-fall trajectories at multiple positions and yaw angles without exposing human subjects to risk, and can be used as a testbed for examining how different representations behave when motion patterns are varied systematically. By fixing the CNN architecture, preprocessing pipeline, and train/test split across all experiments, this study isolates representation-level effects

that would be difficult to disentangle from architectural choices or environmental variability in measured-data comparisons.

In this section, the digital twin serves three purposes. First, it provides a balanced synthetic fall/non-fall dataset in a furnished assisted-living-style room using a Perceive-EM-based workflow and a ceiling-mounted 60 GHz FMCW radar model under a fixed stratified train/test split. Second, it enables a controlled comparison of three input representations derived from the same range–Doppler sequence: an all-receiver spectrogram (`spec`), a per-receiver spectrogram stack (`spec3`), and a time-pooled range–Doppler map (`rdm`), all evaluated using the same compact CNN backbone and consistent training settings. Third, it supports a qualitative synthetic-versus-measured comparison to determine which aspects of the fall signature are reproduced faithfully by the twin and which mismatches remain, particularly in amplitude statistics and receiver-dependent effects.

5.3.1 Digital-Twin Simulation Setup

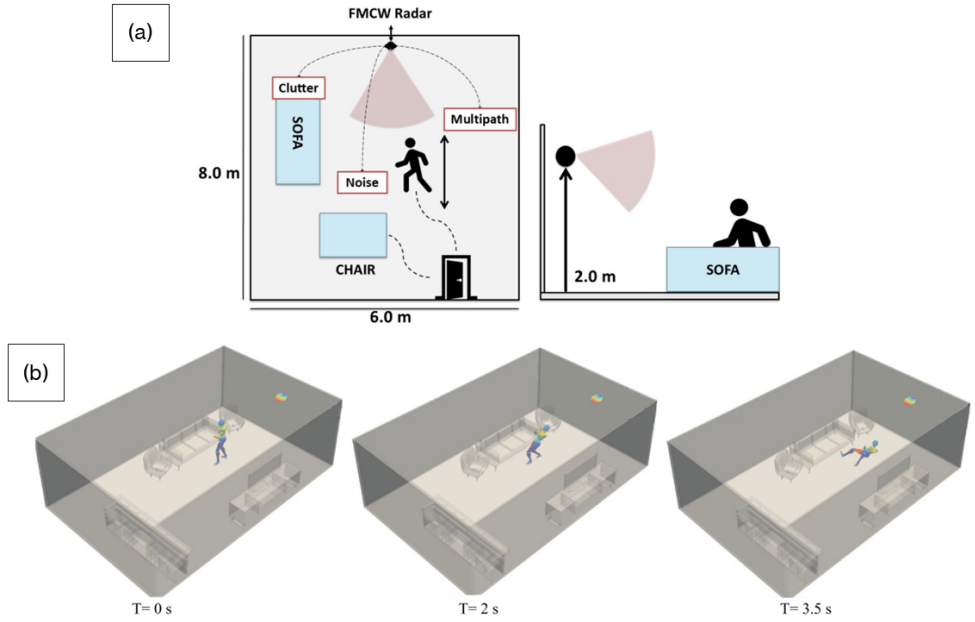

Figure 5.10: Digital-twin simulation setup. (a) Top-down and side-view schematic of the 6×8 m furnished room, showing the ceiling-mounted FMCW radar, its field of view, and example reflecting objects in the modelled environment. (b) Perceive-EM renderings of one fall at three time instants.

The digital twin is used to control geometry and propagation physics while removing measurement noise and subject recruitment constraints. Perceive-EM [90], a commercial EM simulation tool for radar digital twins, uses a SBR solver to model indoor multipath propagation in furnished rooms [85, 87]. Human meshes from Mixamo [79] and room geometry are imported as triangle meshes with assigned 60 GHz material properties. Radar pose, antenna patterns, and FMCW waveform parameters are scripted via a Python Application Programming Interface (API) so that the same configuration can be reused across runs. Figure 5.10 and Figure 5.11 summarize the room layout, position–yaw grid, and 3D digital-twin view. For each simulated clip, the solver traces rays from the transmit aperture through the furnished scene, accumulates field contributions at the receive aperture, and exports complex baseband samples per antenna and per chirp. The receiver chain is idealized: no explicit thermal noise, oscillator phase noise, automatic gain control, or non-stationary clutter suppression stages are modelled, and the analog-to-digital converter is assumed linear over the dynamic range of interest. Under these modelling assumptions, the EM solution is intended to be the dominant source of scene-dependent variability, so differences between clips are driven by motion and pose rather than hardware artifacts.

Virtual Radar Configuration The virtual sensor is a ceiling-mounted 60 GHz FMCW radar with one transmit and three receive channels, configured via an Infineon-style JSON description. The 60 GHz carrier frequency provides compact ceiling-mounted apertures with regulatory availability for indoor sensing applications. The active waveform (mode 1) uses a center frequency of $f_c = 60$ GHz and an FMCW bandwidth of $B = 499.7$ MHz, with $N_\mathrm{s} = 32$ ADC samples per chirp at a 1 mega-sample/s (1 MS/s) sampling rate. Each Coherent Processing Interval (CPI) contains $N_\mathrm{p} = 64$ chirps over a duration of $T_\mathrm{CPI} = 26.3$ ms, so the chirp repetition interval is $T_\mathrm{chirp} = T_\mathrm{CPI}/N_\mathrm{p} \approx 0.41$ ms. The CPI duration balances Doppler resolution against computational cost for real-time processing. The ideal range resolution is

$$\Delta R = \frac{c}{2B} \approx 0.30 \text{ m},$$

sufficient to isolate human targets from furniture, and the post-processing stage uses $N_R = 64$ range bins, covering approximately 0–19 m. Doppler processing uses $N_D = 256$ FFT bins per CPI. With $N_\mathrm{p} = 64$ chirps and wavelength $\lambda = c/f_c = 5$ mm, the theoretical Doppler resolution and unambiguous velocity are

$$\Delta v \approx \frac{\lambda}{2T_\mathrm{CPI}} \approx 0.10 \text{ m/s}, \qquad |v_\mathrm{max}| \approx \frac{\lambda}{4T_\mathrm{chirp}} \approx 3.0 \text{ m/s},$$

and the 256-point Doppler FFT samples this interval. The antenna front-end consists of one vertically polarized parametric transmit patch (Tx1) and three vertically polarized

parametric receive patches (Rx1–Rx3), each modeled with half-power beamwidths of 30.5° in azimuth and 60.5° in elevation and positioned within a few millimeters of the radar platform origin, forming a compact 1×3 array. The transmit channel is excited with 1 W input power; absolute power scaling is later removed by per-sample zero-mean, unit-variance (z-score) normalization of the radar representations, so the study focuses on relative spatial and temporal structure rather than link-budget limits.

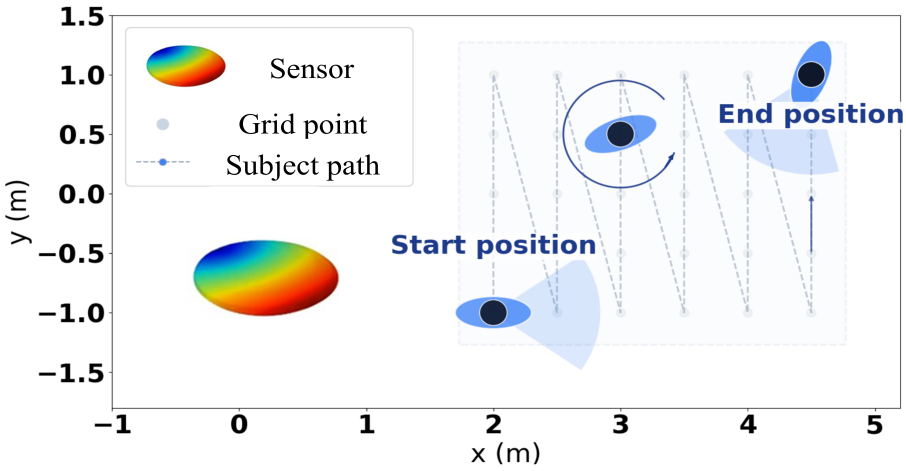

Figure 5.11: Discrete grid over subject position and yaw relative to the radar, with one example fall trajectory shown.

Different Position-Yaw Trajectory Placements The motion library covers a range of fall mechanics and everyday non-fall activities that are plausible in an assisted-living room while keeping class labels simple (fall versus non-fall). Human motions are built from animated meshes derived from the Mixamo database [79], a publicly available character animation library. Falls include knee-first, hip-first, forward, backward, side, stumble/trip, running, and hit-and-fall motions. Non-fall motions include walking, jumping in place, crouching to pick up an object, crouching then standing, sitting and clapping, and a baseball swing. Each motion template is assigned a binary label (fall/non-fall) and a numeric identifier; no severity labels are used in this study.

To probe how representations behave under changing subject pose and orientation, each motion is simulated on a discrete position–yaw grid relative to the radar. The grid spans five range positions, three cross-range positions, and sixteen yaw angles, yielding up to

$5 \times 3 \times 16 = 240$ distinct subject pose configurations (unique position–yaw combinations) per motion. Simulations are launched programmatically via a batch script and Python driver that iterate over motion ID, position, and yaw, so the full grid can be regenerated deterministically. The position–yaw grid and an example trajectory are illustrated in Figure 5.11.

5.3.2 Dataset Construction

Table 5.1: Simulated fall and non-fall motions in the digital-twin dataset.

ID	Motion description	Class	# poses
1	Knee fall forward	Fall	64
2	Hip fall with tuck 1 (backwards)	Fall	62
4	Full-body jump fall backwards	Fall	65
6	Full-body walk trip forward	Fall	57
8	Full-body running fall forward	Fall	67
9	Full-body get hit fall backward	Fall	69
10	Full-body backwards	Fall	56
11	Hip fall landing on buttocks	Fall	70
12	Full-body forward	Fall	54
13	Trip forward with flip	Fall	60
14	Full-body backwards	Fall	64
15	Hip fall with tuck 2 (backwards)	Fall	64
16	Hip fall with tuck 3 (backwards)	Fall	65
17	Forward knees-first fall	Fall	65
18	Full-body sideways fall	Fall	55
19	Backward fall (full body + hip-first)	Fall	57
21	Full-body side fall	Fall	59
22	Diagonal forward knee fall	Fall	63
23	Jump fall backwards	Fall	68
24	Full-body backwards	Fall	66
25	Hip fall backwards with tuck	Fall	67
26	Full-body stumble backwards	Fall	54
27	Knee fall forward	Fall	69
28	Jump up	Non-fall	240
29	Crouch to pick up object	Non-fall	240
30	Sitting clap	Non-fall	240
31	Crouching then standing	Non-fall	240
32	Baseball swing	Non-fall	240
33	Walking forward	Non-fall	240

After simulation, all clips are indexed by scanning the digital-twin output directory and recording, for each motion and pose configuration, the available radar files: all-receiver range–Doppler, all-receiver spectrogram, and three per-receiver spectrograms. Table 5.1 records all metadata with labels (fall/non-fall), motion names, file paths, and pose configuration. Across all motions and poses this yields 6,959 indexed clips, with 5,519 labeled as falls and 1,440 as non-falls. The raw dataset is strongly imbalanced toward falls. To avoid letting the classifier exploit this imbalance, all 1,440 non-fall clips are retained, and 1,440 fall clips are drawn uniformly at random with a fixed seed, producing a balanced set of 2,880 clips. A stratified 80/20 split with respect to the fall/non-fall label then yields $N_{\text{train}} = 2{,}304$ training clips and $N_{\text{test}} = 576$ test clips, with equal numbers of falls and non-falls in each set. This split is generated once and reused for all representation modes so that only the input representation changes between experiments. The split groups different pose configurations of the same motion templates, so the evaluation measures generalization across positions and orientations rather than entirely novel motion patterns.

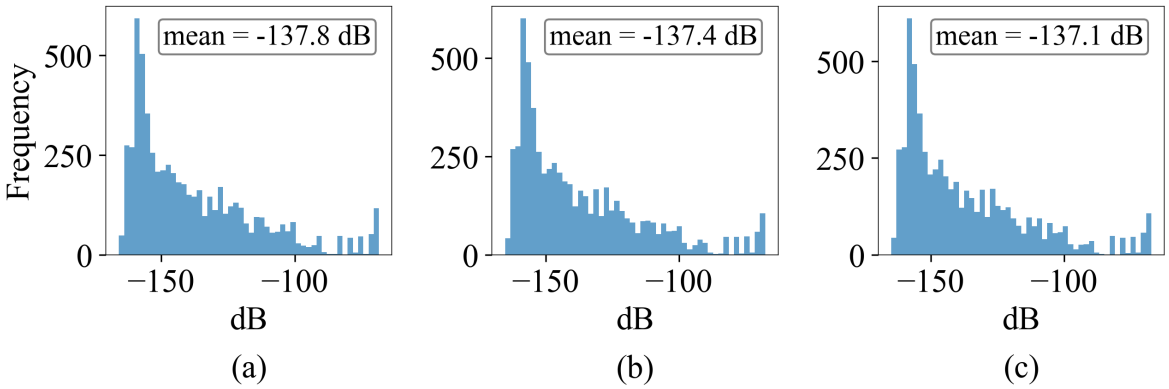

Figure 5.12: Empirical amplitude histograms (in dB) over all pixels for the three per-RX Doppler–time spectrograms. The means are similar but not identical, suggesting that each receive patch sees a slightly different multipath power distribution rather than an exact copy of the same signal.

5.3.3 Processing Method

Standard FMCW processing is applied offline. For each receive channel, a sequence of range–Doppler maps $X_t(r, q)$ is formed by applying a 2-D FFT over fast time and chirps for each CPI, with t indexing the CPIs. Velocity–time spectrograms are then constructed by sampling $X_t(r, q)$ at a fixed range bin $r^\star \approx 3.9$ m and stacking the resulting Doppler

slices over time, converting magnitudes to decibels before resizing and normalization. Three input modes derived from the same radar measurements are compared. All inputs to the CNN are treated as tensors of shape $(C \times H \times W)$, where C is the number of channels, H the vertical dimension (Doppler or range), and W the horizontal dimension.

All-receiver spectrogram (`spec`): For each clip, the digital twin exports an all-receiver Doppler–time spectrogram, which is converted to decibels, cropped or zero-padded along the time axis to 26 frames, and z-scored per sample. The resulting tensor has shape $1 \times 26 \times 256$, where $H = 26$ indexes slow-time frames (one frame per CPI) and $W = 256$ indexes Doppler bins.

Per-receiver spectrogram stack (`spec3`): Three per-RX Doppler–time spectrograms (one per receive antenna) are loaded, converted to decibels, aligned, resized to 26×256, stacked, and z-scored per sample, yielding tensors of shape $3 \times 26 \times 256$. Here $H = 26$ indexes slow-time frames, $W = 256$ indexes Doppler bins, and $C = 3$ encodes per-antenna diversity. The per-RX amplitude histograms in Figure 5.12 differ slightly, consistent with three distinct receive patches.

Static time-pooled range–Doppler map (`rdm`): Let $X_t(r, q)$ denote the complex range–Doppler map at slow-time index t for a given receive channel, where r indexes range bins, q indexes Doppler bins, and $t = 1, \ldots, T$ indexes all CPIs in a clip. A time-pooled magnitude map is defined as

$$X_{\text{pool}}(r, q) = \max_{t=1,\ldots,T} \left| X_t(r, q) \right|, \tag{5.4}$$

Then $X_{\text{pool}}(r, q)$ is averaged over receive channels, converted to decibels, resized, and z-scored per sample. The resulting tensors have shape $1 \times 256 \times 64$, where $H = 256$ indexes Doppler bins and $W = 64$ indexes range bins. Temporal evolution within the clip is removed by the max over t, making `rdm` a static "snapshot" baseline.

Table 5.2: Input dimensions and interpretation of the H and W axes for each radar mode (balanced dataset of 2,880 clips).

Mode	Shape $(C \times H \times W)$	H axis	W axis	Clips
`spec`	$1 \times 26 \times 256$	Time frames	Doppler bins	2,880
`spec3`	$3 \times 26 \times 256$	Time frames	Doppler bins	2,880
`rdm`	$1 \times 256 \times 64$	Doppler bins	Range bins	2,880

Raw clips range from 22 to 53 slow-time frames (approximately 0.6–1.4 s at $T_{\text{CPI}} = 26.3$ ms), depending on motion duration. To ensure uniform input dimensions, each clip

is standardized to $T = 26$ frames by cropping longer sequences or zero-padding shorter ones, yielding a fixed temporal window of approximately 0.68 s. This yields 2,880 clips and $2{,}880 \times 26 = 74{,}880$ frames in total (about 0.55 h of simulated motion) as shown in Table. 5.2. Figure 5.13 shows Doppler–time spectrograms at the fixed range bin for one

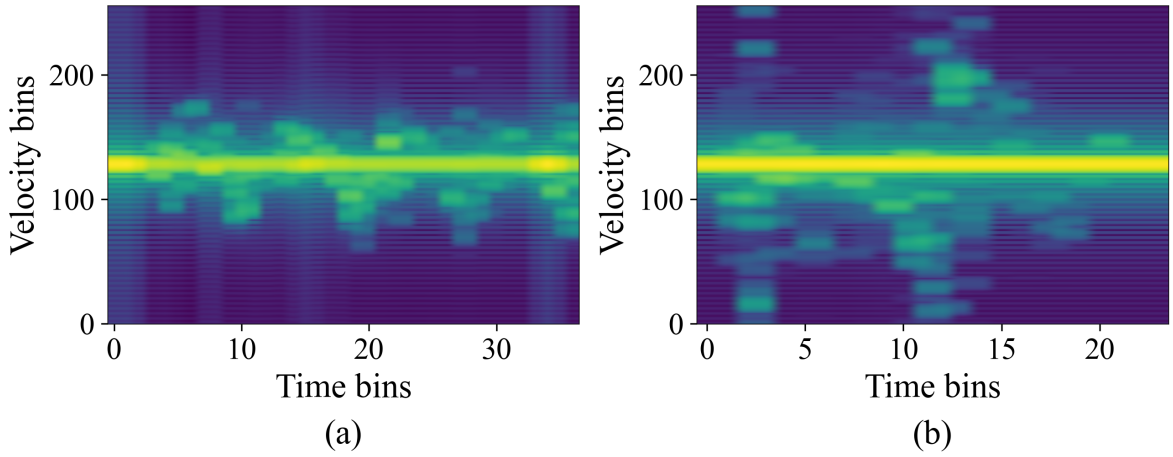

Figure 5.13: Example Doppler–time spectrograms at the fixed range bin $r^\star \approx 3.9$ m for a non-fall (a) and a fall (b) clip in the digital twin.

non-fall and one fall, highlighting the stronger transient velocities during falls.

CNN probe and training protocol All experiments use the same compact CNN, SmallFallCNN, with only the input channel count C adapted to the mode. The network has three convolutional blocks (each: 3×3 convolution, batch normalization, ReLU, and 2×2 max-pooling with 16, 32, and 64 filters) followed by adaptive global average pooling to $(1, 1)$ and a fully connected layer with two outputs (fall / non-fall). For `spec` and `rdm` ($C = 1$) this yields 23,650 parameters; for `spec3` ($C = 3$) it yields 23,938 parameters.

Models are implemented in PyTorch and trained from scratch with cross-entropy loss and the Adam optimizer (learning rate 10^{-3}). Batch size is 16 for single-channel modes and 8 for `spec3`. Training runs for 100 epochs; after each epoch, the model is evaluated on the fixed test split and at epoch 100, confusion matrices and per-class precision, recall, and F1-score are computed.

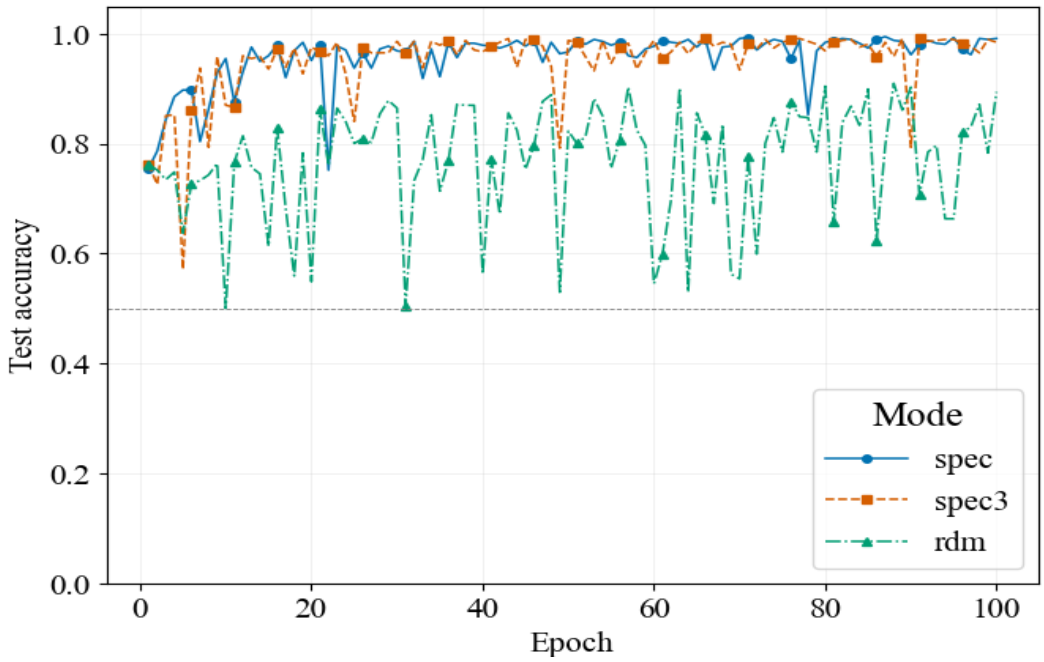

Figure 5.14: Test accuracy versus epoch for each representation mode. The spectrogram modes approach and stay near 99%, while the time-pooled RDM mode follows a lower curve with larger fluctuations over epochs.

5.3.4 Results

All results use the balanced 80/20 train/test split described in Section 5.3.3, with $N_{\text{train}} = 2{,}304$ clips and $N_{\text{test}} = 576$ clips (288 falls and 288 non-falls). For each mode, the corresponding SmallFallCNN is trained under the common protocol and evaluated on the test set after each epoch. Figure 5.14 shows the resulting test-accuracy curves. The spectrogram-based modes (`spec` and `spec3`) pass 95% test accuracy within the first 10–20 epochs and then remain near 98–100% for the rest of training, with only small fluctuations. The static RDM mode (`rdm`) also improves well above chance but exhibits much larger oscillations and settles noticeably below the spectrogram curves.

Table 5.3 summarizes the final test performance after 100 epochs, including overall accuracy and class-wise precision and recall for non-fall (NF) and fall (F). All three modes perform reasonably well on this in-domain synthetic task, but the spectrogram modes are consistently stronger and more balanced across classes. The static RDM mode attains 89.4% accuracy and noticeably lower non-fall recall, indicating more false alarms even

Table 5.3: Test performance on the balanced synthetic test set ($N_{\text{test}} = 576$) after 100 epochs.

Mode	Acc. (%)	Prec. (NF)	Rec. (NF)	Prec. (F)	Rec. (F)
`spec`	99.1	0.983	1.000	1.000	0.983
`spec3`	98.4	0.976	0.993	0.993	0.976
`rdm`	89.4	0.945	0.837	0.854	0.951

though its overall accuracy remains high.

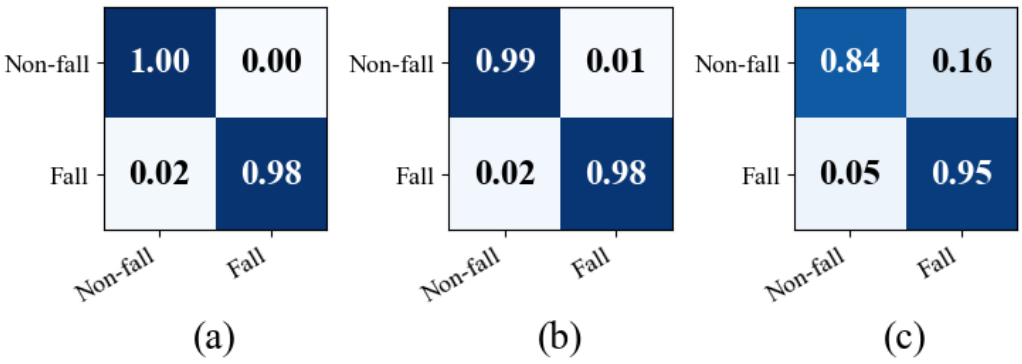

Figure 5.15: Normalized confusion matrices on the balanced test set for (a) `spec`, (b) `spec3`, and (c) `rdm` (rows: true labels, columns: predicted).

Figure 5.15 shows normalized confusion matrices for all three modes. For the spectrogram-based modes, diagonal entries are close to one, and off-diagonal errors are rare. For the RDM mode, fall recall remains high, but non-fall recall drops, so the classifier generates more false alarms (non-falls misclassified as falls) despite a respectable overall accuracy. Under the fixed CNN probe and preprocessing used here, temporal spectrograms therefore provide more reliable fall/non-fall discrimination than the static RDM mode in this digital-twin setting.

5.3.5 Conclusion and Discussion

This section studied fall-related radar sensing from a representation perspective using a EM digital twin. By generating synthetic fall and non-fall sequences in a controlled furnished room, the study isolated how different radar input modes behave when the CNN

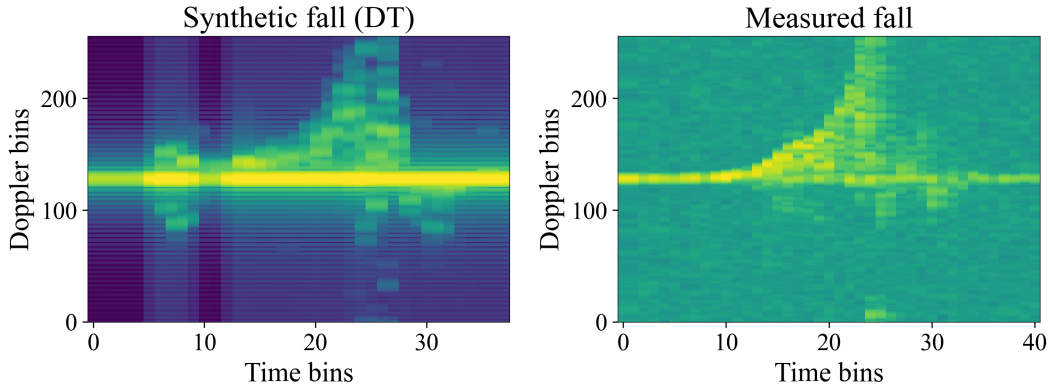

Figure 5.16: Doppler–time spectrograms for one fall: (left) synthetic from the digital twin and (right) measured from a 60 GHz radar. Both show a roughly triangular Doppler–time pattern during descent, followed by near-stationary returns; this is a qualitative check on gross signal structure.

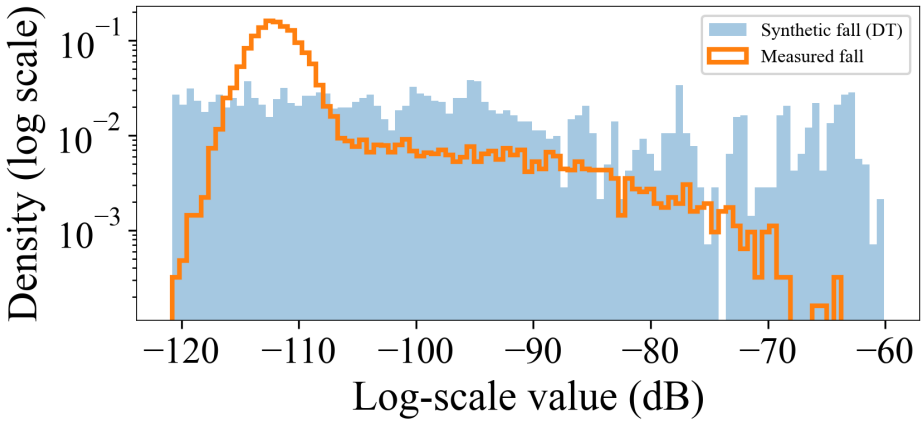

Figure 5.17: Pixel-wise amplitude histograms (log-scale density) for the synthetic and measured fall spectrograms in Figure 5.16. For this example, the measured data show a tighter distribution around the modal value, consistent with receiver processing that the idealized twin omits.

architecture, preprocessing pipeline, and training protocol are held fixed. Under this controlled setup, the spectrogram-based representations achieved stronger and more balanced test performance than the time-pooled range–Doppler-map representation, indicating that temporally resolved Doppler structure carries more discriminative information for this task than a static summary image. To check whether the digital twin generates plausible fall

signatures, Figure 5.16 compares one synthetic and one measured fall spectrogram, and Figure 5.17 compares their pixel-wise amplitude histograms. The Doppler–time patterns during descent are qualitatively similar, but the amplitude distributions differ, with the measured data showing a tighter cluster around a modal value. This is consistent with receiver processing and environmental factors that the idealized twin omits. The comparison is deliberately limited to a single measured example and is intended only as a plausibility check, not as a statistical validation of sim-to-real transfer.

The qualitative comparison between synthetic and measured fall spectrograms further clarified both the usefulness and the limitations of the digital twin. On one hand, the synthetic data reproduced the coarse Doppler–time structure of a representative fall, suggesting that the twin captures the gross motion geometry that drives the dominant spectrogram pattern. On the other hand, the pixel-amplitude histograms revealed non-negligible distributional differences between synthetic and measured data, consistent with receiver processing effects, hardware impairments, and noise characteristics that are not fully modelled by the present simulation pipeline. For this reason, the results of this section should be interpreted primarily as representation-level evidence under controlled synthetic conditions, rather than as final clinical performance claims.

Viewed together with Section 5.2, the fall case study highlights two complementary roles of digital twins in this thesis. In the occupancy study, the twin was used as a data source that required calibration to support sim-to-real transfer. In the fall study, the twin was used more as an analysis environment for controlled comparison of radar representations when large measured fall datasets are difficult to obtain. These two uses are consistent: both treat simulation not as a replacement for real measurement, but as a structured tool for reducing experimental burden, probing design choices systematically, and guiding which representations or preprocessing strategies are worth carrying forward into later real-data studies.

Several extensions follow naturally from this work. A more realistic twin could incorporate additional receiver noise, front-end impairments, and measurement-domain calibration similar to the calibrated domain randomization strategy used in Section 5.2. It would also be valuable to expand the motion library to include a wider variety of clinically relevant fall and non-fall transitions, and to validate the synthetic representations against a larger measured reference set. These directions would move the present case study from qualitative representation analysis toward stronger sim-to-real fall-learning pipelines.

In conclusion, this Section 5.3 used a Perceive EM digital twin to compare radar representations for fall detection in a furnished 60 GHz FMCW radar environment. From the same range–Doppler sequence, three input modes were derived and fed to an identi-

cal compact CNN. Temporal spectrogram representations achieved 98–99% test accuracy with balanced precision and recall, while the static time-pooled RDM reached 89.4% with reduced non-fall recall, suggesting that Doppler–time dynamics are more informative than static snapshots for fall discrimination. The train/test split evaluates generalization across positions and orientations of the same motion templates rather than to novel motions.

Chapter 6

Conclusion

6.1 Summary

This thesis investigated privacy-preserving human remote sensing in long-term care environments using low-cost 60 GHz FMCW radar. The central theme throughout the thesis was that many clinically and operationally important indoor states are weakly moving or quasi-static, and therefore difficult to detect with conventional motion-triggered sensing. To address this problem, the thesis combined signal processing, lightweight learning models, and digital-twin simulation to study how much useful room-level awareness can be obtained from practical low-cost radar platforms.

Chapter 2 introduced the background needed for the remainder of the thesis. It reviewed the BGT60TR13C FMCW radar hardware used throughout the work, described the data-collection setups, and summarized the main signal-processing and machine-learning tools employed later, including beamforming, CFAR detection, CNNs, and recurrent models. This chapter established the sensing pipeline and notation used by the experimental chapters.

Chapter 3 focused on two clinically relevant detection problems: semi-static occupancy and post-fall floor occupancy. The first part of the chapter proposed RASSO, a Doppler-domain warping and resampling method that improves sensitivity to weak near-zero Doppler signatures while preserving the structure needed for downstream spatial imaging. The second part studied post-fall floor-occupancy detection in a furnished LTC-like environment, showing that low-cost radar can detect persistent floor presence more reliably when the processing chain is tuned to weak quasi-static returns rather than only to strong motion events.

Chapter 4 shifted from representation formation to image-like radar processing for edge deployment. It first evaluated denoising and adaptive preprocessing techniques for radar spectrograms, then proposed imaging-based algorithms for quality scoring, temporal tracking, and lightweight quasi-static detection directly on Range–Azimuth maps. A main conclusion of this chapter was that, once an informative radar image has been formed, simple image-space detectors can provide competitive detection performance at substantially lower computational cost than more traditional CFAR-style pipelines, which is valuable for low-power embedded platforms such as the Raspberry Pi.

Chapter 5 extended the thesis into digital-twin and sim-to-real methods. Section 5.2.2 developed a calibrated digital-twin framework for radar occupancy estimation, showing that noise-floor-aware calibration can make simulated Range–Doppler data substantially more useful for real-world evaluation. Section 5.3 then used a digital twin as a controlled testbed for fall-related representation analysis, showing that temporally resolved spectrogram-based representations are more informative for fall discrimination than a static pooled Range–Doppler representation in the studied setting. Together, these results demonstrated that simulation can serve not only as a data source, but also as a systematic design tool for probing radar representations before large real-data campaigns.

Overall, the thesis showed that meaningful occupancy and post-fall awareness can be obtained from low-cost FMCW radar systems when signal representation, detector design, and data generation are developed together. The broader conclusion is that privacy-preserving radar sensing for assisted living should not be viewed as a single algorithmic problem, but as an integrated pipeline problem spanning hardware constraints, preprocessing, learning, and deployment-oriented evaluation.

6.2 Discussion

The results presented in this thesis should be interpreted in the context of the sensing environments in which they were developed and evaluated. A substantial part of the experimental work was conducted in the MIRADA room at the Schlegel–UW Research Institute for Aging and in related indoor apartment-style environments. Although the proposed methods consistently showed useful performance under the studied conditions, direct deployment to a different nursing home, hospital room, or residential setting would still require care. Changes in room dimensions, radar mounting height and angle, clutter structure, wall reflectivity, and furniture arrangement can all affect the observed multipath and background statistics, and therefore influence both the signal representations

and the downstream detector behaviour. For this reason, transferability across environments should not be assumed to be automatic. Rather, one of the key conclusions of this thesis is that transferability must be supported by representation design, calibration, and environment-aware validation. This is precisely why Chapter 5 introduced the digital-twin and calibrated sim-to-real framework. The digital twin provides a practical mechanism for studying how radar measurements change across layouts, placements, and propagation conditions before conducting expensive real-world trials. It addresses the engineering question of how methods developed in one room may be adapted, stress-tested, or recalibrated for another.

Another natural question is why this thesis focuses on radar rather than Light Detection and Ranging (LiDAR). The reason is primarily deployment-driven. This thesis is motivated by ambient assisted-living monitoring, where privacy, unobtrusiveness, cost, and robustness under low-light conditions are central. Radar is well aligned with these requirements: it is contactless, non-visual, compact, and available in relatively low-cost hardware. LiDAR, while valuable for geometric and depth reconstruction, is less central to the present thesis here. The thesis's emphasis here is on low-cost privacy-preserving room monitoring rather than explicit scene reconstruction. Nevertheless, if LiDAR cost continues to decrease, it could become a useful complementary modality. Radar–LiDAR fusion may be promising for improving room geometry understanding, multi-person localization, and posture interpretation, while preserving radar's advantages for always-on low-light sensing. Overall, the thesis argues that low-cost radar is already a meaningful and practical modality for assisted-living sensing, but it also recognizes that no single modality is universally optimal. The current work establishes radar-based methods that are privacy-preserving, deployment-oriented, and computationally lightweight, while the digital-twin framework begins to address their transferability across environments. Future work should build on this foundation by studying stronger cross-room robustness, environment-specific calibration, and, where appropriate, complementary sensing modalities such as LiDAR to expand the reliability and applicability of ambient monitoring systems.

6.3 Ongoing Research and Future Work

6.3.1 Minew Radar-Based Occupancy Detection and People Counting

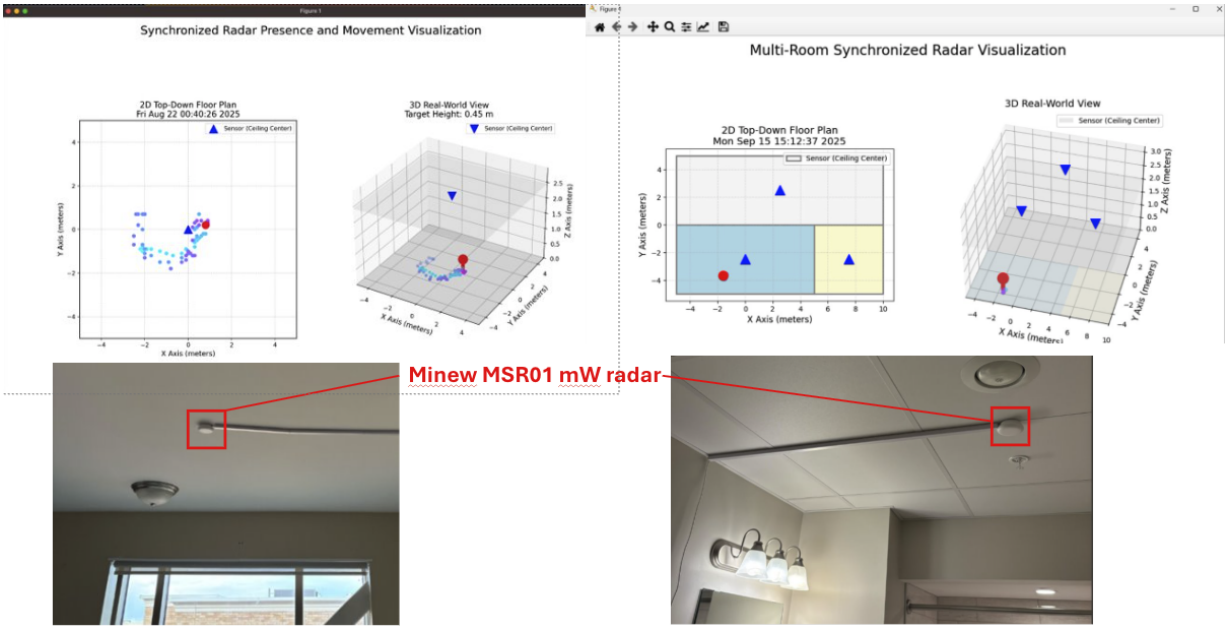

Figure 6.1: Overview of the multi-room Minew radar setup and synchronized visualization prototype. **Top-left**: GUI showing synchronized 2-D and 3-D views for the living room. **Top-right**: GUI view of all monitored rooms in the shared apartment coordinate frame. Bottom: ceiling-mounted Minew MSR01 radar nodes installed in the living room (left) and bathroom (right).

One ongoing research direction extends the room-level radar sensing developed earlier in this thesis toward a practical multi-room deployment using commercially available Minew MSR01 millimetre-wave radar nodes [91]. Three radar units were installed in the living room, bedroom, and bathroom of the MIRADA apartment-like environment, as shown in Figure 6.1. Each radar is configured through the vendor MinewLink mobile application and mounted on the ceiling so that its sensing region covers the corresponding room while minimizing duplicated detections across room boundaries. The broader goal of this direction is to move from single-room occupancy awareness toward synchronized apartment-level visualization, occupancy state tracking, and, ultimately, robust people counting across multiple

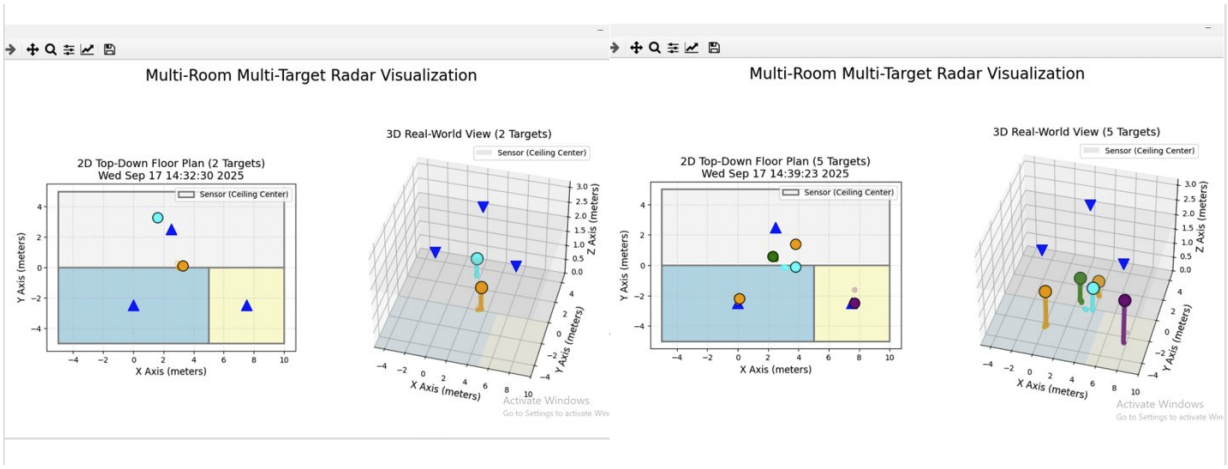

Figure 6.2: Demonstration of the multi-room synchronized radar visualization at Doors Open Waterloo Region 2025. Left: a simpler apartment state with two detected targets. Right: a more complex state with up to five simultaneously visualized targets across three rooms. The 2-D floor-plan view and 3-D room view are generated from the same room-registered radar coordinates.

rooms. The current prototype focuses on the data ingestion and coordinate registration stage. Each Minew radar publishes processed target outputs to the vendor cloud broker using Message Queuing Telemetry Transport (MQTT). In the current implementation, a Python-based visualization program subscribes to the room-specific topics of the form `/MSR01/<MAC>/radar`, where each topic corresponds to a specific sensor identified by its MAC address. The incoming payload is JSON-encoded and includes, for each frame, the reported number of detected targets and their associated 3-D coordinates in the local sensor coordinate system. The current synchronization prototype parses the coordinate string of the dominant reported target in each room, converts it into metric (x, y, z) values, and stores these values together with a short temporal trail for visualization. This design allows all room streams to be synchronized in a single software layer while keeping the vendor-side detection and target extraction unchanged.

To form a unified apartment view, the local coordinates of each radar are transformed into a common global layout. In the current implementation, each room is assigned geometric bounds in a global apartment floor plan, together with a room-specific offset relative to the living room origin. The local Minew coordinate system is then mapped to the global frame by flipping the local x-axis, preserving the local y-axis, and applying the corresponding room offset. The reported vertical coordinate is converted into height-above-floor using

the known ceiling-mounted sensor height. As a result, the same radar report can be shown consistently in both a 2-D top-down floor plan and a 3-D room-level visualization. The software also maintains a short trail of recent detections using a fixed-length queue, allowing the user to inspect motion continuity rather than only the most recent point.

Figures 6.1 and 6.2 summarize the current system. The synchronized Graphical User Interface (GUI) displays the current target positions in 2-D and 3-D, with room outlines, sensor locations, current detections, and short recent motion traces. The blue triangles denote the ceiling-mounted radar positions, while the detected target positions are rendered as room-registered points. In the 3-D view, the transformed vertical coordinate is shown through a point-and-stick representation extending from the floor upward, which gives an immediate visual cue for standing, sitting, or floor-level postures. The present prototype, therefore, provides the system-level infrastructure needed for future apartment-scale monitoring: MQTT subscription, JSON parsing, coordinate transformation, room registration, and synchronized 2-D/3-D visualization. The synchronized visualization shown in Figure 6.2 is generated directly from the room-registered radar coordinates. After decoding the MQTT payload, the software converts each room-local target coordinate into the common apartment frame and renders it simultaneously in two linked views: a 2-D floor plan used for intuitive room-level localization and a 3-D view used to interpret height relative to the ceiling-mounted sensors. In the present interface, each radar node is drawn as a blue triangle at the ceiling center of its assigned room. The latest detected position is emphasized in red in the 2-D plot, while recent motion history is shown by a short colour-coded trail. In the 3-D plot, the same coordinate is displayed with an estimated height-above-floor, allowing rough visual distinction between upright and lower postures.

A second prototype in this research direction moves closer to raw board-level access. Instead of relying only on the vendor coordinate outputs, the radar board is connected through a Transistor-Transistor Logic (TTL) to Universal Serial Bus (USB) serial adapter, as shown in Figure 6.3. A Raspberry Pi 4B is then used as a lightweight edge host to read the incoming serial data stream and visualize the resulting point-cloud style detections using Python. This prototype is intended to provide more direct access to the radar output and to support future embedded deployments without requiring a laptop-based visualization environment. The current implementation focuses on verifying the end-to-end edge workflow: board connection, serial acquisition, parsing of the incoming data stream, and rendering of the output in both 2-D and 3-D. Figure 6.3 shows an example of the resulting point-cloud visualization for a quasi-static human target. The immediate research value of this prototype is that it provides a lower-level interface for future development of point-cloud filtering, target clustering, count estimation, and activity recognition. Future work will complete this prototype with a Raspberry Pi 4B as the main edge controller for

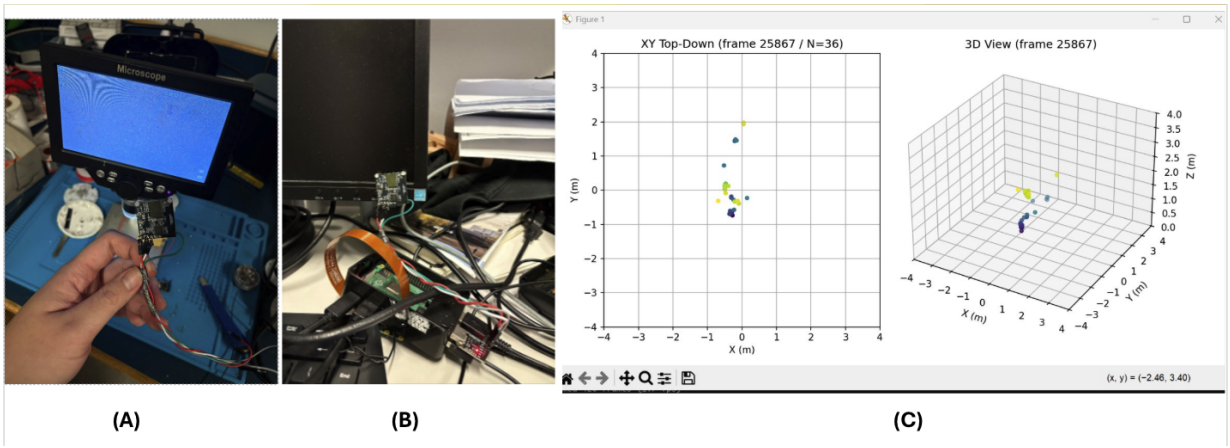

Figure 6.3: Prototype point-cloud visualization workflow for the Minew radar board. (A) Hardware connection through a TTL–USB serial adapter. (B) Prototype data-acquisition setup connected to a Raspberry Pi 4B. (C) Example Python-based 2-D and 3-D visualization of the received point cloud for a quasi-static human target.

the Minew radar board, followed by Printed Circuit Board (PCB) integration and enclosure packaging for more reliable field testing.

6.3.2 Multi-People Close-Interaction Based Localization

This work extends the imaging-based radar localization ideas of Chapter 4 to the more difficult case of multiple people interacting at close range. A practically important motivation is assisted living, where clinically relevant scenarios often involve two or more people in proximity: for example, a caregiver helping a resident stand up, two people sitting next to each other, or a helper leaning toward a resident who is sitting or lying down. In such cases, the radar returns from multiple bodies overlap strongly in angle, range, and multipath structure, making the separation of individuals substantially more difficult than in single-person occupancy detection. These scenarios generate complex radar signatures due to overlapping reflections, non-homogeneous clutter, and target coupling. As a result, both localization and people counting become challenging when subjects are close together or partially occlude one another in the radar field of view. This ongoing work, therefore, asks whether the radar imaging and lightweight image-processing ideas developed earlier in the thesis can be pushed beyond single-target detection toward small-group localization in cluttered indoor scenes. Figure 6.4 summarizes preliminary experiments

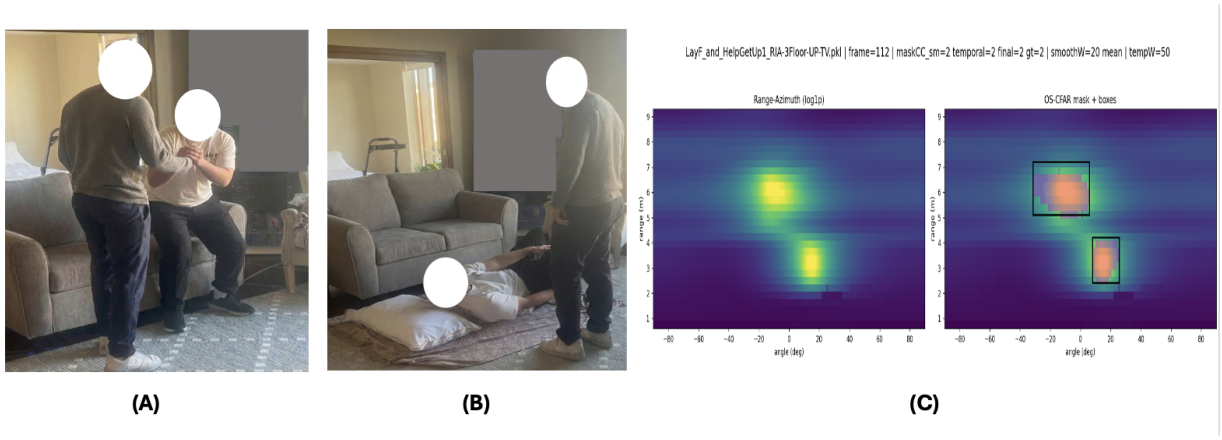

Figure 6.4: Preliminary close-interaction localization experiments. (A) and (B) show two assisted-living interaction scenarios involving two people in proximity. (C) shows the corresponding radar-derived range–angle visualizations used to study whether the two nearby subjects can be separated under overlapping reflections and clutter.

for this direction. Figures 6.4(A) and 6.4(B) show two representative assisted-living interaction scenarios: one person helping another to stand from a seated position and one person assisting another near floor level. The corresponding radar-derived range–angle maps are shown in Figure 6.4(C) and Figure 6.4(D), where the goal is to localize two nearby people despite overlapping returns and partial clutter contamination. The present implementation builds on the designated preprocessing and beamforming stages introduced earlier in the thesis. The signal-processing chain first suppresses static clutter and forms a spatially resolved radar image, after which image-based detection is used to separate the dominant blobs corresponding to nearby people. Preliminary observations suggest that, in these close-interaction cases, classical OS-CFAR alone may under-detect the weaker target because the stronger target and surrounding clutter bias the local threshold. For this reason, the ongoing approach continues the Chapter 4 idea of moving beyond purely pixel-wise thresholding and instead introducing lightweight mask refinement based on connected components and local spatial consistency. At the current stage, this direction should be viewed as a proof-of-feasibility study rather than a complete close-interaction tracking system. The initial experiments indicate that spatially resolved radar maps can reveal multiple nearby occupants, but robust counting and localization remain sensitive to angular separation, clutter leakage, and the relative dominance of the stronger reflector. Future work will therefore focus on stronger blob separation, temporal tracking, and better use of sequential context so that multi-person close-interaction localization can be performed

more reliably in assisted-living settings.

6.3.3 Radar-Based Sim2Real Fall Detectxion

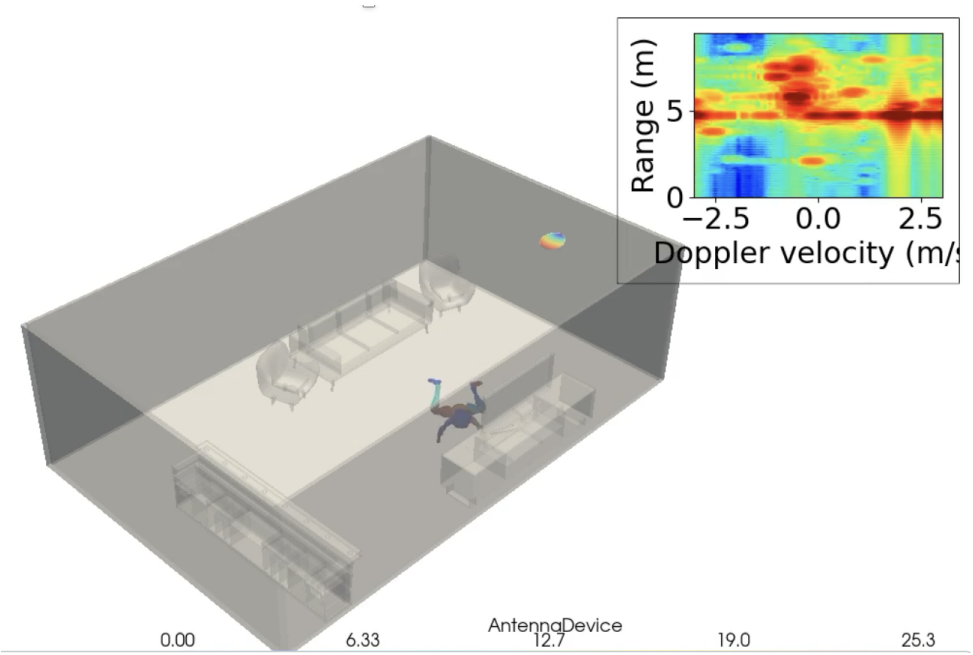

Figure 6.5: Digital-twin setup for a representative fall case. A humanoid model performs a forward fall with bent knees in a living-room scene modelled after the RIA environment, while a wall-mounted virtual radar captures the synthetic radar response.

The final ongoing research direction extends the simulation work of Section 5.2 and the fall-representation study of Section 5.3 toward radar-based sim-to-real fall detection. The motivation is that large, diverse, and clinically representative fall datasets are difficult to acquire safely in the real world. Real falls are rare, dangerous to stage, and ethically difficult to collect at scale. As a result, simulation remains a promising way to generate controlled fall trajectories and to study how representation choice and calibration affect downstream learning.

The current prototype begins with a simplified living-room digital twin modelled after the RIA environment. As shown in Figure 6.5, a humanoid model is simulated to perform a representative fall trajectory, while a virtual radar is placed on the wall at approximately two meters height to generate synthetic radar measurements. From the simulated

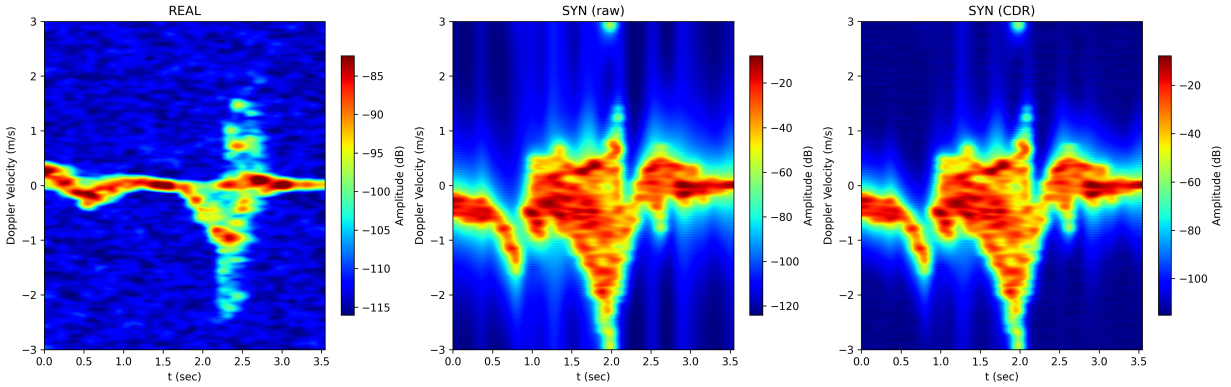

Figure 6.6: Doppler–time spectrograms for a representative fall case in which the subject falls forward with bent knees. Left: measured real data. Middle: raw simulated data from the digital twin. Right: simulated data after calibrated domain randomization (CDR).

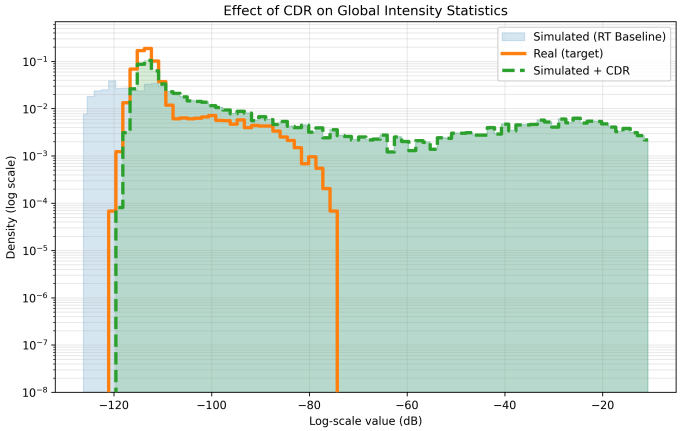

Figure 6.7: Effect of calibrated domain randomization on the global intensity statistics of the simulated fall spectrogram. The histogram compares the baseline ray-tracing simulation, the measured real target data, and the CDR-adjusted simulation in the dB domain.

radar cube, Doppler–time spectrograms are computed using the same type of preprocessing pipeline used for the measured data. This allows a direct qualitative comparison between synthetic and real spectrograms under a matched signal-processing chain. Figure 6.6 shows one representative example for the case of a subject falling forward with the knees bent first. The left panel shows measured data collected from a real experiment, the middle panel shows the raw simulated spectrogram, and the right panel shows the same simulated data after applying CDR from Section 5.2.2. The main purpose of this comparison is not

to claim exact waveform matching, but to determine whether simulation can recover the dominant time–Doppler structure of the fall and whether CDR can move the synthetic background closer to the measured one.

To complement the qualitative spectrogram comparison, Figure 6.7 compares the global intensity statistics of the measured and simulated data. The histogram summarizes how the baseline ray-tracing simulation and the CDR-adjusted simulation distribute intensity values in the dB domain relative to the real target data. Consistent with the occupancy study in Section 5.2, the uncalibrated simulation produces a background that is too clean, whereas the CDR-adjusted simulation shifts the global distribution toward the real data. This does not eliminate all sim-to-real mismatch, but it supports the hypothesis that calibrated background alignment is also relevant in fall-related radar sensing.

At the current stage, this direction remains preliminary and is intended as an extension path rather than a completed fall classifier. What has been demonstrated so far is a feasible simulation-to-measurement comparison pipeline for representative fall signatures, together with a first indication that calibration strategies developed for occupancy sensing may also help in fall-related domains. Future work will expand the motion library, include more fall types and non-fall activities, and evaluate whether these calibrated synthetic datasets can improve downstream fall-classification performance when measured fall data are limited.

6.4 Publications

Below are the publications and research activities relevant to the work in this thesis:

- **Chapter 3** is adapted from:
  - Trinh, H., Creager, E., & Shaker, G. (2026). Doppler-Domain Warping and Resampling for Semi-Static Human Occupancy Detection Using Low-Resolution SIMO FMCW Radar. **(Accepted)**
  - Trinh, H., Thai, P.T., Creager, E., & Shaker, G. (2026). Reliable Quasi-Static Post-Fall Floor-Occupancy Detection Using Low-Cost Millimetre-Wave Radar. **(Accepted)**
- **Chapter 4** is adapted from:
  - Trinh, H., & Shaker, G. (2026). Lightweight Range-Angle Imaging Based Algorithm for Quasi-Static Human Detection on Low-Cost FMCW Radar. **(Accepted)**

- **Chapter 5** is adapted from:
  - Trinh, H., Ratto, S., Creager, E., & Shaker, G. (2026). A Physics-Informed Digital Twin Framework for Calibrated Sim-to-Real FMCW Radar Occupancy Estimation. **(Accepted)**
  - SebastianRatto, V., Trinh, H., Sayed, A.N., Elbadrawy, A., Sligar, A.P., & Shaker, G. (2026). Radar-Based Fall Detection for Assisted Living: A Digital-Twin Representation Case Study. **(Accepted)**

References

[1] World Health Organization. Ageing and health. WHO Newsroom Fact Sheets, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (Accessed: Mar. 18, 2026).

[2] Statistics Canada. Older adults and population aging. Government of Canada Statistics Portal, 2026. [Online]. Available: https://www.statcan.gc.ca/en/subjects-start/older_adults_and_population_aging (Accessed: Mar. 18, 2026).

[3] Vera Winter, Jonas Schreyögg, and Andrea Thiel. Hospital staff shortages: Environmental and organizational determinants and implications for patient satisfaction. *Health Policy*, 124(4):380–388, None 2020.

[4] Public Health Agency of Canada. Falls among older adults in canada. Government of Canada Health Infobase, 2025. [Online]. Available: https://health-infobase.canada.ca/falls-in-older-adults/ (Accessed: Mar. 18, 2026).

[5] Public Health Agency of Canada. Surveillance report on falls among older adults in canada. Surveillance Report HP35-154/2022E-PDF, Public Health Agency of Canada, Ottawa, ON, 2022. Available: https://www.canada.ca/en/public-health/services/publications/healthy-living/surveillance-report-falls-older-adults-canada.html [Accessed: Mar. 18, 2026].

[6] Marianne Sinoo, J. Hoof, and Helianthe Kort. Light conditions in the nursing home. *Gerontechnology*, 11, 06 2012.

[7] Clara Berridge, Jodi Halpern, and Karen Levy. Cameras on beds: The ethics of surveillance in nursing home rooms. *AJOB Empirical Bioethics*, 10(1):55–62, 2019. PMID: 30794112.

[8] Sarah Wilson, Riona Mc Ardle, Clare Tolley, and Sarah Slight. Usability and acceptability of wearable technology in the early detection of dementia. *Alzheimer's & Dementia*, 18, 12 2022.

[9] Ta-Wei Guu, Anna-Katharine Brem, Christopher P. Albertyn, Pooja Kandangwa, Dag Aarsland, and Dominic ffytche. Wrist-worn actigraphy in agitated late-stage dementia patients: A feasibility study on digital inclusion. *Alzheimer's & Dementia*, 20(5):3211–3218, 2024.

[10] Bodunrin Bakare, Moses Ajaegbu, and V Idigo. A comprehensive review of radar system technology. 8:16–23, 08 2022.

[11] Paramveer Singh, Shree Menkudale, Samraddhi Soni, Vanita Tank, and Arockia Raj. Comprehensive survey on radar systems and its target classification techniques. *Journal of Electronics and Electrical Engineering*, 1:234–269, 02 2025.

[12] Giovanni Diraco, Alessandro Leone, and Pietro Siciliano. A radar-based smart sensor for unobtrusive elderly monitoring in ambient assisted living applications. *Biosensors*, 7(4), 2017.

[13] Sonia and Semwal Tushar. *Smart Health Monitoring for Elderly Care in Indoor Environments*, pages 379–399. 2021.

[14] Sven Schellenberger, Kilin Shi, Fabian Michler, Fabian Lurz, Robert Weigel, and Alexander Koelpin. Continuous in-bed monitoring of vital signs using a multi radar setup for freely moving patients. *Sensors*, 20(20), 2020.

[15] Infineon Technologies AG. BGT60TR13C rf microelectromechanical sensor datasheet, 2023. https://www.infineon.com/cms/en/product/sensor/radar-sensors/.

[16] Saverio Trotta, Dave Weber, Reinhard W. Jungmaier, Ashutosh Baheti, Jaime Lien, Dennis Noppeney, Maryam Tabesh, Christoph Rumpler, Michael Aichner, Siegfried Albel, Jagjit Singh Bal, and Ivan Poupyrev. 2.3 soli: A tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 42–44, 2021.

[17] Infineon Technologies AG. Digital beamforming using the demo BGT60TR13C radar sensor. Application Note AN155322, Infineon Technologies AG, 2023. Rev. per vendor site. Available: https://www.infineon.com/assets/row/public/documents/24/42/infineon-an155322-digital-beamforming-using-the-demo-bgt60tr13c-radar-sensor-applicationnotes-en.pdf.

[18] Hajar Abedi, Ahmad Ansariyan, and George Shaker. Contactless in-bed detection using a low-cost low-resolution radar. In *2024 IEEE SENSORS*, pages 1–4, 2024.

[19] Joshua Visser, Hajar Abedi, Sachin Karmani, and George Shaker. Ai based activity monitoring in washrooms using low-resolution radar. In *2025 IEEE MTT-S Latin America Microwave Conference (LAMC)*, pages 78–81, 2025.

[20] Hajar Abedi, Shenghang Luo, and George Shaker. On the use of low-cost radars and machine learning for in-vehicle passenger monitoring. *2020 IEEE 20th Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF)*, pages 63–65, 2020.

[21] Christoph Will, Prachi Vaishnav, Abhiram Chakraborty, and Avik Santra. Human target detection, tracking, and classification using 24-ghz fmcw radar. *IEEE Sensors Journal*, 19(17):7283–7299, 2019.

[22] Infineon Technologies AG. Ceiling-mounted occupancy detection using xensiv™ demo BGT60TR13C 60 ghz radar. Application Note AN141319 AN141319, Infineon, Feb 2024. Available: https://www.infineon.com/assets/row/public/documents/24/42/infineon-an141319-ceiling-mounted-occupancy-detection-using-xensiv-demo-bgt60tr13c-60-ghz-radar-applicationnotes-en.pdf.

[23] Hajar Abedi, Ahmad Ansariyan, Plinio P. Morita, Alexander Wong, Jennifer Boger, and George Shaker. Ai-powered noncontact in-home gait monitoring and activity recognition system based on mm-wave fmcw radar and cloud computing. *IEEE Internet of Things Journal*, 10(11):9465–9481, 2023.

[24] Ali Samimi Fard, Mohammadreza Mashhadigholamali, Samaneh Zolfaghari, Hajar Abedi, Mainak Chakraborty, Sachin Karmani, Luigi Borzì, Masoud Daneshtalab, and George Shaker. Fall detection in ambient-assisted living environments using fmcw radars and deep learning. In *2025 IEEE International Radar Conference (RADAR)*, pages 1–6, 2025.

[25] Minh Q. Nguyen, Reinhard Feger, Thomas Wagner, and Andreas Stelzer. Analysis of 2d ca-cfar for ddma fmcw mimo radar. In *2023 20th European Radar Conference (EuRAD)*, pages 423–426, 2023.

[26] Robert Miller. Fundamentals of radar signal processing (richards, m.a.; 2005) [book review]. *IEEE Signal Processing Magazine*, 26:100–101, 2009.

[27] Hermann Rohling. Radar cfar thresholding in clutter and multiple target situations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(4):608–621, 1983.

[28] Ibomoiye Domor Mienye and Theo G. Swart. A comprehensive review of deep learning: Architectures, recent advances, and applications. *Information*, 15(12), 2024.

[29] Dimple Patil, Nitin Rane, Pravin Desai, and Jayesh Rane. *Machine learning and deep learning: Methods, techniques, applications, challenges, and future research opportunities*, pages 28–81. 10 2024.

[30] Maria Trigka and Elias Dritsas. A comprehensive survey of deep learning approaches in image processing. *Sensors*, 25(2), 2025.

[31] Ambuj Mehrish, Navonil Majumder, Rishabh Bharadwaj, Rada Mihalcea, and Soujanya Poria. A review of deep learning techniques for speech processing. *Inf. Fusion*, 99(C), November 2023.

[32] Pavel Davidson, Huy Trinh, Sakari Vekki, and Philipp Müller. Surrogate modelling for oxygen uptake prediction using lstm neural network. *Sensors*, 23(4), 2023.

[33] Philipp Müller, Khoa Pham-Dinh, Huy Trinh, Anton Rauhameri, and Neil J. Cronin. Estimating intra- and inter-subject oxygen consumption in outdoor human gait using multiple neural network approaches. *PLOS ONE*, 19(9):1–19, 09 2024.

[34] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 04 2019.

[35] Jianfei Yang, Xinyan Chen, Dazhuo Wang, Han Zou, Chris Lu, Sumei Sun, and Lihua Xie. Deep learning and its applications to wifi human sensing: A benchmark and a tutorial, 07 2022.

[36] Avik Santra, Souvik Hazra, Lorenzo Servadei, Thomas Stadelmayer, Michael Stephan, and Anand Dubey. *Methods and Techniques in Deep Learning: Advancements in mmWave Radar Solutions*. 11 2022.

[37] Abdurrahman Isbitirici, Paolo Falcone, Laura Giarre, and Wen Xu. LSTM-based virtual load sensor for heavy-duty vehicles, 11 2023.

[38] Ravi Raj and Andrzej Kos. An extensive study of convolutional neural networks: Applications in computer vision for improved robotics perceptions. *Sensors*, 25(4), 2025.

[39] Vijaya Nair. Batch norm explained visually — how it works, and why neural networks need it. Towards Data Science, May 2021. https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739.

[40] Ibomoiye Domor Mienye, Theo G. Swart, and George Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9), 2024.

[41] Hao Kong, Cheng Huang, Jiadi Yu, and Xuemin Shen. A survey of mmwave radar-based sensing in autonomous vehicles, smart homes and industry. *IEEE Communications Surveys & Tutorials*, 27(1):463–508, 2025.

[42] Hajar Abedi, Ahmad Ansariyan, Plinio P Morita, Alexander Wong, Jennifer Boger, and George Shaker. Radar-based in-home monitoring system for supporting aging and wellness. In *2024 IEEE Microwaves, Antennas, and Propagation Conference (MAPCON)*, pages 1–4, 2024.

[43] Kun Yan, Shiyou Wu, and Guangyou Fang. Detection of quasi-static trapped human being using mono-static uwb life-detection radar. *Applied Sciences*, 11(7), 2021.

[44] Hongchun Li, Lili Xie, Qian Zhao, Jun Tian, and Takeshi Konno. Static human localization using fmcw mimo radar. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2023.

[45] Avik Santra and Souvik Hazra. *Deep Learning Applications of Short-Range Radars*. 09 2020.

[46] Chenyan Song, Ehsan Yavari, Xiaomeng Gao, Victor M. Lubecke, and Olga Boric-Lubecke. Respiration signal pattern analysis for doppler radar sensor with passive node and its application in occupancy sensing of a stationary subject. *Biosensors*, 15(5), 2025.

[47] Yihong Luo and Xiaoxia Li. Indoor human location method for fmcw radar using standard deviation weighting. In *2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*, pages 159–163, 2022.

[48] Kevin Kaiser, Thomas Stadelmayer, Jens Felgentreff, Robert Weigel, Fabian Lurz, and Avik Santra. Improved indoor semi-static human target detection and localization

using fmcw radar. In *2023 20th European Radar Conference (EuRAD)*, pages 298–301, 2023.

[49] Sabri Mustafa Kahya, Muhammet Sami Yavuz, and Eckehard Steinbach. Mcrood: Multi-class radar out-of-distribution detection, 2023.

[50] Stirling Scholes, Alice Ruget, Feng Zhu, and Jonathan Leach. Human pose inference using an elevated mmwave fmcw radar. *IEEE Access*, 12:115605–115614, 2024.

[51] Mareike Wendelmuth, Alexander Yarovoy, and Francesco Fioranelli. Breathing rate estimation & apnea detection with multiple elevated and tilted fmcw radars. In *2025 IEEE Radar Conference (RadarConf25)*, pages 1022–1027, 2025.

[52] S. Umesh, L. Cohen, and D. Nelson. Frequency warping and the mel scale. *IEEE Signal Processing Letters*, 9(3):104–107, 2002.

[53] Sabri Mustafa Kahya, Muhammet Sami Yavuz, and Eckehard Steinbach. Hood: Real-time human presence and out-of-distribution detection using fmcw radar. *IEEE Transactions on Radar Systems*, 3:44–56, 2025.

[54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[55] Knox Community Hospital. Pressure ulcers. Knox Community Hospital Health Library, 2026. [Online]. Available: https://www.kch.org/pressure-ulcers (Accessed: Mar. 17, 2026).

[56] Miguel Terroso, Natacha Rosa, António Marques, and Ricardo Simoes. Physical consequences of falls in the elderly: A literature review from 1995 to 2010. *European Review of Aging and Physical Activity*, 11:51–59, 04 2013.

[57] Kimberly Charlton, Carolyn M. Murray, and Saravana Kumar. Perspectives of older people about contingency planning for falls in the community: A qualitative meta-synthesis. *PLOS ONE*, 12(5):1–14, 05 2017.

[58] Kylie Johnston, Anthea Worley, Karen Grimmer-Somers, Michele Sutherland, and Lorraine Amos. Personal alarm use to call the ambulance after a fall in older people: Characteristics of clients and falls. *Australasian Journal of Paramedicine*, 8:1–9, 2010.

[59] SAMUEL R. NYMAN and CHRISTINA R. VICTOR. Use of personal call alarms among community-dwelling older people. *Ageing and Society*, 34(1):67–89, 2014.

[60] Jenny Kubitza, Iris Tamara Schneider, and Bernd Reuschenbach. Concept of the term long lie: a scoping review. *European Review of Aging and Physical Activity*, 20, 2023.

[61] Anuradha Singh, Saeed Ur Rehman, Sira Yongchareon, and Peter Han Joo Chong. Sensor technologies for fall detection systems: A review. *IEEE Sensors Journal*, 20(13):6889–6919, 2020.

[62] Omar Flor-Unda, Rafael Arcos-Reina, Cristina Estrella-Caicedo, Carlos Toapanta, Freddy Villao, Héctor Palacios-Cabrera, Susana Nunez-Nagy, and Bernardo Alarcos. A comparative overview of technological advances in fall detection systems for elderly people. *Sensors*, 25(24), 2025.

[63] Hajar Abedi, Ahmad Ansariyan, Plinio P Morita, Jennifer Boger, Alexander Wong, and George Shaker. Sequential deep learning for in-home activity monitoring using mm-wave fmcw radar. In *2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (APS/URSI)*, pages 1499–1500, 2021.

[64] JIA-RONG YEH, JIANN-SHING SHIEH, and NORDEN E. HUANG. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in Adaptive Data Analysis*, 02(02):135–156, 2010.

[65] Zhenghui Li, Julien Le Kernec, Qammer Hussain Abbasi, Francesco Fioranelli, Shufan Yang, and Olivier Romain. Radar-based human activity recognition with adaptive thresholding towards resource constrained platforms. *Scientific Reports*, 13, 2023.

[66] S. Butterworth. On the Theory of Filter Amplifiers. *Experimental Wireless & the Wireless Engineer*, 7:536–541, October 1930.

[67] NgocBinh Nguyen, MinhNghia Pham, Van-Sang Doan, and VanNhu Le. Improving human activity classification based on micro-doppler signatures of fmcw radar with the effect of noise. *PLOS ONE*, 19(8):1–26, 08 2024.

[68] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[69] Yuan He, Xinyu Li, Runlong Li, Jianping Wang, and Xiaojun Jing. A deep-learning method for radar micro-doppler spectrogram restoration. *Sensors*, 20(17), 2020.

[70] Ibrahim Alnujaim, Shobha Sundar Ram, Daegun Oh, and Youngwook Kim. Synthesis of micro-doppler signatures of human activities from different aspect angles using generative adversarial networks. *IEEE Access*, 9:46422–46429, 2021.

[71] Raspberry Pi Ltd. Raspberry pi 4 model b. *Raspberry Pi Product Page*, 2019. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (Accessed: April 15, 2025).

[72] Sebastian Ratto, Ahmed N. Sayed, Abdelrahman Elbadrawy, Arien Sligar, Omar Ramahi, and George Shaker. Digital twin-driven radar for human monitoring optimization. In Lucio Tommaso De Paolis, Pasquale Arpaia, and Marco Sacco, editors, *Extended Reality*, pages 33–45, Cham, 2026. Springer Nature Switzerland.

[73] Sebastian Ratto V, Ahmed N. Sayed, Abdelrahman Elbadrawy, Arien Sligar, Omar M. Ramahi, and George Shaker. Digital twin system for realistic radar optimization in healthcare monitoring. In *2025 IEEE 20th International Symposium on Antenna Technology and Applied Electromagnetics (ANTEM)*, pages 182–185, 2025.

[74] Travis D. Bufler, Ram M. Narayanan, and Traian Dogaru. Radar signatures of indoor clutter for through-the-wall radar applications. In *Defense + Security Symposium*, 2014.

[75] Zhanjun Hao, Hao Yan, Xiaochao Dang, Zhongyu Ma, Peng Jin, and Wenze Ke. Millimeter-wave radar localization using indoor multipath effect. *Sensors*, 22(15), 2022.

[76] Vasileios-Rafail Xefteris, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris. Performance, challenges, and limitations in multimodal fall detection systems: A review. *IEEE Sensors Journal*, 21(17):18398–18409, 2021.

[77] Minakshi Debnath, Sana Alamgeer, Md Shahriar Kabir, and Anne H. Ngu. Enhancing wearable fall detection system via synthetic data. *Sensors*, 25(15), 2025.

[78] V. Sebastian Ratto, Ahmed N. Sayed, Arien Sligar, Omar M. Ramahi, and George Shaker. Radar-based people presence estimation using digital twin simulation framework. In *2025 International Telecommunications Conference (ITC-Egypt)*, pages 164–166, 2025.

[79] Adobe. Mixamo: Characters and animations, 2021. Available: https://www.mixamo.com/, Accessed: Aug. 19, 2025.

[80] CMU Graphics Lab Motion Capture Database. CMU motion capture database. http://mocap.cs.cmu.edu/, 2003–2007.

[81] Huy Trinh, Elliot Creager, and George Shaker. Doppler-domain respiratory amplification for semi-static human occupancy detection using low-resolution simo fmcw radar, 2026.

[82] Joshua Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[83] Fereshteh Sadeghi and Sergey Levine. CAD2RL: real single-image flight without a single real image. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[84] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, V. Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018.

[85] U. Chipengo, A. P. Sligar, S. M. Canta, et al. High-fidelity physics simulation-based cnn for automotive radar target classification using micro-doppler. *IEEE Access*, 9:82597–82610, 2021.

[86] Sebastian Ratto, Ahmed N. Sayed, Arien Sligar, Omar M. Ramahi, and George Shaker. Radar-based people presence estimation using digital twin simulation framework. In *2025 International Telecommunications Conference (ITC-Egypt), Cairo, Egypt*, pages 164–166, 2025.

[87] Sebastian Ratto V., Ahmed N. Sayed, Abdelrahman Elbadrawy, Arien Sligar, Omar M. Ramahi, and George Shaker. Enhancing healthcare monitoring through a fully integrated radar digital twin. In *2025 International Applied Computational Electromagnetics Society Symposium (ACES), Florida, USA*, pages 1–1, 2025.

[88] Ahmed N. Sayed, Omar M. Ramahi, and George Shaker. Frequency-modulated continuous-wave radar perspectives on unmanned aerial vehicle detection and classification: A primer for researchers with comprehensive machine learning review and

emphasis on full-wave electromagnetic computer-aided design tools. *Drones*, 8(8), Aug. 2024.

[89] Sebastian Ratto V, Ahmed N. Sayed, Abdelrahman Elbadrawy, Arien Sligar, Omar M. Ramahi, and George Shaker. Digital twin system for realistic radar optimization in healthcare monitoring. In *2025 IEEE 20th International Symposium on Antenna Technology and Applied Electromagnetics (ANTEM), St. John's, NL, Canada*, pages 182–185, 2025.

[90] Synopsys, Inc. Perceive–EM. [Online]. Available: https://www.synopsys.com/verification/virtual- GPU-accelerated solver for radar and wireless digital-twin simulation.

[91] Shenzhen Minew Technologies Co., Ltd. *MSR01 Millimeter Wave Radar Sensor User Manual*. Shenzhen Minew Technologies Co., Ltd., Shenzhen, China, 2023. Rev. 1.0. Available: https://www.minew.com/product/msr01-millimeter-wave-radar-sensor/.

[92] Ehsan Yavari, Chenyan Song, Victor Lubecke, and Olga Boric-Lubecke. Is there anybody in there?: Intelligent radar occupancy sensors. *IEEE Microwave Magazine*, 15(2):57–64, 2014.