Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Lehrstuhl für Messsystem- und Sensortechnik

Prof. Dr.-Ing. habil. Dr. h.c. Alexander W. Koch

Master´s Thesis

Advanced AI Techniques for Interference Reduction in Automotive Radar Systems

Congyan Wang

April 22nd, 2025

Supervisor:
Jürgen Schäfer, Wolfgang Kurz, Martin Jakobi

Master’s Thesis at
Lehrstuhl für Messsystem und Sensortechnik (MST)
Title: Advanced AI Techniques for Interference Reduction in Automotive Radar Systems
Author: Congyan Wang

Congyan Wang
Neufahrner Straße 20
85748 Garching b. München
congyan.wang@tum.de

Abstract

The explosion of automotive radar systems in advanced driver-assistance systems and autonomous driving has led to increasing mutual interference challenges that compromise detection reliability and safety-critical functions. While traditional signal processing methods show limitations in complex, dynamic interference environments, recent deep learning approaches have demonstrated promise but often focus on single-domain processing, neglecting complementary information.

This research proposes a novel dual-path autoencoder architecture for automotive radar interference mitigation that simultaneously processes raw time-domain radar signals and their frequency-domain range-Doppler representations. By leveraging multi-modal information fusion, the proposed approach captures both the fine-grained temporal characteristics critical for phase preservation and the intuitive spatial features essential for target detection.

Initial experiments with various autoencoder architectures revealed significant challenges in preserving phase information and maintaining target resolution in range-Doppler maps. The analysis demonstrates that conventional convolutional networks with excessive pooling and batch normalization can degrade critical signal characteristics, while transposed convolutions introduce undesirable artifacts. Comprehensive performance evaluation using both traditional metrics and radar-specific indicators provides insights into optimization strategies for preserving both amplitude and phase fidelity.

The proposed multi-modal approach addresses the limitations of single-domain processing by enhancing the signal interference ratio while maintaining target separability and reducing false detections in common traffic scenarios. This research contributes to the exploration of deep learning applications in radar signal processing and suggests a potential direction for enhancing interference mitigation in automotive radar systems through multi-domain processing approaches.

Table of Contents

1. Introduction

1.1 Motivation

Currently, automotive radar has become one of the key sensors for environmental perception and safe driving. Detecting targets, identifying obstacles, and tracking cars depend on modern automotive radars—especially those employing millimeter-wave technology with frequency modulated continuous wave (FMCW) and chirp sequence techniques—as they are very dependable, work well in all weather, and provide great detail in distance and speed. However, the increasing number of on-board radars has led to a serious interference problem between multiple radar systems. The interference phenomenon not only decreases the amplitude of the target signal but also may destroy the signal phase, thereby affecting the accurate estimation of distance and speed, ultimately restricting the performance and safety of automotive driving systems.

1.2 Research Challenges

Ongoing studies regarding radar interference problems mainly focus on traditional signal processing methods and deep learning-based approaches. Traditional methods include filtering, time-frequency analysis, and adaptive signal processing. Although they can alleviate interference to a certain extent, they generally have limitations such as insufficient adaptability to dynamic and complex interference, loss of signal details, and poor real-time performance. In recent years, deep learning methods have made remarkable leaps in interference suppression with their end-to-end learning and powerful feature extraction capabilities. However, most existing deep learning solutions use single-mode input, such as processing only in the time domain or frequency domain, which makes it difficult to fully utilize the multi-dimensional information contained in radar signals.

1.3 Research Objectives

By investigating innovative applications of deep learning methods for interference reduction, this thesis tackles the increasing problem of radar interference in contemporary automotive settings. The inspiration comes from the seen shortcomings of conventional suppression techniques, which more and more battle with the complexity of real-world traffic situations with many radar-equipped cars running concurrently.

This study's main goal is to create a strong radar signal processing system that can precisely identify real objects while efficiently removing false replies brought on by noise and interference. The work uses realistic car radar datasets impacted by different kinds of interference to this goal, hoping to extract necessary target characteristics even under compromised signal conditions.

The study initially investigates the basic constraints of current interference suppression methods, especially their difficulties in keeping phase coherence and sustaining target resolution in order to reach this objective. The study suggests an alternate method that combines complimentary information from several signal representations by means of these limitations, hence improving the model's capacity to separate useful signal components from interference.

This work's foundation is the creation and assessment of a dual-path autoencoder architecture that concurrently processes radar signal representations in time-domain and frequency-domain. While being computationally efficient and appropriate for deployment on resource-constrained automotive platforms, this multi-domain technique is meant to show better resilience to interference than single-domain strategies.

Apart from architectural growth, this dissertation underlines the creation of thorough assessment procedures customized for radar interference reduction systems. These comprise not only conventional signal reconstruction criteria but also radar-specific performance measures including false alarm rate and probability of detection. Such evaluation criteria provide further understanding of the practical possibilities of the suggested approaches and allow significant comparisons with current procedures.

This study adds to the changing field of radar signal processing by leveraging deep learning in a focused, performance-conscious way, hence solving the growing electromagnetic complexity of automobile sensing environments. The results of this study provide a strong basis for future enhancements in the perception reliability of car radar systems working under actual interference situations.

1.4 Scope of work

This research focuses on developing and evaluating deep learning approaches for automotive radar interference mitigation, with particular emphasis on the dual-path autoencoder architecture. The scope encompasses several key areas while necessarily excluding others to maintain focus.

The work consists of the simulation of realistic automotive radar scenarios with several interference types, the design and implementation of neural network architectures for interference suppression, and a detailed assessment of these approaches against conventional methods. The research specifically addresses FMCW radar systems operating in highway environments, as these represent the most common deployment scenario for current automotive radar technology.

The scope of this research is bounded in several important ways. While the developed methods are evaluated on simulated data that closely mirrors real-world conditions, testing on physical radar hardware in actual driving environments remains outside the current scope. Furthermore, this study highlights interference between automotive radars over handling all possible sources of electromagnetic disturbance. The implementation considerations are limited to current automotive computing platforms, though detailed hardware optimization is left for future work.

The thesis is organized as follows: Chapter 2 presents the theoretical background of automotive radar systems and a comprehensive literature review. Chapter 3 details the methodology and model development process, including design iterations and optimization strategies. Chapter 4 presents experimental results and performance analysis. Finally, Chapter 5 concludes the thesis and suggests directions for future work.

2. Technical Background and Related Work

This chapter describes the fundamental theoretical information, some related work regarding current interference mitigation technologies and equipment required for development, and the experiment. The development process was conducted on MATLAB and Python IDE.

2.1 Fundamentals of Automotive Radar Systems

Figure 2.1.1 illustrates the fundamental signal processing chain of an automotive FMCW radar system operating in a complex electromagnetic environment. The diagram captures the essence of radar interference challenges in modern traffic scenarios.

The transmitter (TX) in this system transmits chirp signals (blue waves) that bounce off target objects back to the receiver (RX). Simultaneously, interference signals from nearby radar systems (orange waves) may enter the same receiver. This creates a challenging signal environment where desired reflections and unwanted interference are superimposed.

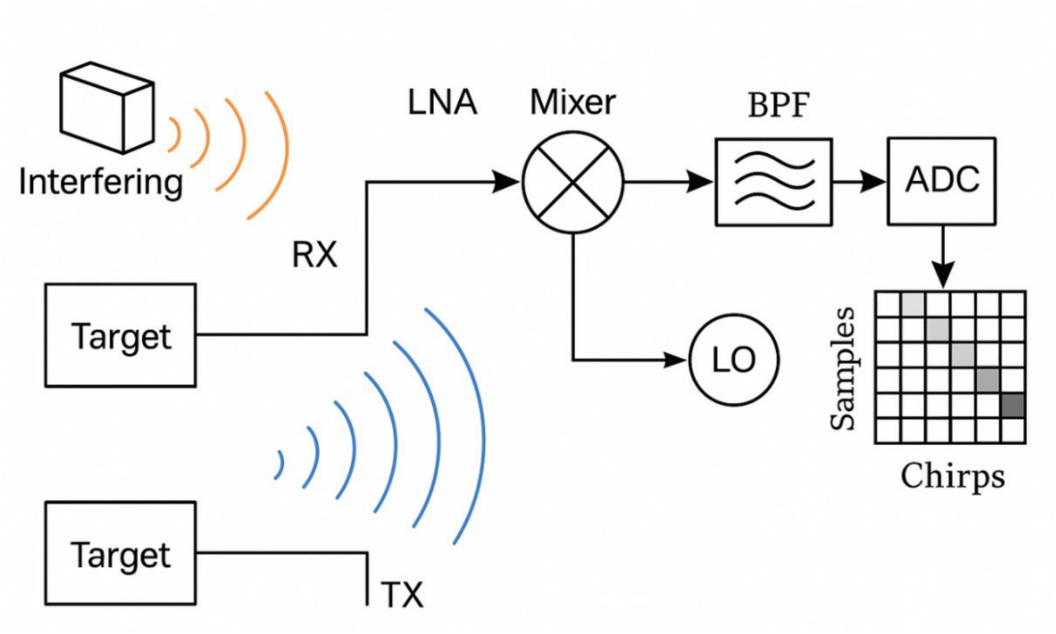

Figure 2.1.1. Radar Signal Processing Process

The mixed signal obtained goes through several processing stages. First, a low-noise amplifier (LNA) increases the signal while reducing the introduction of supplementary noise. The amplified signal is then combined with the reference signal from the local oscillator (LO) to create an intermediate frequency (IF) signal encoding target information in the frequency components. By means of this vital mixing technique, the FMCW radar can extract range information from frequency differences.

A bandpass filter (BPF) on the resulting signal removes out-of-band noise and restricts the bandwidth to the frequency range of interest. Finally, the analog-to-digital converter (ADC) samples the filtered signal, organizing the resulting two-dimensional data matrix with rows representing samples inside a chirp (fast-time) and columns showing successive chirps (slow-time).

This processing chain forms the basis of later signal analysis in vehicle radar systems. Noise, however, greatly decreases the quality of these data, therefore generating false alarms or detection failures. The proposed dual-path autoencoder architecture of this thesis solves this problem by simultaneously controlling the interfering signals in both temporal and frequency domains.

FMCW radar operates by transmitting a continuous signal whose frequency changes over time, typically in a linear pattern. The transmitted signal can be mathematically expressed as

$$s_t(t) = A_t \exp\left(j2\pi\left(f_0 t + \frac{1}{2}\alpha t^2\right)\right) \qquad \text{(Eq. 1)}$$

where A_t is the signal amplitude, f_0 is the starting frequency, and α is the chirp rate (or frequency slope). When this signal reflects off a target at range R and returns to the radar, the receiving antenna collects the reflected signal:

$$s_r(t) = A_r \exp\left(j2\pi\left(f_0(t-\tau) + \frac{1}{2}\alpha(t-\tau)^2\right)\right) \qquad \text{(Eq. 2)}$$

where $\tau = \frac{2R}{c}$ represents the round-trip time delay, with c being the speed of light.

The radar system then mixes the received signal with the currently transmitted signal to obtain an intermediate frequency (IF) signal and continues to subject it to operations such as low-pass filtering and down-conversion. This process results in a beat signal, which is a key part of the FMCW radar processing. The beat signal can be expressed as

$$s_b(t) = A_b \exp\left(j2\pi\left(f_0\tau + \alpha\tau t - \frac{1}{2}\alpha\tau^2\right)\right) \qquad \text{(Eq. 3)}$$

This can be simplified to:

$$s_b(t) = A_b \exp\left(j2\pi(f_{beat}t + \phi_0)\right) \qquad \text{(Eq. 4)}$$

Where $f_{beat} = \alpha\tau = \frac{2\alpha R}{c}$ is the beat frequency proportional to the target range, and Φ_0 is the initial phase.

The beat signal contains information about the target's range and velocity. The complex nature of this signal (with real and imaginary components) enables the extraction of both range and Doppler information through further processing.

To extract range information, a Fast Fourier Transform (FFT) is applied to the IF signal:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{kn}{N}} \quad \text{(Eq. 5)}$$

where x[n] represents the sampled beat signal and N is the number of samples.

For velocity (Doppler shift) information, multiple chirps are processed. The 2D FFT for range-Doppler (RD) processing is given by:

$$X[k,l] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x[n,m]e^{-j2\pi\left(\frac{kn}{N}+\frac{lm}{M}\right)} \quad \text{(Eq. 6)}$$

where M is the number of chirps, $x[n,m]$ is the sampled beat signal for the m-th chirp at the n-th range bin.

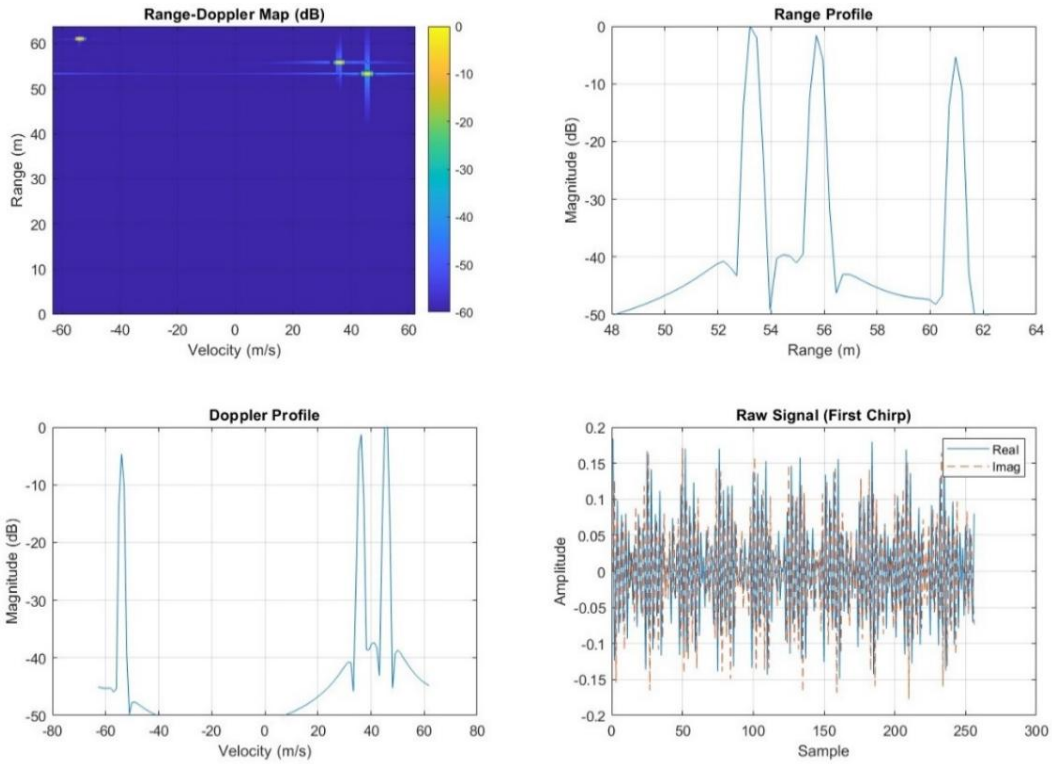

Figure 2.1.2. Characteristics of Original (Clean) Signal

The result is an RD map as shown in Figure 2.1.2, which displays targets in the range-velocity domain. The peaks in this map correspond to detected targets. Under clean circumstances, these peaks are very obvious against the background noise level. On the other hand, other radar systems' interference can greatly distort this image and complicate target identification.

2.2 Problem Formulation

In automotive radar systems based on FMCW technologies, detection accuracy is essential. Target detection in these systems is based on the transmitted chirp signal and its matching received beat signal; by extracting the frequency components, one can estimate the range and relative velocity. However, as the density of radar-equipped vehicles increases, mutual interference between radar sensors becomes a key challenge.

The beat signal of FMCW radar is essentially a superposition of the target reflection and unwanted interference signals and noise. The target component is usually characterized by different sinusoidal waves corresponding to specific ranges and Doppler shifts. In contrast, interference signals from other vehicles or external electromagnetic sources exhibit a wider frequency distribution and usually distort the amplitude and phase information of the beat signal [1].

This interference leads to an increase in the noise floor in the RD diagram, which directly reduces the accuracy of target detection and parameter estimation. Therefore, the presence of interference poses several technical challenges:

I. Interference not only masks the amplitude of the target but also may destroy the phase information, which is important for accurate Doppler shift estimation and determining the target's velocity.

II. Interference can be non-stationary and vary over time and in different frequency bins, making it difficult for traditional linear filtering methods to isolate and remove interference without affecting the target signal.

III. Traditional signal processing techniques (such as FIR filtering, adaptive filtering, and zeroing methods) often fail to handle the complexity and variability of interference patterns in dynamic driving environments [1].

2.2.1 Types of Interference

In dense traffic environments with multiple radar systems operating simultaneously, the following types of interference can occur:

I. Direct Interference: This happens when radar signals from another vehicle's transmitter directly enter the victim radar's receiver. Strong, focused interference patterns are produced by the reflected direct path of the interfering signal. Direct interference appears as vertical stripe-like patterns across the range dimension in the RD map, as shown in Figure 2.2.1, significantly raising the noise floor. The corresponding range profile (top right) indicates that although targets stay visible as separate peaks, interference increases the general noise floor. The Doppler Profile (bottom left) confirms this more strongly by showing several spurious peaks at different speeds. With clear changes in both real and imaginary components, the time-domain representation (bottom right) shows phase distortion in the raw signal.

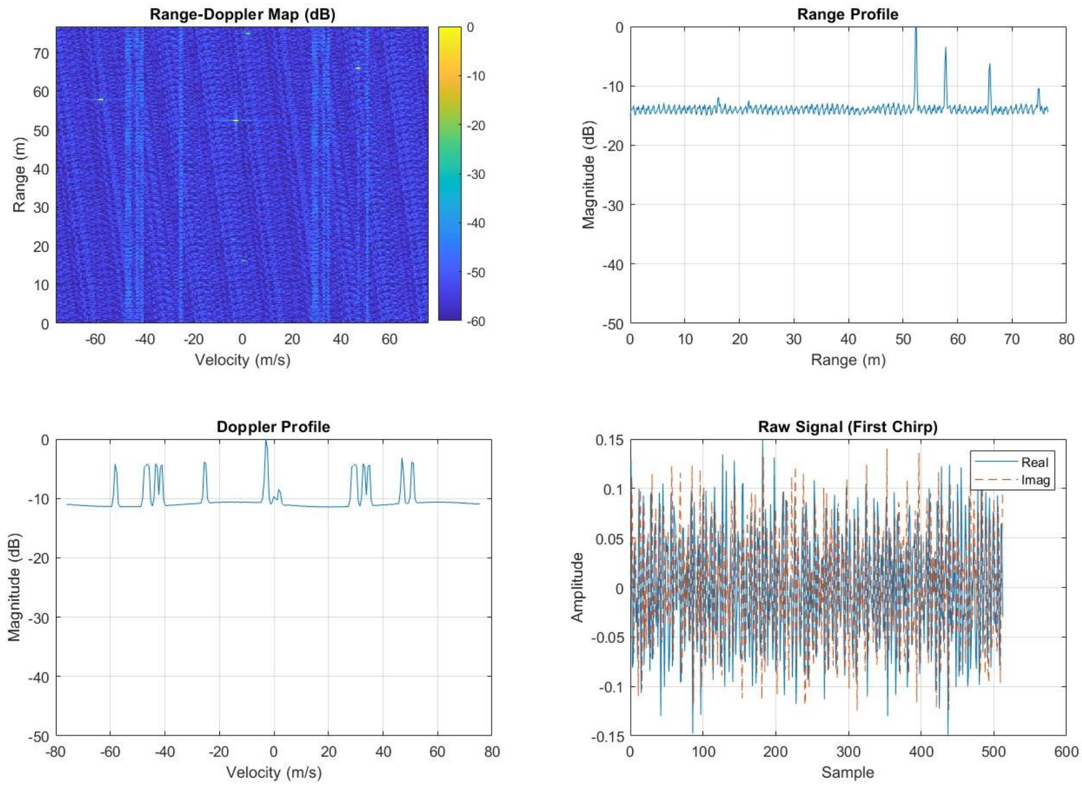

Figure 2.2.1. Characteristics of Different Interference Sources Affecting the Radar Signal in the Time and Frequency Domain

II. Multipath Interference: This kind of interference happens when the radar signal follows several routes to the receiver because of reflections from different objects in the environment (road surface, buildings, other vehicles). Arriving at the receiver with varying time delays and phase shifts, these reflections distort the RD plot. Unlike direct interference, which creates expected patterns, multipath interference creates false targets (ghost targets) at random locations in the frequency domain. These ghost targets might be particularly problematic since they seem like valid goals but are actually non-existent items. Such behavior in multi-antenna systems can result in incorrect target localization.

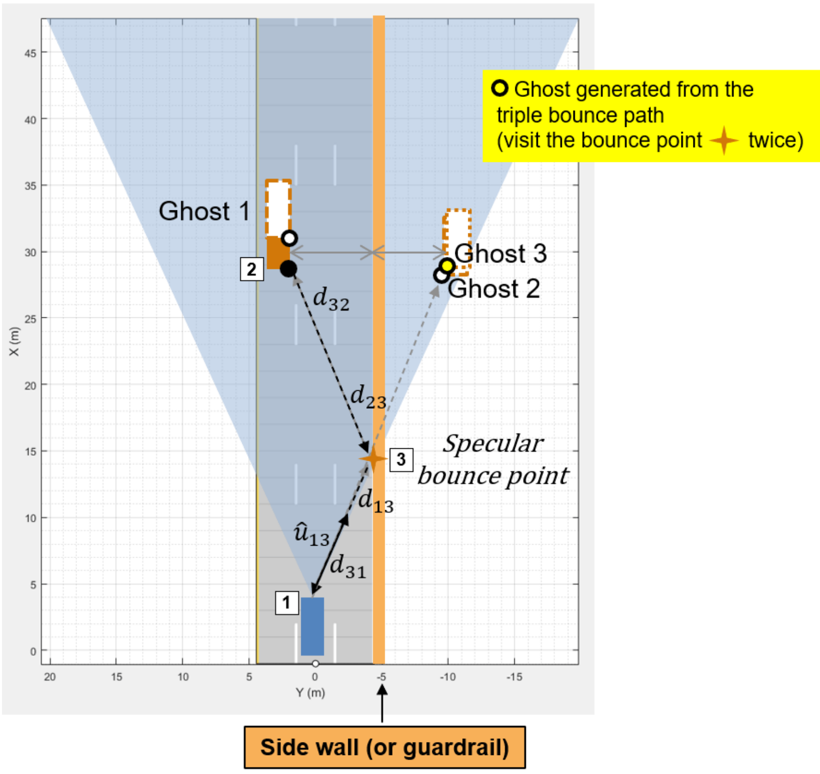

Figure 2.2.2. The Case of the Voxel under Three Bounce Paths [11]

In practical multi-antenna radar systems, there is a third type of interference called crosstalk. Crosstalk interference results from signal leakage between the transmit and receive channels inside the same radar system or between adjacent radar modules in a multi-radar configuration. Usually modeled as a signal with comparable characteristics to the victim radar's transmitted signal but with different timing and possibly frequency changes, this kind of interference in simulation is caused by channel coupling. This type of interference is especially difficult because it can be related to the signal we want, and future research might look at multi-antenna radar systems that include direction of arrival (DoA) processing, where this interference is given more focus.

Each of these interference types presents unique challenges for detection and mitigation. The spectral characteristics of these interference modes make target detection more challenging, and thus suppression techniques are particularly needed.

2.2.2 Impact of Interference

Radar interference greatly affects system performance over several assessment metrics. Interference can reduce radar signal quality in several significant ways:

Interference raises the background noise in the RD map, so degrading detection performance. This higher noise level reduces the signal-to-noise ratio, so distinguishing actual targets from background noise is more difficult and increasing the likelihood of missing detections and generating false alerts.

Range and Velocity Estimation Errors: Interference can distort both the amplitude and phase information of target signals, which leads to inaccurate estimation of target range and velocity. Even when targets remain detectable, their estimated parameters may contain significant errors that compromise tracking performance.

With those hazards considered, several key metrics are used to quantify the negative effects:

Mean Square Error (MSE):

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}|s_i - \widehat{s_i}|^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\left(s_i^R - \widehat{s_i^R}\right)^2 + \left(s_i^I - \widehat{s_i^I}\right)^2\right] \qquad \text{(Eq. 7)}$$

In radar signal processing, since the signal is usually in complex form (including amplitude and phase information), MSE calculation needs to consider the difference between the real part and the imaginary part at the same time. In the formula, represents the original signal sample, represents the reconstructed or processed signal sample, and the superscripts R and I represent the real part and the imaginary part respectively.

Signal-to-Noise-plus-Interference Ratio (SNIR):

$$\mathrm{SNIR} = \frac{P_{\text{signal}}}{P_{\text{noise}} + P_{\text{interference}}} = \frac{\left|A_{\text{signal}}\right|^2}{\sigma_{\text{noise}}^2 + \sigma_{\text{interference}}^2} \qquad \text{(Eq. 8)}$$

where P_{signal} is the power of the desired signal, σ^2_{noise} denotes the noise power, and $\sigma^2_{\text{interference}}$ represents the power of interfering signals. SNIR is a critical performance metric in radar systems, particularly under non-ideal conditions involving external interference.

Probability of Detection (P_d):

$$P_{\text{d}} = \int_T^{\infty} p(x|H_1)\ dx \qquad \text{(Eq. 9)}$$

where T is the detection threshold, and $p(x|H_1)$ denotes the probability density function under the alternative hypothesis, i.e., when a target is present.

Probability of False Alarm (P_{fa}):

$$P_{\text{fa}} = \int_T^{\infty} p(x|H_0)\ dx \qquad \text{(Eq. 10)}$$

where T is the detection threshold, and $p(x|H_0)$ is the probability density function under the null hypothesis i.e. no target presents. In radar signal processing, this metric is commonly referred to as the false alarm rate (FAR).

Earlier figure's range-Doppler maps show how interference transforms a clean map with clear target peaks into a noisy one, so masking the identification of targets and reducing detection reliability and estimation accuracy.

2.2.3 Development Environment and Tools

This section describes the development environments, software tools, and data structures employed in this research for both radar signal simulation and deep learning model implementation.

Utilizing the various toolboxes, including the Radar Toolbox, which is vital for producing realistic radar signals, applying RD processing, and examining radar data—MATLAB R2022b was the main platform for radar signal simulation and analysis. This toolbox offers detection, target modeling, and waveform generation functions.

The Signal Processing Toolbox was used for general signal processing operations, such as filtering, FFT analysis, and window functions crucial for accurate radar signal processing. Phased Array System Toolbox was employed for designing and simulating radar sensor arrays, modeling interference sources, and implementing spatial filtering techniques.

In the Python development environment, Python 3.11 was used for implementing the radar data pre-handling, the deep learning models, and the evaluation process, with Visual Studio Code (VS Code) as the integrated development environment (IDE). Key Python libraries included:

PyTorch (1.13.0) is the core deep learning framework, providing tensor computation, automatic differentiation, and GPU acceleration capabilities. It was utilized for the design, training, and evaluation of all neural network models. NumPy is regarded as a fundamental library for scientific computing, used for efficient array operations and numerical calculations. SciPy is used for signal processing functions, specifically FFT operations, window functions, and various filtering methods. In addition, Matplotlib, a visualization library, was utilized for plotting RD-maps, signal profiles, and training metrics. Mat73 is a specialized library for loading MATLAB MAT-files containing radar data into Python.

Radar data structure is then considered in this study. The primary data structure used throughout this research contains complex-valued radar signals. The radar cube has the dimensions of samples × chirps × frames. Samples (range dimension) represent discrete range bins. Chirps (Doppler dimension) contain frequency modulated pulses for velocity estimation. Frames (time dimension) represent instances of measurements at different times, which may involve measurements with different radar parameters and different driving environment settings.

In general, a radar cube's every element is shown as a complex value with the real component representing the in-phase (I) component and the imaginary part representing the quadrature (Q) component of the received signal. Accurate radar signal processing depends on amplitude and phase information, both of which this refined representation preserves. The data flow is modeled as if the simulated data were generated in MATLAB, exported to a MAT file, then imported into Python for deep learning processing and comparative evaluation.

2.3 Traditional Interference Mitigation Techniques

Automotive radar systems mainly use FMCW or linear frequency modulation sequence technology to estimate the target distance and relative speed. Traditional interference suppression methods include several major categories:

Time Domain Methods: Common methods include detecting and zeroing time domain samples contaminated by interference or using short pulse receiving windows to reduce the probability of interference. Though simple to apply, these techniques inevitably result in the loss of important information when the interference duration is extended or the target and interference signals overlap, therefore compromising the detection accuracy [1].

Dynamic filter coefficient adjustment by adaptive filtering methods like the Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithms track interference changes. Even though LMS is straightforward and simple to use, it might be slow to converge in some situations. Conversely, RLS offers quicker convergence but at the price of more computational complexity and noise sensitivity, which could create problems for real-time implementation in automotive radar systems [42].

These characteristics necessitate careful consideration when selecting appropriate adaptive filtering methods for interference mitigation in such applications [22].

Matrix Decomposition-Based Approaches: These comprise independent component analysis (ICA) and principal component analysis (PCA), which are used to decompose target and interference signals. These methods usually assume that the signals have different statistical characteristics, which will lead to poor separation when the target signal amplitude is low, or the interference energy is widely distributed.

Those methods all come with two advantages: the first one is high theoretical maturity and relatively simple implementation. Besides that, it also brings satisfactory interference suppression effects under specific conditions (limited interference sources or stable patterns).

However, the limitations of them are obvious as well:

- Information Loss: Time domain zeroing methods usually suppress part of the target signal while removing interference samples
- Poor Adaptability: Traditional filters have difficulty adjusting parameters in real time for non-stationary dynamic interference
- Phase Information Loss: Many traditional frequency domain filtering methods prioritize amplitude recovery and lack effective phase information preservation

Computational complexity may also be a problem, considering the adaptive filtering and matrix decomposition methods have high computational requirements, challenging for on-board real-time processing. Although traditional radar interference suppression methods remain practical in simple or specific scenarios, their limitations become increasingly apparent as the number of on-board radar systems increases and interference scenarios become more complex.

2.4 Deep Learning Approaches

Present automotive radar interference reduction calls for thorough performance assessment across multiple sectors. Recent developments in deep learning have produced a number of novel methods for reducing radar interference.

By considering RD maps as image-like representations, one study shows that the fully convolutional neural network (CNN) has exhibited exceptional efficacy in processing them. While restoring target signal amplitudes, these techniques use the strong spatial feature extraction capacity of convolutional layers to reduce interference. Many CNN-based techniques, therefore, have a major drawback: they do not retain phase information well, which is essential for precise Doppler estimation [2].

Another study suggested a recurrent network for a temporal processing method using recurrent neural network (RNN) architectures, particularly those using gated recurrent unit (GRU) or long short-term memory (LSTM) networks, which have been used to capture the temporal dynamics in radar beat signals. Self-attention techniques help these models to follow time-varying interference patterns more. RNN-based methods show great potential for temporal modeling, but they struggle to preserve amplitude and phase fidelity across the whole range-Doppler spectrum, which is usually constrained by training stability and computing cost [2].

One particularly intriguing path right now is advanced autoencoder architectures, which is to unfold iterative optimization methods into deep network structures. Recent studies have shown how well unfolding a resilient principal component analysis (RPCA) model is coupled with a residual overcomplete autoencoder (ROC-AE) block [3] works. There are various benefits to this strategy, one of them is the subtle signal characteristics are better captured by the overcomplete latent representation, which is higher dimensionality than the input. Residual links enable consistent training and gradient flow. While keeping computational efficiency, the unfolded optimization structure offers interpretability.

Particularly in conserving amplitude and phase information while keeping real-time processing capabilities, these sophisticated architectures show better performance in complicated interference situations.

2.5 Autoencoder Architectures

Recent research has shown how effectively autoencoder designs are developed particularly for radar interference reduction. One previous study proposed a totally convolutional denoising autoencoder for noise reduction in ECG signals by sharing designs relevant to radar signal denoising [8]. Their approach not only offered signal compression options but also surpassed denoising performance, therefore maintaining signal integrity and reducing the data size by 32.

More directly applicable to automotive applications, Fuchs et al. [9] created a convolutional autoencoder particularly for reducing car radar interference. Their method approaches interference reduction as an image denoising operation conducted directly on the RD spectrum, therefore bypassing preparatory interference detection stages. Their approach gathered notable increases in signal-to-noise-plus-interference ratio by training the network on both magnitude and phase data, hence maintaining phase information better than conventional mitigation strategies. This image-based method shows the possibility of considering radar interference suppression as a denoising issue in the transformed domain rather than in the original time domain.

An autoencoder is a specific kind of neural network designed for fast data encoding and reconstruction. Fundamentally, their architecture is made up of two main parts: an encoder that compresses the input data into a lower-dimensional latent representation and a decoder that tries to reconstruct the original input from this compressed representation. This bottleneck structure forces the network to learn the most salient features of the input data, making autoencoders particularly effective for noise reduction and signal restoration tasks.

Usually, conventional autoencoders handle signal sequences as time-domain representations in the context of radar interference suppression. Often implemented as a series of convolutional or fully connected layers with progressively lower dimensions, the encoder component compresses the interference-contaminated signal into a compact latent code. This compression acts as a filter, the network learns to keep necessary target information and throw away interference elements. The decoder then reconstructs a clean signal from this filtered representation, attempting to generate an output that matches the characteristics of interference-free radar returns, as shown in Figure 2.5.1.

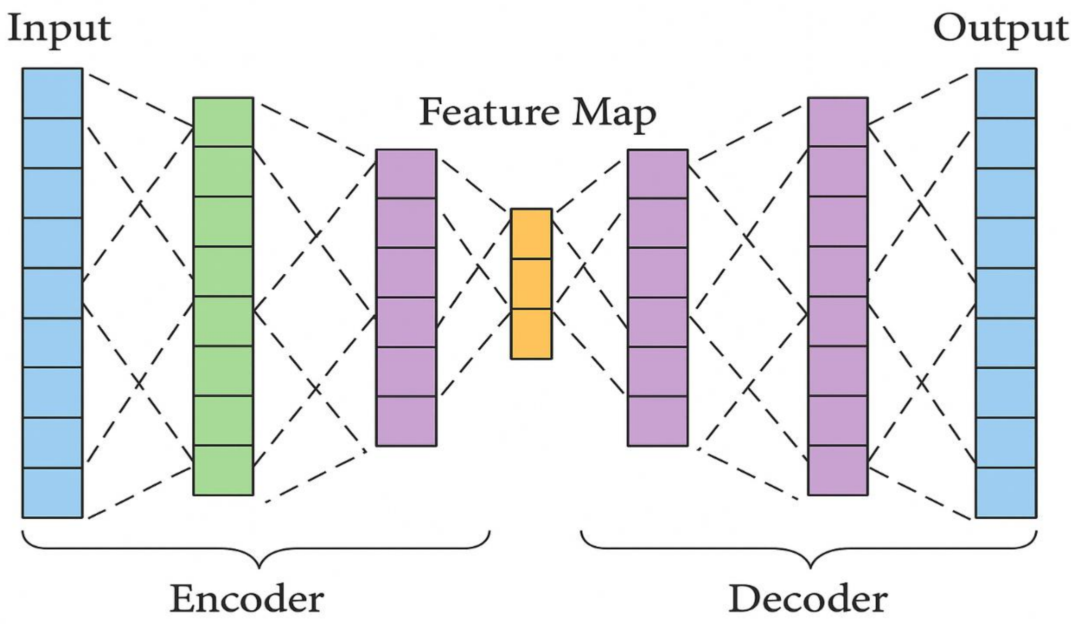

Figure 2.5.1. General structure of an Autoencoder

Several fundamental characteristics of autoencoders explain their effectiveness for interference suppression. First, their ability to learn non-linear transformations enables them to capture complex interference patterns that traditional linear filtering methods cannot address. Unlike traditional methods depending on pre-defined signal models, autoencoders dynamically learn the underlying structure of both target and interference signals straight from data. This data-driven strategy lets them manage various interference situations without needing clear mathematical modeling of every interference kind. Moreover, the adaptability of autoencoder designs allows for different improvements suited for radar uses. For example, denoising autoencoders are explicitly taught to rebuild clean outputs from damaged inputs by adding controlled noise throughout training. Especially for radar interference suppression, this training method is crucial since it mirrors the actual effort of recovering clean signals from observations compromised by interference. Variational autoencoders (VAEs) offer another development that could improve robustness against unexpected interference patterns by integrating probabilistic modeling in the latent space. Many of the advances have also concentrated on specific autoencoder variations for radar uses. Residual autoencoders allow gradient flow during training and preservation of fine-grained signal characteristics by use of skip connections between corresponding encoder and decoder layers. Accurate Doppler estimate depends on phase information, hence these residual connections are particularly useful for preserving it. Overcomplete autoencoders have shown better performance in capturing subtle signal properties and separating target returns from interference than undercomplete autoencoders, which use a latent representation with larger dimensionality than the input.

The unfolding optimization-inspired structure is yet another significant development. These designs transform iterative optimization methods into trainable neural network layers by combining the theoretical foundation of classical signal processing with the learning ability of deep networks. For example, autoencoders built on unfolded robust principal component analysis (RPCA) concepts have demonstrated extraordinary efficacy in breaking down radar signals into low-rank target components and sparse interference components.

Autoencoders also provide computational benefits that fit nicely with the real-time needs of vehicle radar systems. Once trained, the feed-forward feature of these networks enables efficient inference with constant processing costs. So allowing autoencoders to be employed in resource-constrained automotive environments, several model compression techniques—including pruning and quantization—can also aid in reducing the computational load without significantly compromising performance.

Notwithstanding these benefits, most current autoencoder methods for radar interference suppression still run in one domain, hence emphasizing either time-domain or frequency-domain processing. This constraint drives the investigation of multi-domain architectures able to concurrently use the complimentary data provided in several signal representations. Aiming to close this gap, the dual-path autoencoder method suggested in this study combines concurrent processing streams for time-domain signals and RD maps, hence perhaps attaining more complete interference suppression while keeping important target information.

The shortcomings of current autoencoder methods for radar interference suppression drive the investigation of multi-domain architecture suggested in this study. Most of today's autoencoder techniques run in one domain, hence they only concentrate on either time-domain or frequency-domain processing. For automotive radar interference, where various interference kinds show differently across domains, this restriction is especially concerning. By including parallel processing streams for both time-domain signals and RD maps, the dual-path autoencoder method seeks to close this gap and maybe achieve more complete interference suppression while maintaining important target information.

2.6 Research Opportunities

Several important research possibilities arise from the constraints of current techniques covered in earlier parts. Although most deep learning techniques and conventional methods work just in the time domain or frequency domain, actual radar signals include complementing information across domains. Although the frequency domain offers clear visualization of targets as distinct peaks, the time domain keeps entire phase information required for exact velocity estimation. Current methods do not effectively integrate these complimentary benefits, suggesting that a unified approach could perhaps counterbalance the limitations of single-domain processing.

Parallel processing of both domains instead of sequential processing, which signals are handled first in one domain and then translated to another which could provide more complete feature extraction. This approach would allow specialized processing to fit for the distinctive characteristics of each domain and provide natural redundancy against domain-specific corruption. For automobile radar applications, where interference patterns differ significantly, such redundancy could significantly improve resilience. An interesting area of study is also the creation of smart fusion techniques that may dynamically weight information from several areas depending on their dependability and relevance. This approach could dynamically change to various interference scenarios, hence maybe surpassing current fixed-strategy methods.

The found gaps in current research suggest a multi-domain strategy combining time and frequency representations could address fundamental limitations in automobile radar interference mitigation. Using complementary information across domains and employing adaptive fusion techniques could help to produce more robust systems able to control the more intricate electromagnetic environments of modern roadways.

3. Methodology and Development

3.1 Radar Simulation Environment

This chapter outlines the methodological approach used to create and assess the automobile radar interference reduction strategies. High fidelity radar signal simulation is combined with sophisticated deep learning techniques in an integrated strategy. This approach provides a solid basis for comparing conventional and deep learning-based methods to minimize these impacts and lets one carefully investigate interference effects in actual driving conditions.

The five key components of the study method are: simulation environment setup, automotive radar system design, highway scenario generation, interference modeling, and signal processing. Every component is carefully designed to ensure that the entire system accurately reflects real-world automotive radar problems while maintaining the flexibility needed for comprehensive research and algorithm development.

3.1.1 Simulation Environment Setup

A combined simulation system that combines MATLAB's signal processing capabilities with Python's deep learning tools helps to design and evaluate radar interference reduction techniques. This hybrid approach combines the benefits of both platforms: MATLAB's efficient radar signal processing and visualization facilities, and Python's huge relevant libraries.

Efficient data flow between MATLAB and Python components shapes the simulation environment. Radar parameter configuration, signal generation, highway scenario simulation, interference modeling, and RD processing for data validation are all done in MATLAB main environment. Python adds to this: neural network training, deep learning model creation, interference reduction processing, and model performance analysis.

MATLAB is the main platform for:

- Radar parameter configuration and signal generation
- Highway scenario simulation with multiple vehicles
- Interference modeling and injection
- RD processing for data validation

Python complements this with

- Deep learning model development
- Neural network training on simulated radar data
- Interference mitigation processing
- Model performance analysis

The integration between these environments is achieved through a file-based data exchange mechanism. Specifically, MATLAB creates and stores radar data with ground truth information, which is subsequently imported into Python for model training and inference. The processed Python outputs are then brought into another script for performance assessment and comparison.

A file-based data exchange system connects various environments. Specifically, MATLAB creates and stores radar data with ground truth information, which is subsequently imported into Python for model training and inference. The processed Python outputs are then brought into another script for performance assessment and comparison.

3.1.2 Automotive Radar System Design

This part presents a MATLAB-based FMCW automobile radar simulation tool meant to produce high-fidelity radar signal data. Considering the interrelationships between several factors, the system parameters were chosen according to typical criteria of modern automotive radars.

The simulation of radar signals especially takes into account radar parameters and constraints. The fundamental radar system characteristics are carrier frequency fc, bandwidth B, sampling frequency fs, chirp duration tm, and chirp slope s. These characteristics have to meet a sequence of physical and signal processing limits:

- Range resolution determines the radar's ability to distinguish between two adjacent targets, governed by bandwidth:

$$\Delta R = \frac{c}{2B} \qquad \text{(Eq. 11)}$$

After considering the various constraints and requirements, a bandwidth of B = 600 MHz and a range resolution of ΔR = 0.25 m were selected in this design.

- With the sampling frequency and chirp slope the maximum unambiguous range can be calculated as:

$$R_{max} = \frac{c \cdot f_s}{2 \cdot s} \qquad \text{(Eq. 12)}$$

  To achieve a maximum detection range of 100 meters, the corresponding the sampling frequency of f_s=20 MHz and a slope of s=30 MHz/µs are designed.

- Velocity Resolution is calculated with following equation:

$$\Delta v = \frac{\lambda}{2 \cdot T_{PRI} \cdot N_d} \qquad \text{(Eq. 13)}$$

  Where λ is the wavelength. T_{PRI} is the pulse repetition interval, and N_d is the number of chirps. In this system, with f_c = 77 GHz and T_{PRI} = 20 µs, a reasonable velocity resolution with approx. 0.76 m/s is obtained.

These parameter choices highlight the basic trade-offs in radar system design, i.e. higher sampling rates can raise the maximum range but raise data volume. And longer chirp durations enhance velocity resolution but lower maximum observable velocity; increasing bandwidth adds system complexity but improves range resolution.

With those relationships and constraints considered, a particular radar configuration is proposed for simulation, which tries to find optimal balance. The key radar parameters used in the simulation are explained and listed in the following table 1:

Categories	**Symbol**	**Value**	**Description**
Carrier Frequency	fc	77 GHz	Industry standard for automotive radar
Bandwidth	B	600 MHz	Provides ~0.25m range resolution
Chirp Duration	T_{PRI}	20 µs	Balances velocity resolution and maximum range
Sampling Frequency	Fs	20 MHz	Ensures proper sampling of beat frequencies
Chirp Slope	S	30 MHz/µs	Derived from bandwidth and chirp duration
Number of Chirps	N_d	128	Chirps per frame

Table 3.1.2 Radar Parameter Settings for Simulation

3.1.3 Highway Scenario Generation

A meticulous highway scenario simulation framework is created to assess radar interference avoidance strategies in realistic settings. This module creates dynamic traffic scenarios with multiple vehicles reflecting different driving circumstances usually experienced on highways.

For scenario configuration, the highway scenario consists of a multi-lane roadway with vehicles traveling in both directions. The simulation environment includes

I. A three-lane highway in each direction
II. Variable vehicle speeds ranging from 80 km/h to 130 km/h
III. Random placement of vehicles across available lanes
IV. Realistic spacing between vehicles based on typical highway conditions

The ego vehicle (equipped with the radar system) is positioned at the reference origin, while other vehicles are placed at various ranges according to realistic traffic patterns. Each vehicle is defined by several key parameters. For modeling a realistic European highway driving scenario, the following aspects should be focused on:

Parameter	Value	Source/Reference
Lane Width	3.5-3.75 m	European Commission Directive 2008/96/EC [12]
Speed Limits	80-130 km/h	European Road Safety Observatory (ERSO) [13]
Vehicle Density	15-25 vehicles/km	Trans-European Transport Network (TEN-T) guidelines [14]
Safety Distance	Min. 2 seconds	European Traffic Safety Council recommendations [15]

Table 3.1.3 Scenario Configurations for European Highway Driving

Target Vehicle Generation Algorithm: The number and characteristics of target vehicles are randomly generated based on radar constraints and realistic driving scenarios. The generation algorithm ensures that:

- Target ranges do not exceed the maximum unambiguous range
- Relative velocities remain within the detection limits
- Targets are sufficiently separated to be resolvable
- At least one oncoming vehicle is included to create realistic interference conditions

The script of scenario visualization is then utilized to obtain an intuitive traffic setup. The generated highway scenario is visualized in a bird's-eye view to provide an intuitive understanding of the traffic situation. Figure 3.1.3 shows an example scenario with multiple vehicles at various ranges and lanes.

In the visualization, the ego vehicle (blue) is positioned at the coordinate [5.25, 0], with other vehicles shown at their respective ranges and lanes. While oncoming cars are displayed at their actual positions with negative speeds, vehicles moving in the same direction as the ego car show at positive velocities. A straight line of the same hue marks every car's direction of travel.

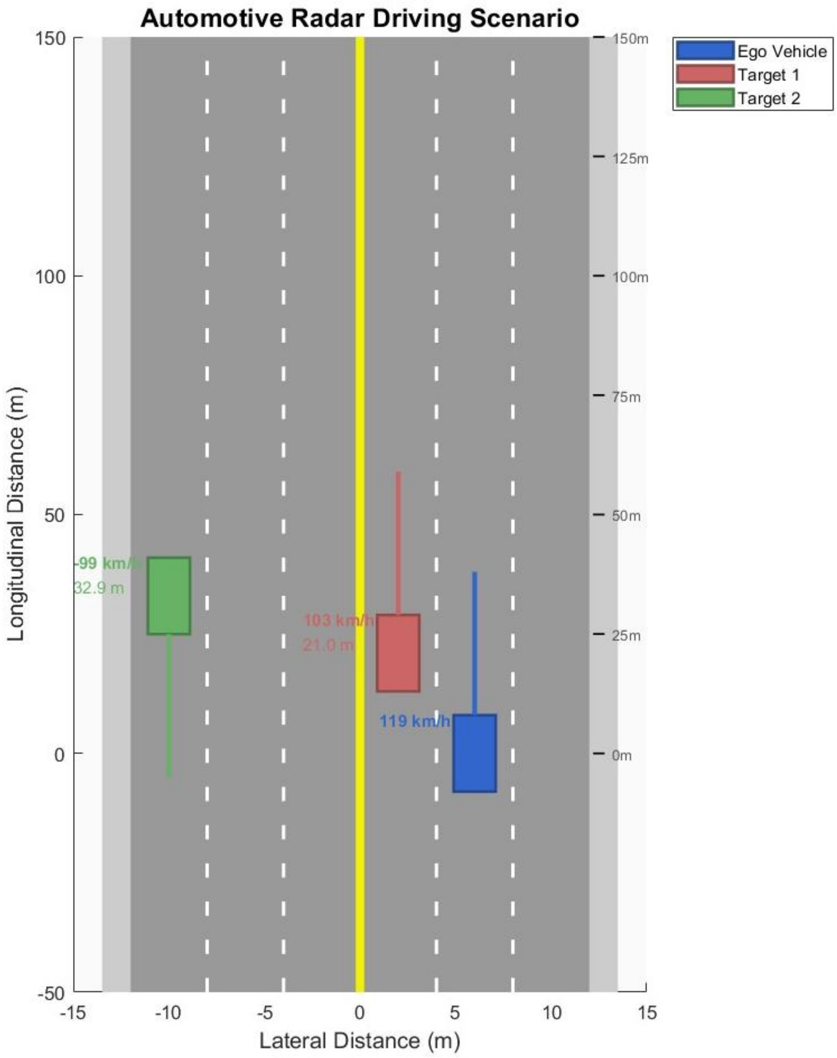

Figure 3.1.3. Bird's-eye View Visualization of the Highway Scenario

Each vehicle is labeled with its velocity to provide a comprehensive view of the traffic dynamics. The lane markings and range indicators help demonstrate the scenario within a realistic highway environment.

This visualization serves as a crucial validation tool to ensure that the generated scenarios match real-world driving conditions and properly represent the challenging interference situations that modern automotive radars must handle.

For vehicles traveling in the same direction as the ego vehicle, the relative velocity is simply the difference in speed. For oncoming vehicles, the relative velocity is the sum of the absolute speeds. The suggested scenario creation system offers a strong base for testing ways to reduce interference in many real-life driving situations, making sure that the methods being tested work well in actual car use.

3.1.4 Interference Modeling and Analysis

Assessing mitigation strategies under realistic driving conditions depends on the simulation of radar interference. Within the scope of this work, a thorough interference modeling method including two main interference types experienced in automotive radar situations—direct interference and multipath interference—is used. The paper also addresses crosstalk interference as a supplementary interference kind. Every interference kind is parameterized to reflect realistic conditions while keeping configurability for systematic assessment.

Interference Generation Process

The interference generation process begins with the configuration of global interference parameters that define the overall characteristics of the interference environment. These parameters include:

Parameter	Value Range	Description
Interference Power	0 to 20 dB	Overall power level of interference sources
Interference Types	Direct, Multipath	Types of interference to include
Surface Properties	Reflection coefficient: 0.3-0.5	Material properties for multipath
Power Distribution	10-70% per interference type	Relative contribution of each interference type

Table 3 Interference configurations

For each frame in the simulation, interference signals are generated based on the current vehicle configuration and added to the clean radar signals. The process ensures phase coherence and proper time-frequency characteristics to accurately model real-world interference.

Direct interference models the impact of radar signals from other vehicles operating in the same frequency band. The key parameters for direct interference include:

Frequency offset is randomly generated within ±100 MHz of the carrier frequency, representing the slight differences in operating frequencies between different radar systems. Chirp duration has a random offset within 0-100%, modeling the asynchronous operation of different radar systems. The changes of power scaling depend on how far away the interfering vehicle is, following the radar equation, plus a random adjustment of ±3 dB to consider differences in how the equipment is made and how the antennas work. Slope Variation: Random variation of ±10% in chirp slope, representing differences in radar configurations across manufacturers.

Multipath interference occurs when radar signals reflect off road surfaces, barriers, or other objects before returning to the receiver. The multipath model includes four factors:

Reflection surfaces are configured with specific heights (0-1.5 m) and reflection coefficients based on material properties (concrete, metal, asphalt). Path length calculation is geometrically calculated based on surface positions and vehicle locations.

Combined Interference Modeling

Appropriate scaling of all interference types in the last interference signal produces the correct signal-to-interference ratio (SIR). The procedure guarantees that:

- Total interference power reflects reasonable distributions seen in field observations.
- Temporal and spectral features fit actual interference patterns.
- Different kinds of interference interact in physically correct ways.
- Frame-to-frame fluctuation reflects actual variations in the driving environment.

3.2 Data Preprocessing and Integration

The integration of radar simulation data from MATLAB into a Python-based deep learning system provides a crucial link between traditional radar signal processing and modern machine learning techniques. This section outlines the systematic approach used for importing, processing, and preparing the complex-valued radar data to fit neural network training. Comprising several stages, the preprocessing pipeline imports data from MAT73 files, manages complicated data types, organizes different scenarios, does preliminary data analysis for validation, and converts the data into formats useable by the neural network. Every level was carefully designed to preserve the basic characteristics of radar signals while permitting effective deep learning model training.

3.2.1 Data Structure and Import

The transition from MATLAB-based radar simulation to Python-based deep learning called for careful consideration of data structures and formats. Exported from MATLAB as MAT-files (version 7.3), the simulation data stores variables hierarchically using HDF5. This section describes how to import and manage this extensive data into the Python environment.

Every MAT73 file had two main radar data sets: one with interference and one without. The measurements corresponded to 128 chirps per frame for Doppler processing, and 256 range samples corresponding distance bins. The publications additionally included metadata variables with simulation parameters such radar specifications and target information as reference.

Data import used a particular Python package made for loading MATLAB 7.3 format files. The execution allowed for combining and loading several driving scenarios into one dataset. Every scenario consists of various driving conditions, target configurations, and interference patterns. A scenario information allowed for unified processing while keeping knowledge about the original material by means of metadata for each scenario. This method preserved the integrity of individual simulation settings while allowing for systematic handling of various driving circumstances.

Stored as NumPy arrays with complex128 data type, the imported radar data preserved amplitude and phase information for radar analysis. The study of the imported data revealed usual distributions of signal amplitudes; interference conditions had far greater amplitude variations than clean reference signals. Particularly for neural network input preparation, which usually calls for real-valued inputs, the complicated character of the data called for unique handling in later processing stages. The next part on sophisticated data processing for deep learning applications covers this issue.

3.2.2 Initial Data Analysis and Visualization

Following successful radar data importation, a preliminary analysis was done to better understand the signal characteristics and confirm the integrity of the imported data. This study aimed to confirm the proper data transfer between environments, to grasp how interference affects radar signals, and to set baseline performance criteria for subsequent comparison. Target information is revealed by RD processing, which converts the time-domain radar signals into the frequency domain. A sequence of operations—windowing, FFT, and frequency shifting—was used to perform this transformation. Important for precisely identifying targets, the windowing process employed Hamming windows in both range and Doppler dimensions to lower spectrum leakage.

The research produced comparative representations of both clean and disturbed signals across several domains for every situation. The RD map, showing signal power over range and speed dimensions, was the main visualization. These maps showed how interference appeared as dispersed noise or false targets, maybe hiding or distorting valid targets. One-dimensional profiles were drawn and examined to go along with the two-dimensional maps. Obtained by calculating the maximum value along the range dimension for each velocity bin, the Doppler profile showed how interference influenced velocity estimation. The range profile, too, found variations in target range detecting capacity.

These visualizations showed the SIR and computed the dynamic range of both clean and interfering signals; statistical analysis was done on them. Peak detection found valid targets in the clean signal, which acted as ground truth for further performance assessment. Particularly useful was this peak information since it set the reference points against which the interference reduction strategies would be assessed. Emphasizing the variation in interference effects, this function summarizes the statistical results across several scenarios.

Neural network architecture's later design choices were driven by vital lessons from this first study, which also offered. It showed that interference influenced phase and amplitude information, hence implying that preservation of both elements would be necessary for efficient mitigation.

3.2.3 Data Organization and Scenario Management

After the first study, the radar data needed particular changes to be fit for neural network processing. Creating a suitable dataset structure, managing complicated values, and defining correct training and validation splits were the main goals of this preparation phase. To effectively handle the radar data and enable its use with deep learning frameworks, a special dataset class was created. The generated class "RadarDataset" contains the functionality for transforming and accessing radar signals. This class's primary function was to handle the complicated character of radar data and transform it into a format suitable for neural networks.

Neural network processing was fundamentally challenged by the complicated character of radar signals (complex 128 data format). This constraint called for the breakdown of complicated radar signals into their component elements. The defined class handled this decomposition by separating the real and imaginary components of the radar cube, therefore converting the data from a 3D complex-valued array to a 4D real-valued array with an extra channel dimension.

While ensuring compatibility with conventional neural network operations, the translation method maintained all signal information:

Parameter	Description
Original data	(Samples, chirps, frames) with complex128 values
Transformed data	(2, samples, chirps, frames) with float32 values
Channel 0	Real component
Channel 1	Imaginary component

Table 3.2.3 Radar Signal and Channel Information

Through means of shared weights, this channel-based model allowed the neural network to simultaneously process both components, hence is expected to understand the inherent links between amplitude and phase data.

Using a set ratio of 80% for training and 20% for validation, the dataset was split into training and validation subsets post transformation. Widely acknowledged in machine learning research, this 80/20 split offers a reasonable mix between keeping enough for meaningful validation and having enough data for training. Random sampling was used to accomplish the split, so guaranteeing both subgroups had representative samples from all accessible situations and so avoiding scenario-specific biases in the training process.

Model parameter optimization was done on the training subset; the validation subset acted as an independent assessment of model generalization, hence assisting in the detection of overfitting and direction of hyperparameter adjustment. Given the rather small number of radar frames, usually 100-200 frames across all scenarios. This split ratio guaranteed that roughly 80-160 frames were available for training and set aside 20-40 frames for validation.

Those setups complete the data preparation pipeline from advanced radar data to neural network-ready inputs, therefore stressing the transformation and splitting techniques. Maintaining the essential characteristics of radar signals, this well considered data preparation approach sets the stage for effective neural network training.

Choosing the perfect batch size for training involved considering many contradictory factors. Batch size influences model performance significantly as well as computing efficiency, thus these elements have to be judiciously balanced. In their empirical study, [19] found that lower batch sizes appear to enhance generalization performance; batch sizes between 2 and 32 typically provide models with better test accuracy. This outcome agrees with the theoretical work of [18], who demonstrated that large batch training tends to converge to sharp minimizers of the training function, which generalize less effectively than the flatter minimizers found by small-batch techniques.

A sensible batch size from 4 to 8 was selected for this work following experimental validation. This choice weighed computer problems against statistical learning features. Batch sizes should be adjusted depending on available memory and training stability since smaller datasets often gain from smaller batches [20]. A batch size of 4 guaranteed several updates per epoch while preserving consistent gradient estimates given the radar dataset's small amount of training samples—roughly 80-160 frames.

From a memory point of view, the high-dimensional radar data made this batch size reasonable as well. Larger batch sizes would have created notable memory needs, especially during the backward pass when gradients have to be recorded for all operations in the computational graph [18], given each frame's 256×128 complex values (subsequently translated to two real-valued channels).

Once the batch size was decided, distinct data loaders for the training and validation datasets were developed. Shuffling was allowed on the training dataset to randomize sample order in every epoch.

In neural network training, this shuffling action has several uses. [16] clarifies that randomizing the sample order prevents the network from learning misleading correlations between consecutive samples, enhances convergence by raising the stochasticity of the optimization process, and lowers the chance of local minima entrapment. Shuffling guaranteed that consecutive frames from the same situation were not always processed simultaneously for radar data particularly, so guiding the model to acquire scenario-invariant characteristics instead of storing particular temporal sequences.

By contrast, the validation data loader was set up without shuffling. This setup decision shows the various goals of validation data. Consistency between validation runs is more crucial than randomization since validation is used to assess model performance rather than change parameters. Keeping a consistent validation order guarantees that validation measures are comparable across epochs and experiments [17] and helps to track particular instances the model struggles with. To guarantee consistent memory use and to immediately compare training and validation metrics, both loaders were set with the same batch size. Maintaining the same size allowed for a more exact comparison of metrics between training and validation phases even while conventional implementations may employ bigger validation batch sizes to accelerate assessment.

3.2.4 Normalization and Transformations

Normalizing and changing the radar data to guarantee the best neural network training was the last preparation step. These actions were essential for enhancing model convergence, stability, and eventually performance on the interference mitigation challenge.

To handle the natural qualities of radar data, several stages of signal normalization were used. Unlike picture data, which frequently maintains a consistent scale, radar signal amplitudes can vary greatly depending on target reflectivity, distance, and interference levels. Initial statistical study of the radar cube showed significant amplitude differences between situations.

The change between time domain and frequency domain representations was a fundamental one for the dual-path architecture. A well-crafted processing function that reflected traditional radar signal processing methods converted the time domain signals into RD maps. Three main steps made up this change:

Hamming windows were first used in both range and Doppler dimensions to lower spectral leakage. Given its outstanding sidelobe suppression qualities, the Hamming window was chosen especially for radar processing since sidelobes can conceal weaker objects or be confused for spurious targets. The windowed data was then subjected to a 2D FFT, changing the signal from the time domain to the frequency domain. This modification was done utilizing PyTorch's native FFT methods, which provide GPU acceleration when available, therefore optimizing the efficiency of neural network computation. At last, FFT shift operations centered the zero-frequency component, hence generating typical RD maps with velocities centered around zero. For interpretation, this centered representation is more natural; for CNNs, it offers improved learning qualities.

The transformation function was designed to manage individual frame processing during evaluation and batch processing during training. Consistency between these representations was vital since both time domain signals and RD maps were employed as inputs to the dual-path architecture. The normalization and transformation process guaranteed that both domains had similar information in complementary representations, therefore enabling the neural network to use various features of the signal characteristics for interference reduction.

Although batch normalization is often used in deep learning models to normalize inputs at every layer, a conscious choice was made to omit this method from the last preprocessing step. Several radar-specific factors influenced this decision. By design, batch normalization normalizes inputs depending on batch statistics, which could change the subtle amplitude correlations in radar signals vital for precise target recognition. When it comes to interference reduction, maintaining the original signal qualities as near as feasible was considered more crucial than the putative training advantages of normalization. Radar data's complex-valued character also offers special difficulties for batch normalization. Originally designed for real-valued data, the method may distort phase relationships vital for radar processing when used to complicated signals split into real and imaginary parts. Independent normalization of real and imaginary components could degrade important target qualities carried by phase information in radar signals. Moreover, batch normalization creates sample dependencies inside a batch, which could be troublesome for radar applications since every frame could represent separate situations with varying interference patterns and target setups. Given the memory limits, the tiny batch sizes used in this study would make batch statistics insufficiently representative across many radar settings. The preprocessing method relies on careful scaling of the input data to a constant range rather than batch normalization, hence retaining the relative connections between signal components. This more conservative strategy guaranteed that important radar signal features stayed intact all through the processing pipeline, hence enabling the neural network to learn the actual patterns of both targets and interference instead of normalized representations.

Though finally not used for this study, data augmentation methods often employed in computer vision applications were examined. This choice was influenced by the particular kind of radar interference, whereby artificial augmentations might change the genuine interference patterns the model needs to learn or create false artefacts. The range of target configurations, interference kinds, and signal characteristics in the training data gave natural variation.

3.3 Model Architecture Development

Developing a good neural network design for reducing vehicle radar interference called for close examination of both theoretical ideas and pragmatic limitations. From first ideas to iterative improvements to the final dual-path model, this part describes the architectural exploration process. Several important criteria particular to radar interference mitigation guided the architectural development: preservation of vital target information, efficient separation of interference patterns from legitimate signals, computational efficiency for possible real-time operation, and generalization across different interference scenarios. These factors guided the investigation across several architectural paradigms, finally landing on a creative dual-path solution that exploits complimentary processing of time-domain inputs and frequency-domain representations. Beginning with the assessment of traditional architecture and their constraints, the next subsections describe this developmental path to the suggested dual-path architecture that solves these issues and finally the optimization process that improved the architecture for practical use in automotive systems. Every design choice during this time was driven by the particular qualities of radar signals and the particular difficulties of interference reduction in dynamic driving conditions.

3.3.1 Initial Ideas and Limitations

An iterative strategy with each iteration tackling certain shortcomings of prior methods guided the creation of a successful neural network architecture for radar interference reduction. Emphasizing their strengths, shortcomings, and the insights they offered for later projects, this part describes the evolution of architectural ideas investigated during this project.

Linear Fully Connected Autoencoders

The first method used a simple autoencoder architecture consisting of totally connected (linear) layers. Motivated by the autoencoder's proven capacity to learn compressed representations of data, perhaps isolating signal from interference, this approach was the architecture had a symmetrical decoder to rebuild the clean signal following an encoder that progressively lowered dimensionality through dense layers, hence producing a bottleneck representation. Although theoretically straightforward, this method proved quite constrained when used on radar data. High-dimensional input data caused parameter explosion, which was the main difficulty. Every radar frame had complicated values (converted to two channels), hence the input layer needed more than 8 million parameters for one completely connected layer. Moreover, even with the small training data in the heap area, it took more than 2 hours to train. This not only made the model computationally expensive but also increased the risk of overfitting given the limited training data available. Additionally, fully connected layers discarded the inherent spatial structure of radar data. Range and Doppler dimensions contain important localized patterns that benefit from maintaining their spatial relationships—a characteristic that dense layers failed to preserve. Performance analysis showed that while the linear autoencoder could reduce general noise, it struggled to preserve sharp target peaks and often introduced symmetrical artifacts in RD maps, indicating inadequate learning of the underlying signal structure.

CNN-based Methods

To address the limitations of fully connected architectures, CNN approaches were explored. CNNs offer significant advantages for radar signal processing due to their ability to exploit spatial hierarchies and local patterns while using parameter sharing to reduce model size. The implemented CNN-based autoencoder used a series of convolutional and pooling layers for encoding, followed by transposed convolutions for decoding. These models showed improved preservation of spatial features and reduced the parameter count significantly compared to fully connected networks. Performance analysis revealed better preservation of target peaks and reduced artifacts in the reconstructed signals. However, pure CNN approaches still present limitations. While effective at capturing local interference patterns, they struggled with global interference characteristics that span across the entire RD map. Additionally, achieving sufficiently deep representations required multiple pooling operations, which risked losing fine-grained details crucial for accurate target detection.

Hybrid Architectures

For seeking to combine the global feature learning capabilities of fully connected layers with the spatial feature preservation of CNNs, several hybrid architectures were implemented. These designs typically employed convolutional layers for initial feature extraction, followed by a compressed fully connected representation, and then convolutional layers again for reconstruction. While these hybrid models showed promise, they still faced several challenges. Parameter counts remained high due to the fully connected bottleneck, though substantially reduced from the initial linear models. More critically, the transition between convolutional and fully connected layers created information bottlenecks where spatial information was lost, making it difficult to reconstruct fine details accurately. Another limitation across all these approaches was their exclusive focus on either time-domain or frequency-domain processing. Analysis of interference patterns revealed that certain characteristics were more apparent in one domain than the other, suggesting that an effective solution would need to leverage complementary information from both domains.

It was shown that the single modality, which in this circumstance is conventional autoencoder architectures, whether operating purely in the time or frequency domain, shows the performance limitations. These insights from early architectural explorations—the need for parameter efficiency, spatial information preservation, both local and global feature learning, and multi-domain processing—directly informed the development of the dual-path architecture presented in the next section.

3.3.2 Dual-Path Architecture

Focusing on maintaining the phase information and temporal properties required for precise Doppler estimation, the time-domain branch runs a sequence of convolutional layers on the complicated beat signal (divided into real and imaginary components). The RD map is processed by the frequency-domain branch utilizing a specialized CNN architecture with an eye toward spatial feature extraction and target peak preservation. Particularly in preserving signal integrity across several domains, this parallel processing approach has shown to be quite successful [6].

The dual-path architecture is based on the realization that radar interference shows differently in time and frequency domains. Some forms of interference, like narrowband interference, show up as localized patterns in the frequency domain but are spread out across the time domain. On the other hand, pulsed interference could be more easily seen in the time domain. Processing both domains concurrently helps the network to take use of these complementing traits and produce stronger interference reduction. Inspired by previous uses in voice separation and audio processing [4, 5], this dual-path method used a similar "multi-stream" architecture that succeeded in retention time and spectrum qualities.

Emphasizing its key components and data flow, Figure 3.3.2 shows the whole dual-path design. Four essential components of the design are: time-domain processing path, frequency-domain processing path, feature fusion mechanism, and reconstruction decoder.

Although the suggested architecture is motivated by the traditional autoencoder concept, it differs from the usual definition in several respects. Specifically, the design includes dual-domain inputs and specific encoding routes, which differ from the traditional idea of compressing a single input modality into a latent representation. More precisely, this framework is a prior-enhanced autoencoder in which the model uses complimentary representations from the time and frequency domains to enhance feature learning. Though structurally different, architecture preserves the fundamental idea of information preservation and reconstruction; its design is theoretically supported by the increased expressiveness and durability provided by multi-domain priors.

The time-domain path handles the intricate radar signals straight. Starting with a sequence of convolutional layers that increasingly extract hierarchical features while lowering spatial dimensions, it. Passing through several convolutional blocks with LeakyReLU activations and max pooling operations, the input with dimensions (2 x 256 x 128) represents real and imaginary components. With early layers identifying local patterns like signal discontinuities and later layers collecting more complicated interference structures, every convolutional block collects progressively abstract patterns in the time-domain data. LeakyReLU activation functions allow to keep gradient flow across the network, hence solving the vanishing gradient problem that may influence deep designs.

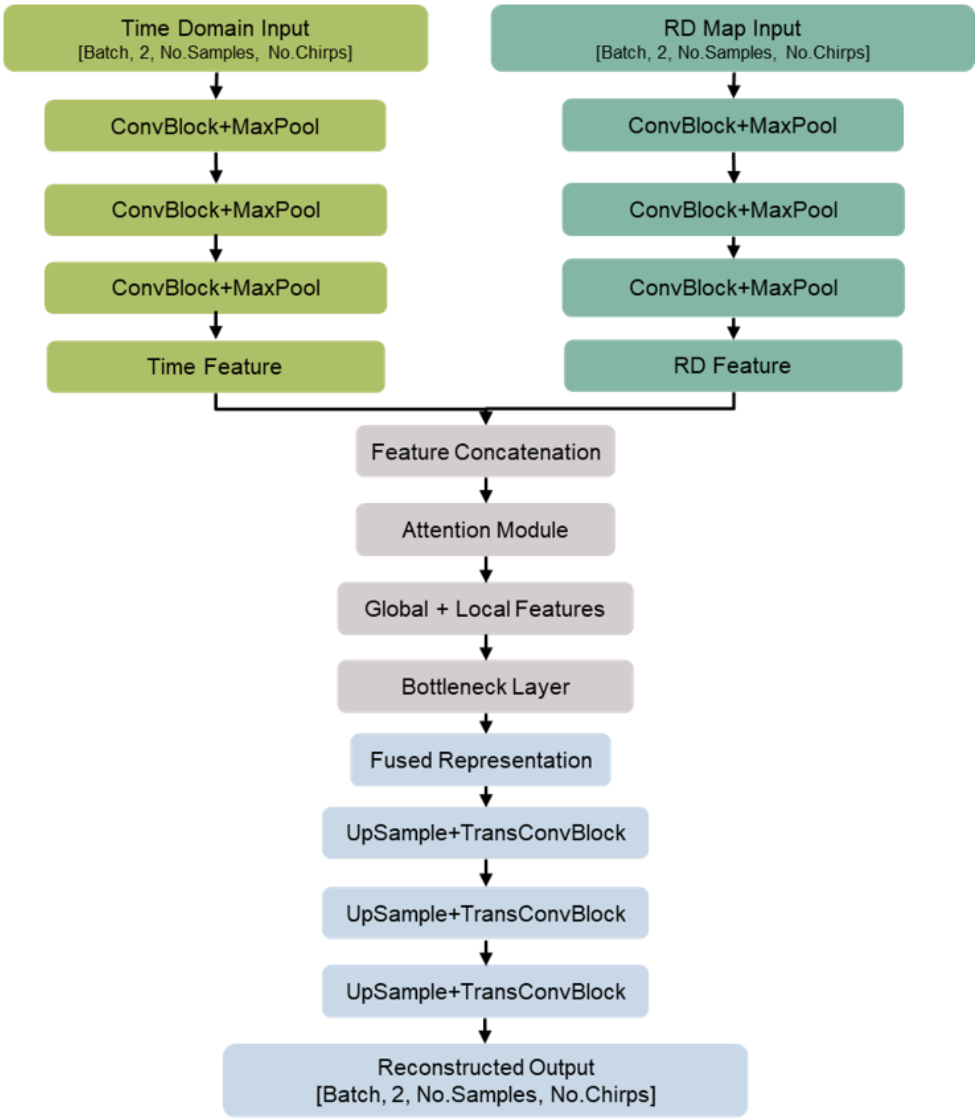

Figure 3.3.2. Dual-Path Radar Autoencoder Architecture

At the same time, the frequency-domain route processes RD maps generated from the same input data. Though especially meant to find patterns typical of frequency-domain representations, this path resembles the time-domain path in structure. The RD maps with dimensions ($2 \times 256 \times 128$) undergo comparable convolutional processing, extracting characteristics particularly pertinent to this domain's representation of targets and interference.

The feature fusion method that combines information from both processing streams is a major architectural innovation. The model uses a mix of attention methods and 1×1 convolutions to smartly weight and blend features instead of just adding features from both routes. Based on the particular interference pattern being analyzed, the lightweight "Squeeze-and-Excitation" attention mechanism lets the network concentrate on the most pertinent features from each domain [7]. Depending on which offers more beneficial information for a given interference situation, this method allows adaptive focus on either time or frequency domain characteristics. Effectively producing a joint representation that keeps the most useful information from both domains, the 1×1 convolutions lower channel dimensionality while learning the best feature combinations.

The last part is the decoder, which rebuilds the combined representation into pure radar signals. Progressively raising spatial dimensions and lowering feature channels, the decoder uses transposed convolutions, also known as deconvolutions. Beginning with the fused feature representation, the decoder uses a sequence of transposed convolutional layers with LeakyReLU activations to produce an output with the original dimensions ($2 \times 256 \times 128$) that represents the reconstructed clean beat signal.

One remarkable feature of this architecture is its end-to-end character; it directly translates interfering radar signals to clean reconstructions without calling for clear intermediate processes. This lets the network develop an optimal interference reduction plan straight from the data, maybe finding more efficient solutions than hand-crafted signal processing techniques.

The proposed model contains approximately 17 million parameters, with the distribution heavily weighted toward specific components. The initial preprocessing stages in both paths are relatively lightweight, with convolutional layers in the time and frequency domain paths each containing only a few thousand parameters due to their shared weight structure. However, the linear layers account for the majority of parameters, particularly in the transitions between convolutional feature extraction and feature fusion components.

In the time domain path, the convolutional layers apply filters with 3×3 kernels to capture local patterns in the radar signal, followed by LeakyReLU activation and max pooling operations that reduce spatial dimensions while preserving important features. This pattern repeats with increasing filter counts to progressively extract more complex signal representations while reducing spatial dimensions. The parameter efficiency of these layers comes from the weight sharing natural in convolutional processes, which lets the network handle high-dimensional radar signals with fairly few parameters.

Processing the RD maps through parallel convolutional layers, the frequency domain route follows a comparable topology. The model can learn complementary features across domains and keep the parameter count balanced by preserving architectural similarities between the two routes. The distinct processing routes guarantee that all domain's particular qualities are retained prior to fusion.

The feature processing stage, when convolutional layer flattened features are processed, is the most parameter-intensive part. Reflecting the difficulty of producing a cohesive representation while maintaining vital signal information, this stage generates more than 90% of the parameters of the model. Though it has many parameters, this part is vital for understanding the intricate interactions between temporal and frequency domain characteristics that support efficient interference reduction.

3.3.3 Parameter Optimization

To balance performance, computational economy, and practical limits for automotive applications, the dual-path architecture underwent multiple optimization stages. This part describes the essential implementation choices and refining process that formed the final model in the next sections.

Strategies for Parameter Efficiency

Although efficient, the first uses of the dual-path design had more than 70 million parameters—far over reasonable limitations for car use. A methodical approach to parameter reduction was thus followed, emphasizing three main tactics.

Increasing convolutional stride and adding more pooling layers to lower feature map dimensions before reaching parameter-intensive levels helped first to improve the input preprocessing processes [22]. This change preserved important signal properties while lowering spatial dimensions. Second, either removed or significantly downsized were completely connected layers, which accounted for most of the parameters. The first design had several fully connected layers each with more than 16 million parameters. In these areas, almost 90% of the parameter count was lowered by replacing these with a mix of global average pooling [23] and far smaller dense layers. Before computationally costly methods, 1×1 convolve were deliberately used for channel dimensionality reduction. Popularized in networks like GoogleNet [24], this approach let expressive feature combinations while reducing parameter count. By means of these optimizations, the model parameter count was effectively lowered from more than 70 million to under 20 million—a drop of almost 72%—while preserving interference mitigation efficacy. This significant drop made the concept feasible for possible real-time use in car radar systems [10].

Attention Mechanism Implementation

A key part of the dual-path architecture, the attention mechanism needed meticulous implementation to properly integrate data from temporal and frequency domains [35]. A lightweight channel attention method was created instead of computationally costly self-attention as found in transformer designs [36]. Based on the combined characteristics from both routes, the system produced attention weights allowing the model to dynamically stress more relevant elements depending on the particular interference pattern. Requiring just a tiny portion of the overall model parameters, the attention module used global average pooling followed by channel-wise weighting to achieve a minimal parameter footprint [37]. Testing showed that this attention mechanism was especially good at managing situations when single domain's interference traits were more noticeable than the other, therefore enabling the model to adaptively concentrate on the more informative representation [38].

Activation Function Selection

Model convergence and interference reduction performance were greatly influenced by the selection of activation functions. Early experiments with Rectified Linear Units (ReLU) showed limitations when processing radar signals, particularly in preserving phase information contained in negative values [39]. Comparative analysis of various activation functions led to the adoption of Leaky ReLU with a negative slope of 0.2 for most intermediate layers [40]. This activation allowed for preservation of negative values while maintaining computational efficiency. For connections processing raw radar signals directly, hyperbolic tangent (tanh) activations were selectively employed to better preserve phase relationships, as tanh's range [-1,1] naturally aligns with the normalized complex components of radar signals [41].

3.4 Implementation and Optimization

Equally important is the training approach that allows the network to acquire suitable feature representations and transformations. Covering the training framework, hyperparameter selection, loss function design, and answers to training issues, this section describes the all-encompassing training approach created for the dual-path architecture.

Careful design of the training procedure took into account the particular qualities of radar data, including its complex-valued character, the relevance of phase information, and the necessity to maintain target detection capabilities. Training techniques were constantly honed throughout the development process using empirical performance, with specific focus on avoiding overfitting considering the somewhat small radar dataset available.

3.4.1 Training Framework

Implementing the dual-path architecture called for a thorough training system meant to manage the qualities of radar data and provide effective and reliable model convergence. Focusing on the structural components and flow of the training process, this part outlines the training approach created for this study. From data preparation through model optimization to performance assessment, the training framework was run as a self-contained function controlling the whole training process. This design decision allowed systematic comparison of various architectural variants and hyperparameter settings and helped experimental repeatability.

Key inputs for the primary training function were the initialized model, training and validation data loaders, epoch count, and learning rate settings. Its main output was the trained model together with a thorough history of training metrics and resource consumption statistics for later study. The structure of the function follows the conventional supervised learning model modified for the particular needs of radar interference reduction.

Before the training loop started, the framework set up the computational environment and automatically found and used GPU acceleration if it was available. Maintaining constant training behavior, our adaptive hardware selection guaranteed best performance across various computer settings. Monitoring memory use during the training phase offered insightful analysis of the computing needs of several architectural configurations.

Ranging from the specified number of epochs to early ending criteria, the epoch-based training loop was the framework's essential component. Every epoch had two separate phases: a training phase where model parameters were changed and a validation phase where generalization performance was assessed.

The model was set to training mode during the training phase, therefore allowing dropout layers and batch normalization updates. The method then ran through mini batches from the training loader, with each iteration following a consistent sequence:

I. Input data preparation, including transfer to the appropriate computing device

II. RD map generation for the frequency domain path
III. Forward pass through the model's dual-path architecture
IV. Loss computation using the custom radar-specific loss function
V. Backpropagation to calculate gradients
VI. Parameter updates through the optimizer

For each mini-batch, the loss value was recorded to track within-epoch performance. This fine-grained monitoring proved valuable for identifying challenging radar scenarios that produced anomalous gradients or unusually high losses.

The validation phase followed a similar structure but with several critical differences. The model was switched to evaluation mode, disabling dropout and freezing batch normalization statistics. The validation data was processed without gradient calculation or parameter updates, providing an unbiased estimate of model generalization. As with the training phase, validation losses were recorded for each mini-batch and averaged to produce the epoch validation loss.

After completing both phases, the framework performed several end-of-epoch operations. The learning rate scheduler evaluated the validation performance and adjusted the learning rate according to its algorithm. Training metrics, including training loss, validation loss, and current learning rate, were stored in the history dictionary for post-training analysis. Additionally, the framework generated periodic progress reports that displayed current performance metrics and learning rate information.

The early stopping mechanism evaluated validation performance at the end of each epoch. If the current validation loss improved upon the previous best, the model state was saved and the patience counter reset. Otherwise, the patience counter incremented, and if it exceeded the predefined threshold, training terminated early. This method guaranteed enough training time and avoided overfitting. Upon finishing, the function can provide the trained model together with the training history and resource use measures. This all-encompassing output allowed for careful assessment of both the efficiency of the training process itself and the final model performance.

3.4.2 Hyperparameter Selection

Appropriate hyperparameter choice is quite important for the performance of deep learning models. Given the complicated character of the data and the needs of the application, this selection procedure was very important for radar interference reduction [16]. This part explains the hyperparameter selections for the dual-path design as well as the reasons behind them.

Setting the Learning Rate

Arguably, the most powerful hyperparameter influencing model convergence and ultimate performance is the learning rate [32]. An initial learning rate of 0.001 was chosen following methodical testing. This value allows the model to make significant progress in early epochs without risking divergence, hence balancing training speed and stability.

An adaptive scheduling method using the "ReduceLROnPlateau" scheduler was used instead of a constant learning rate across training [33]. This scheduler tracked validation loss and lowered the learning rate by a factor of 0.7 for 8 consecutive epochs of performance plateau. To avoid too much parameter stagnation, the minimal learning rate was 1e-6. Where different interference patterns could need different optimization paths, this adaptive approach was quite successful for radar interference reduction. Empirically, the scheduler's patience value (8 epochs) was set to balance the necessity to differentiate between real plateaus and typical performance variations [34].

Algorithm for Optimization

Convergence speed and ultimate model performance are both greatly influenced by the selection of optimization algorithms. The Adam optimizer was chosen for this application in this study. Adam is well suited for radar interference reduction because of its adaptive moment estimating qualities, which allow for significant variation in gradient magnitudes across several model components and various interference situations [25].

Following Kingma and Ba [26], the Adam optimizer was set with beta values of β_1=0.9 and β_2=0.999. These numbers govern the exponential decay rates for moment estimates and have been proved empirically to perform effectively across a wide spectrum of deep learning applications. To avoid overfitting by penalizing large weight values, weight decay (L2 regularization) of 1e-5 was included in the optimizer setup [27].

Batch Size Factors to Think About

Choosing batch size required balancing memory limits, statistical learning characteristics, and computational efficiency. After thoughtful evaluation of these elements, a batch size of four was selected for this use. Theoretical and practical factors both guided the choice of this quite little batch size.

From a learning point of view, lower batch sizes add good noise to the gradient estimation process, hence perhaps enhancing generalization by enabling the model to escape abrupt local minima [26]. Radar data is especially crucial in this regard since the small sample size heightens the danger of overfitting. While keeping consistent gradient estimates, the moderate batch size of 4 offered enough stochasticity.

Memory limits also affected this choice as, especially during backpropagation when intermediate activations have to be retained, the high-dimensional radar data with complicated values demands significant memory. The chosen batch size guaranteed that training could go on without memory exhaustion on standard GPU hardware [20].

Several regularization methods were explored; some were used to avoid overfitting and enhance model generalization. Gradient clipping is one of them; it limits the maximum gradient norm to 1.0 to avoid explosive gradient updates often caused by strong interference patterns. As suggested by Pascanu et al. [29], this approach-maintained training by limiting dramatic weight changes and letting typical optimization run undisturbed.

Training was set to end if validation loss did not improve for 15 straight epochs following a minimum training duration of 50 epochs. This patience-based early halting strategy guaranteed enough learning time while avoiding overfitting to the training data [30]. Observed learning curves, which indicated that significant performance gains occasionally followed short plateaus, guided the minimum epoch criterion.

Though early stopping usually caused termination before this limit, the maximum number of training epochs was set to 100. Most model configurations either converged or started overfitting well before 100 epochs, which helped to set this upper limit. Rapid iteration across the relatively small dataset size—about 80-100 frames for training—made prolonged training possible if necessary. For model development, the combination of early halting with a sensible maximum epoch count was effective since it allowed enough training time and avoided too much computational cost on models at performance plateau [18].

3.4.3 Loss Function Design

Effective training in the dual-path architecture depended on the construction of a suitable loss function. Unlike traditional image processing jobs, where mean squared error or cross-entropy loss would be enough, radar interference mitigation called for a specialized loss function that targeted the particular qualities of radar signals and the particular goals of interference suppression. The primary goal of the loss function was to guide the model toward reconstructing clean radar signals from interfered inputs. However, simple pixel-wise comparison between the model output and clean target proved insufficient for radar applications. Such an approach would fail to capture the importance of preserving critical radar signal characteristics such as target peaks, phase relationships, and overall signal structure while removing interference.

A carefully designed composite loss function was therefore developed, combining multiple components to address different aspects of the radar signal reconstruction problem. At its foundation was the reconstruction loss, measuring the fidelity of the output signal compared to the clean reference:

$$\mathcal{L}_{\text{recon}} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{x_i} - x_i)^2 \qquad \text{(Eq. 14)}$$

This means squared error (L2 Loss) term calculated the average squared difference between the reconstructed signal and the clean reference across all dimensions. While this component provided a basic measure of reconstruction quality, it treated all points in the radar signal equally, regardless of their importance for target detection.

To address this limitation, a detection-focused component was introduced. This element specifically emphasized preservation of target characteristics in the RD domain:

$$\mathcal{L}_{\text{dtc}} = (1 - P_d) + 0.5 \cdot P_{\text{fa}} \qquad \text{(Eq. 15)}$$

The Pd measured the model's ability to preserve legitimate targets, calculated as the ratio of correctly identified peaks in the reconstructed signal to the total peaks in the clean signal. Conversely, the Pfa quantified the presence of spurious peaks in the reconstruction, calculated as the ratio of peaks present in the output but is absent in reference to the total peaks detected in the output. The weight factor of 0.5 for the false alarm term recognizes the relative importance of the two terms, i.e., missing a legitimate target is considered more detrimental than introducing a weak false alarm target.

In earlier iterations of this work, additional loss components were explored to further enhance performance. One such term, the peak magnitude fidelity loss, focused on preserving signal amplitude at target locations:

$$\mathcal{L}_{\text{mag}} = \frac{1}{M}\sum_{j=1}^{M}\left(\widehat{A_j} - A_j\right)^2 \qquad \text{(Eq. 16)}$$

However, empirical evaluation revealed that while these components provided marginal improvements in specific metrics, their overall contribution to the preservation of target peaks and robustness under diverse interference conditions was limited. In particular, the magnitude loss did not significantly improve target peak reconstruction, and the suppression term occasionally introduced undesired regularization effects that hindered generalization. To streamline training and reduce the risk of overfitting local characteristics, both components were subsequently removed from the final objective function.

The resulting total loss was defined as:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{recon}} + \beta \cdot \mathcal{L}_{\text{dtc}} \qquad \text{(Eq. 17)}$$

where α and β were experimentally chosen weighting elements (0.3 and 0.7, respectively), decided upon by methodical testing. Both loss components were tracked throughout training to guarantee equal convergence behavior.

3.4.4 Development Challenges and Solutions

Though the suggested dual-path architecture showed great promise in both theoretical formulation and early experimental findings, its evolution presented some non-trivial difficulties outside architectural design and hyperparameter optimization. Designing a successful feature fusion technique was among the top challenges. Given the architecture runs parallel processing of radar signal time-domain and frequency-domain representations, it was crucial to find the best way to combine the acquired characteristics from both modalities. Extensive testing with several concatenation and gating techniques was therefore required to determine how the fusion should maintain complimentary information without conflict or duplication.

Another major challenge was ensuring sufficient computational efficiency to meet the constraints of real-time processing in automotive environments. While deep learning models offer powerful representational capacity, their complexity must be carefully managed to ensure feasibility for deployment on embedded systems with limited computational resources. This necessitated iterative refinement of the model structure to balance performance and latency, including adjustments to the number of layers, filter sizes, and compression rates in the encoder-decoder pathways.

Processing complex-valued radar data posed a fundamental challenge for neural network implementation. While radar signals naturally exist in the complex domain, containing both amplitude and phase information, standard neural network operations are designed for real-valued data. The initial approach involved separating real and imaginary components into separate channels, but this risked losing the intrinsic relationship between these components. The solution implemented in the final model maintained this separation at the input level but incorporated specific architectural elements to preserve phase relationships. Cross-connections between real and imaginary processing paths enabled the network to learn the interdependencies between components rather than treating them as independent signals. This approach proved more effective than alternative methods such as magnitude-phase representation or complex-valued network layers, particularly for the task of interference mitigation, where phase information plays a crucial role in distinguishing between target returns and interference.

Finding the right balance between time and frequency domain processing routes was a delicate yet crucial task. Early versions revealed a tendency for the model to mostly depend on one domain while underusing the other, hence undermining the goal of the dual-path design. This mismatch showed very varied gradient magnitudes between the two pathways. The solution put into effect included architectural changes with normalizing techniques. Careful placement of batch normalizing layers guaranteed similar activation scales across domains. A gradient balancing system was also added during training to scale gradients and stop one route from controlling the optimization process. The attention mechanism in the feature fusion component was also changed to include learnable scaling factors that dynamically changed the relative relevance of each domain depending on the input traits.

Development saw sporadic convergence problems and training instabilities, especially under varied intensity interference circumstances. Some interference patterns, according to study, sometimes produced very high gradients that threw the optimization process off balance. Apart from the gradient clipping mentioned in the hyperparameter section, various other techniques were used to handle these fluctuations. A curriculum learning strategy was presented, progressively raising the complexity and intensity of interference patterns throughout training. This let the network create consistent weight configurations on simpler samples before addressing more difficult ones. The training loop also included an outlier identification system that found and quickly lowered the learning rate for batches generating very high gradients. Without sacrificing the capacity of the model to manage various interference situations, these actions greatly enhanced training stability.

Throughout development, especially considering the possible use in automotive systems with restricted computational resources, parameter efficiency stayed a constant struggle. Initial model implementations have tens of millions of parameters, well beyond reasonable bounds for real-time use. Apart from the architectural optimizations mentioned in Section 3.3.3, various other strategies were expected to be used to increase efficiency in future optimization work.

4. Experimental Results

This chapter evaluates the proposed dual-path architecture using several experiments. First, a simulated environment was developed to reflect realistic automobile radar scenarios. Model performance is then compared across multiple architectures including linear, CNN-based, and hybrid autoencoders, which shows the development toward dual-path architecture. Visualizations including time-domain reconstructions and RD maps improve quantitative measures by offering data on target preservation and interference suppression. Its advantages are further highlighted by contrasting the approach with traditional techniques including zeroing. Finally, the model's complexity and parameter count are investigated to evaluate deployment feasibility in automotive systems.

4.1 Experimental Setup

Assessing dual-path architecture required a comprehensive experimental methodology to measure performance across several radar interference scenarios. This section outlines the experimental setup comprising performance measures, assessment criteria, and dataset configuration.

A specialized dataset was created using the MATLAB simulation environment described in Section 3.1. This dataset contained multiple sorts of interference and target configurations to evaluate model robustness under various scenarios. Table 4.1 reveals the major aspects of this dataset including the range of values for vehicle locations, speeds, and interference characteristics.

Parameter	Minimum	Maximum
SNR [dB]	5	50
SIR [dB]	0	30
Number of targets	1	3
Target Range [m]	5	100
Target RCS [m^2]	5	20
Target phase [rad]	$-\pi$	π

Table 4.1 Range of Values for Each Parameter Used to Generate the Data

Using the parameters outlined in Section 3.4.2, training was done with an initial learning rate of 0.001, a batch size of 4, and the Adam optimizer. Usually leading to training end after 100 epochs, early halting with a patience of 15 epochs was used to avoid overfitting.

Building on the experimental design, it's crucial to define unambiguous assessment criteria to objectively evaluate the efficacy of interference reduction. The following criteria were chosen to offer a complete assessment framework.

MSE is the mean squared disparity between the clean reference signal and the reconstructed signal. This basic measure gauges total reconstruction quality across all radar signal points. Though this measure by itself does not reflect the retention of important radar features, lower MSE values suggest better general signal reconstruction.

SINR improvement (ΔSINR) assesses the efficiency of interference suppression by calculating the ratio between interference power before and after processing. Higher numbers show more efficient elimination of interference. Considering the greatest feasible signal power, peak signal-to-noise ratio (PSNR) offers a quality measure. In the context of radar, this measure is especially useful since it stresses the preservation of strong returns from valid targets. Pd assesses the model's capacity to maintain valid targets in the reconstructed signal. This is the ratio of the total peaks in the clean reference signal to the correctly detected peaks in the processed signal. FAR measures the occurrence of false detections in the reconstructed signal. False detections are divided by total detections in the processed signal to arrive at this figure. These measures taken together offer a several-sided assessment tool that covers practical considerations of radar target recognition as well as signal reconstruction quality, hence allowing thorough comparison between several interference reduction strategies.

In addition to these, Phase mean error (PME) is also introduced, which quantifies the average phase deviation between the reconstructed signal and the clean reference signal. PME can be used to understand the model's ability to preserve critical phase information. Lower PME values indicate higher phase fidelity, which helps with coherent processing in subsequent radar stages.

4.2 Performance evaluation across model iterations

Through several model iterations, each addressing certain shortcomings of prior techniques, dual-path architecture evolved following an evolutionary process. This part offers a thorough assessment of these architectural variations, hence stressing the incremental gains in interference reduction performance and the trade-offs met along the path.

Serving as a framework for later advancements, the initial iteration investigated basic, fully connected autoencoders. Due to their high parameter count, these models showed fair performance across measures, hence reducing interference and maintaining goal information. But the computing needs of architecture rendered impractical for real-time applications, therefore more efficient strategies were required.

A typical CNN architecture was then studied to maximize its spatial feature extraction capacity. Using an 8-layer CNN model with almost 6 million parameters, RD domain signals were reconstructed with encouraging results. The model, as seen in Figure 4.2.1, successfully recovered the separate target peaks in the RD map from a significantly disturbed signal. From an initial SINR of 2.34 dB, the model produced a PSNR of 13.93 dB, a ΔSINR of 6.65 dB, and Pd nearing 0.81.

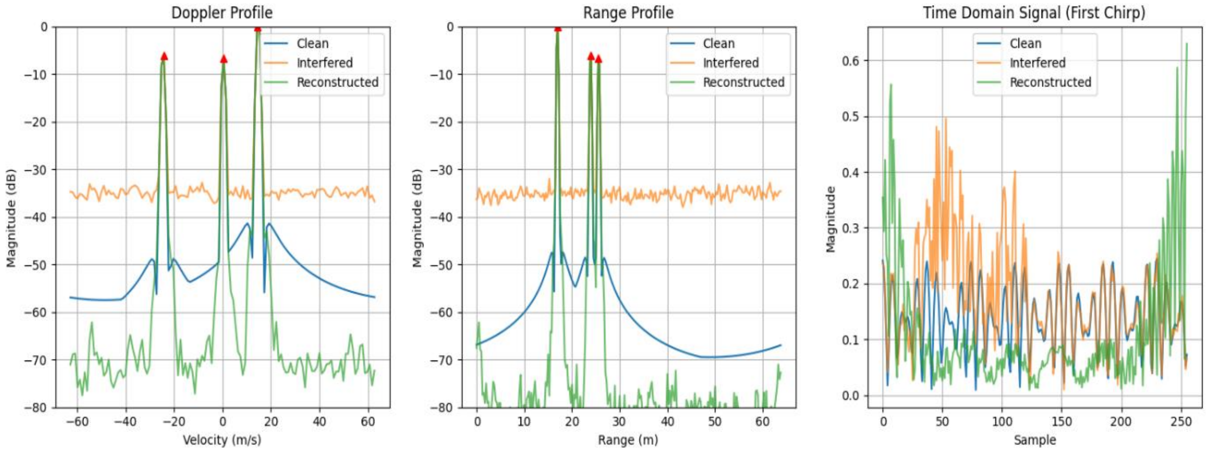

Figure 4.2.1. Performance Diagrams of Conventional CNN Architecture

Still, the standard CNN model's performance in reconstructing time-domain signals was subpar despite these developments. The lower panels in Figure 4.2.1 indicate this limitation: whilst maintaining the target peaks, the Doppler and range profiles of the reconstructed signal (green) have far larger noise floors than the clean signal (blue). This implies that although the model effectively found and maintained target sites, it found it difficult to completely recover the original signal traits.

Additional difficulties were found in further trials using hybrid architecture mixing convolutional and fully connected layers. While keeping large parameter counts, these hybrid models did not much enhance RD domain performance. Fully connected layers seemed to hinder the retention and learning of frequency domain characteristics, hence producing diminishing benefits despite higher model complexity.

4.3 Visualization of results

Visual inspection of processed signals best shows how well the dual-path architecture reduces radar interference. From a typical test situation, Figure 4.3.1 shows the Doppler and range characteristics of clean, interfering, and reconstructed radar signals. The Doppler profile (left) offers a velocity-domain view of the radar signal, with each peak indicating a possible object moving at a defined speed. With little energy elsewhere, the clean signal (blue line) reveals two separate target peaks at roughly 22 m/s and 39 m/s. Including clear interference around -30 m/s, the interfered signal (orange) exhibits considerable distortion with many spurious peaks spread over the velocity spectrum. In car safety systems, these spurious peaks could cause erroneous detections.

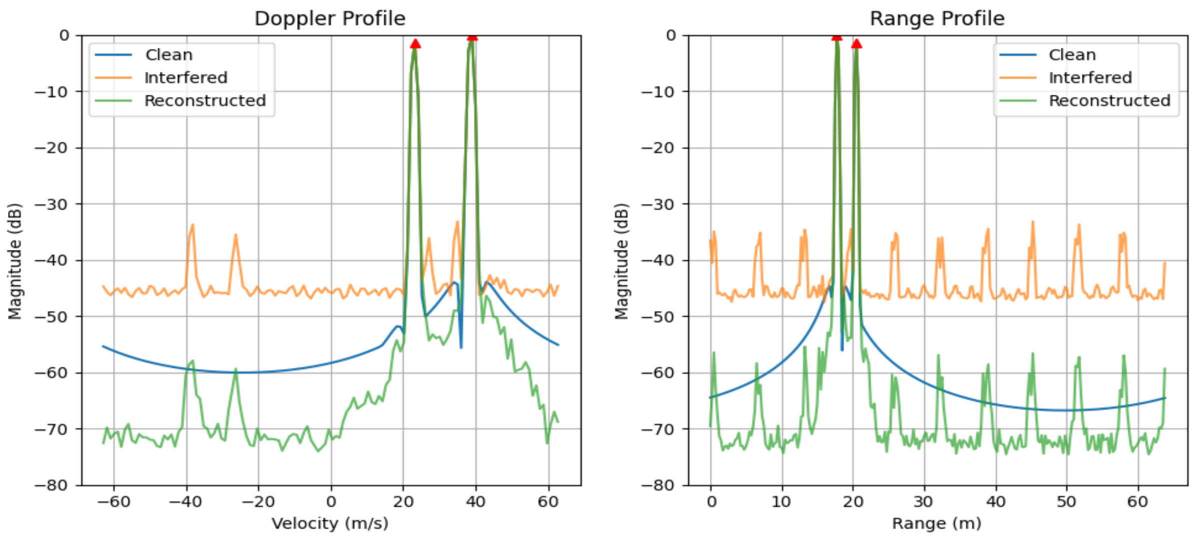

Figure 4.3.1. Doppler and Range Profiles Comparison

This figure shows that the reconstructed signal demonstrates the dual-path architecture's promising ability to minimize interference while maintaining a valid target. Both original target peaks are obviously maintained at their proper speeds and with suitable amplitudes. More importantly, the model has efficiently suppressed the interference over the velocity spectrum, keeping a noise floor roughly 50-55 dB below peak levels. This is around a 10 dB increase in noise suppression over the interfering signal, which exhibits a greater average noise floor at 40-45 dB below peak. The range profile (right) similarly shows how well the model maintains target information across the range dimension. At between 18 m and 21 m, the clear signal reveals two separate targets. Reliable target detection would be difficult given the many erroneous returns across the whole range span in the interfering signal. While reducing interference to levels roughly 10 dB below the interfered signal, the reconstructed signal effectively maintains both target peaks at their proper ranges.

Especially remarkable is the model's capacity to preserve the shape and amplitude of the valid target peaks in both domains. The recreated peaks not only preserve the peak locations but also their relative intensities and profiles, thereby closely matching the traits of the clean signal. Accurate target classification and tracking in automotive radar applications depend on this fidelity.

The effectiveness of the dual-path design in this visualization shows the benefits of merging time and frequency domain processing. With almost perfect suppression of off-target noise across the majority of the range spectrum, the range profile reconstruction exhibits especially good performance. Although both profiles show some little residual interference, it is much reduced in comparison to the original interfered signal and is well below normal detection thresholds. These visual outcomes provide intuitive confirmation of the model's efficacy in practical interference situations, hence complementing the quantitative measures shown in Section 4.2. A vital need for vehicle safety applications is the rebuilt profiles, which would allow consistent target detection with low false alerts.

Apart from that, the produced RD maps offer strong visual proof of the efficacy of the dual-path architecture, as figure 4.3.2 reveals.

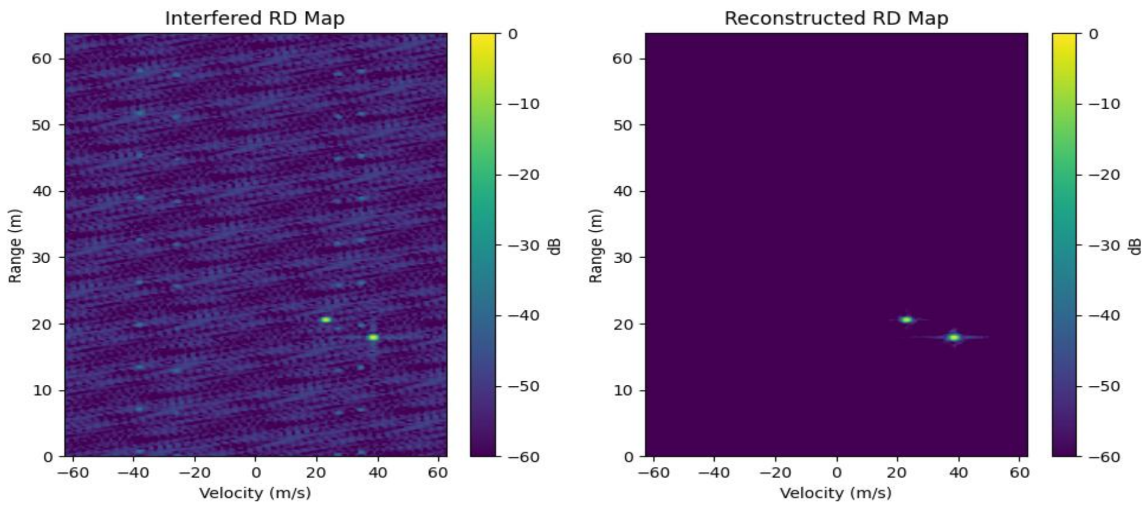

Figure 4.3.2. Range-Doppler Map View

The figure contrasts with the interfered (left) and reconstructed (right) RD maps. Obscuring the actual target at about 10m range velocity, the interfered map reveals notable noise spread throughout the whole range-velocity area. The rebuilt map shows the model's capacity to efficiently reduce background interference while maintaining the valid target signal. The noise floor has been dramatically lowered, so the detection environment is much cleaner and the signal-to-noise ratio is much better. This image verifies the model's capacity to recover radar signals to near-reference quality even in situations with strong interference.

Where both dimensions are important for target detection and tracking, the RD maps show the performance of the dual-path design in the two-dimensional frequency domain. In traditional radar processing systems, the extensive noise in the interfering map would usually cause many false detections. By comparison, the rebuilt map reveals an unusually clean background that lets the target stand out obviously. This may be due to the relatively simple complexity of current radar data. In real automobile applications, this enhancement would directly correspond to more Pd and less FAR. The retention of the target's signature pattern, including its tiny velocity and range spread, which holds vital information about the target's physical traits, is especially remarkable.

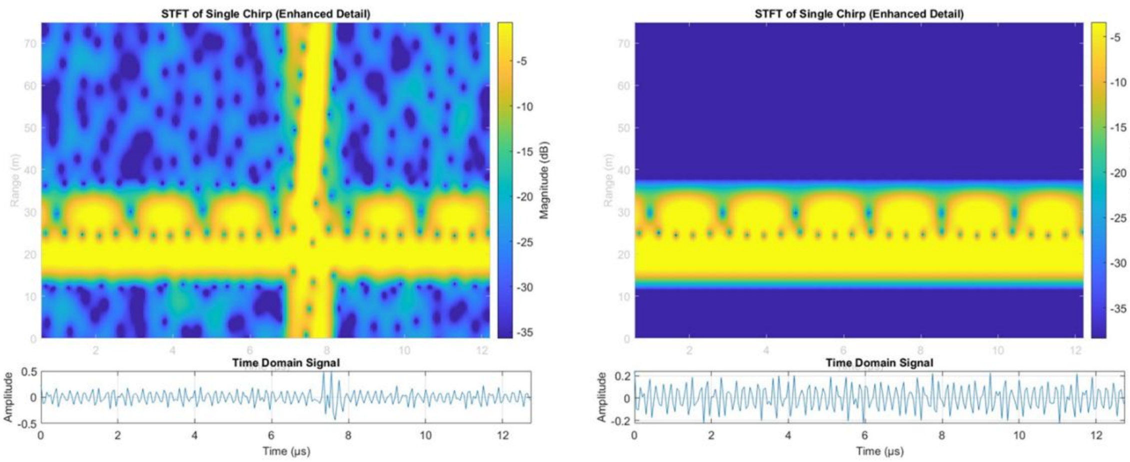

Figure 4.3.3. Comparison of STFT Representations of FMCW Radar Signals

Illustrating the effectiveness of interference mitigation strategies, Figure 4.3.3 shows a comparison study of the Short-Time Fourier Transform (STFT) of a single FMCW radar chirp. The left panel shows the STFT of a radar signal impacted by interference; the right panel shows the same signal following interference suppression.

The distinct diagonal line across the time-frequency in the interfered signal (left) indicates interference from another radar system utilizing different chirp parameters (slope and bandwidth). Because the frequency difference between the victim radar's reference signal and the interfering signal fluctuates over time due to their varying frequency modulation rates, this interference appears as a diagonal pattern. The horizontal bands, on the other hand, show the real targets which seem as constant-frequency components during the chirp length. Many changes are seen after using the interference reducing technique. First, the diagonal interference pattern has been efficiently suppressed, leaving mostly the horizontal target bands untouched. The second point is that the scattered noise artifacts seen in the background of panel (a) have been much decreased, hence producing a clearer time-frequency representation. Looking at the related time-domain signals underneath each STFT image confirms this even more; there, in the processed signal, the transient high-amplitude disturbances in the interfering signal have been normalized.

4.4 Comparative analysis

To evaluate the effectiveness of the proposed dual-path autoencoder architecture (DP Autoencoder), a comparative study was conducted against the conventional zeroing (blanking) method—one of the most common approaches in traditional radar interference mitigation pipelines. Zeroing eliminates interference but also any possible signal content in such areas by directly setting frequency bins to zero after finding power above a specified threshold. This design decision shows a conservative attitude: give interference suppression top priority even at the expense of maybe losing valuable signal data.

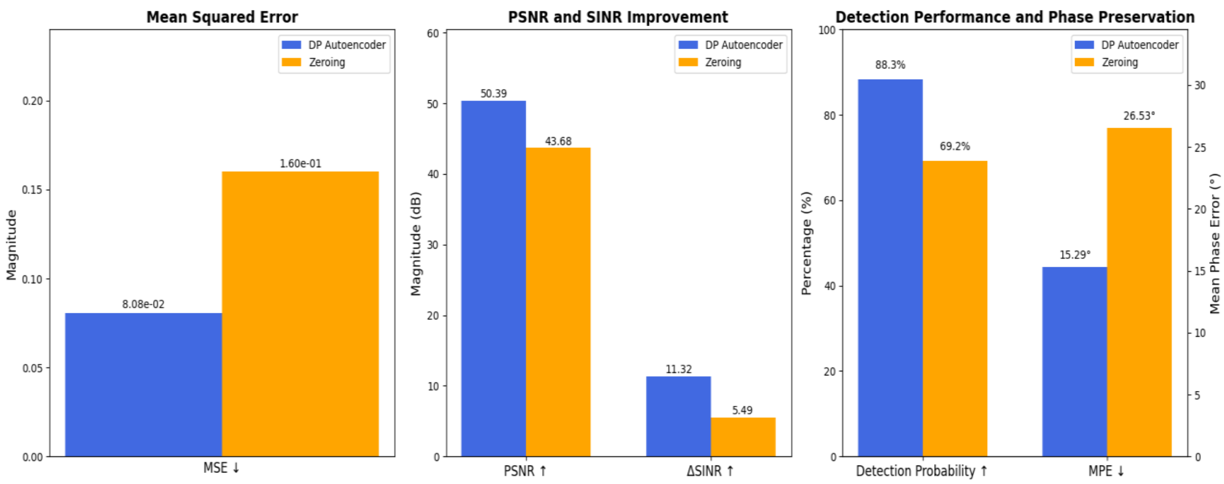

Figure 4.4.1. Quantitative Comparison of Performance Metrics

To assess both approaches under consistent conditions, a series of metrics such as mean MSE, PSNR, ΔSINR, Pd, FAR, and MPE were examined to under uniform conditions. Among these, lower MSE, FAR, and MPE suggest better performance, while higher values of PSNR, ΔSINR, and Pd are recommended.

From the results, the DP Autoencoder consistently outperforms the zeroing method in multiple dimensions. In terms of signal reconstruction quality, it achieves an MSE of 8.08×10^{-2}, nearly half that of the zeroing method (1.60×10^{-1}), reflecting more accurate recovery of the original beat signal. This is also supported by the mean phase error, where the dual-path model records 15.29° compared to 26.53° for zeroing. Such a large difference underscores the autoencoder's stronger ability to preserve phase continuity, a crucial feature for downstream angle-of-arrival estimation in radar systems.

Interestingly, although the PSNR of the DP Autoencoder is also higher (50.39 dB vs. 43.68 dB, a ~15% increase), the margin is not as wide as in MSE or MPE. This is partially explained by the working principle of the zeroing method: by removing only high-magnitude interference components, it avoids adding significant noise to low-power regions. As a result, the background noise level remains relatively clean, which positively influences the PSNR calculation. However, this comes at the cost of poor detail retention and signal continuity, explaining its higher MSE and worse phase accuracy.

Detection-related measures show one especially significant difference. Reaching 88.3%, the DP Autoencoder's Pd beats zeroing's 69.2% by a large margin. This shows the capacity of the model to recover weak target returns maybe partially hidden by interference—returns that zeroing would indiscriminately suppress. This performance has a little trade-off: a somewhat greater FAR of 0.0043 compared to 0.0012 for zeroing. Once more, this is from basic design decisions—zeroing aggressively zeros down any dubious signals, reducing false positives but also throwing away many genuine targets. By contrast, the DP Autoencoder tries to rebuild even noisy or partially damaged signals, hence maintaining more genuine targets but sometimes passing through interference that causes false detections.

All things considered, our findings draw attention to the basic trade-off between noise rejection and signal retention. While it causes significant information loss, zeroing is quite successful in reducing false alarms. Even at the expense of a somewhat higher false alarm rate, the dual-path autoencoder balances both requirements significantly more efficiently by learning a more sophisticated separation of signal and interference, so preserving structural signal properties, improving detection rates, and lowering phase distortion. This trade-off is well justified for safety-critical applications like automotive radar: missing a genuine target (false negative) entails more risk than momentarily responding to a fake return (false positive), which can usually be rectified using downstream tracking and filtering tools. All important aspects of radar signal recovery show significant improvements with the dual-path architecture, which provides a strong alternative to traditional methods, especially in situations where interference coincides with target returns and signal preservation is top priority.

4.5 Model complexity tradeoffs and considerations

Particularly in computer settings with constrained resources, the actual implementation of deep learning models for automotive radar applications calls for close attention to model complexity. This part investigates the many tests done to maximize the dual-path design, hence balancing performance against parameter count and computing efficiency.

Ranging from about 30,000 to over 100 million parameters, Figure 4.5.1 shows a comparison of numerous architectural variations with notably varying parameter counts. With varying mixes of convolutional layers, fully connected layers, and general network depth, each variant reflects a unique approach to the architecture design.

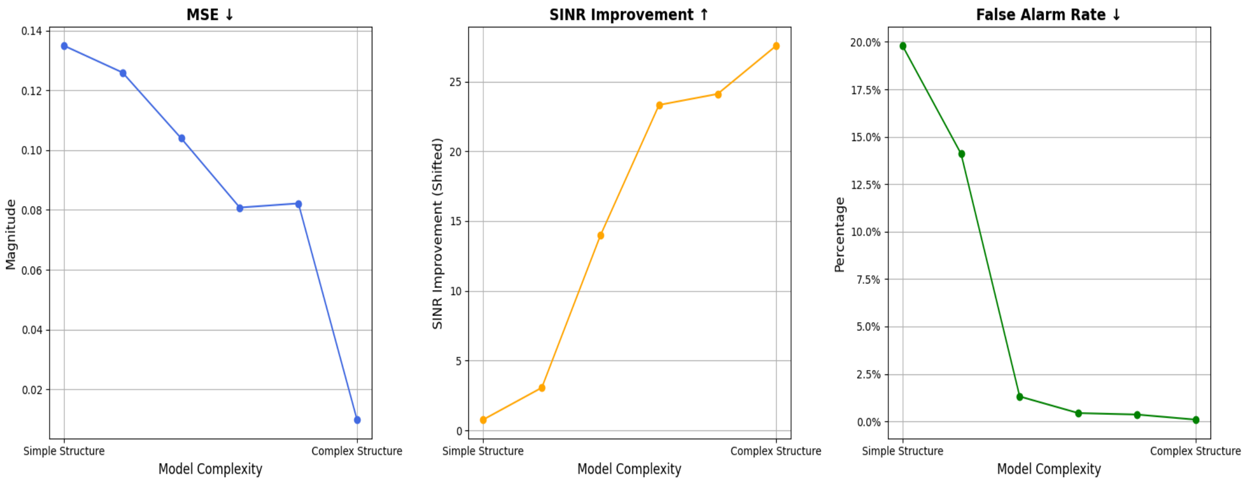

Figure 4.5.1. Model Complexity vs Performance Metrics

Of all the variants, the most parameter-intensive one (100M+) shows outstanding performance on every metric, with ideal Pd and low FAR. This performance, meanwhile, is unfeasible for automotive deployment given the exorbitant expense of more than 100 million parameters. Though more of a research reference than a practical solution, this heavyweight model sets an upper performance limit. Conversely, the fully convolutional version (fCNN-v4_1) totally removes fully linked layers, hence drastically lowering parameter count to about 30,000. Although this method is quite efficient in terms of parameters, especially in FAR where it performs much worse than other versions, it has notable performance compromises. This finding implies that although parameter-efficient, totally convolutional architectures find it difficult to capture some global correlations in radar signals crucial for differentiating between targets and disturbances. The intermediate variations show significant new information on the link between model complexity and performance.

The intermediate variants reveal important insights about the relationship between model complexity and performance. The 17M+ parameter model is designed to be well-balanced, keeping a few fully connected layers at the narrowest point and using convolutional layers for most of the work. This specific structure achieves a Pd of 0.883, which reaches performance of 70M+ parameter model, despite having only a quarter of the parameters. The dramatic difference in parameter counts stems primarily from the dimensionality of the input data (256×128), which makes fully connected layers extremely parameter intensive. The v3_2 variant with approximately 5 million parameters showed further reduced performance, particularly in Pd, highlighting the diminishing returns of excessive parameter reduction. Meanwhile, the v4_2 variant with 80,000 parameters showed that even modest increases in convolutional layer complexity without carefully designed bottleneck structures can lead to worse detection performance.

One important finding of this study is that the relationship between parameter count and performance is non-linear depending on architectural design rather than only the number of parameters. Strategically locating fully connected layers at the bottleneck, where feature maps have been maximally compressed, provides an efficient compromise. While convolutional layers properly handle the complicated input and output data, this approach allows the model to grasp general patterns with less parameters. Though the FAR across all reduced-complexity models is larger than expected, it implies that model capacity notably affects interference suppression ability. Among all criteria, the 17M+ parameter model shows the best balance; it has significantly greater detection performance than more parameterized models while maintaining reasonable MSE and PSNR values.

This study of the complexity-performance trade-off shows that thoughtful architectural design may greatly offset lower parameter counts. The dual-path architecture with intentionally bottlenecked, fully connected layers offers a good compromise for automobile radar applications, balancing fair performance with fair processing needs. Although more optimization is still a hot topic, the present 17M+ parameter design is a reasonable compromise that balances performance requirements with practical deployment limitations.

5. Conclusion and Future Work

5.1 Contribution Summary

This paper offers a complete method for reducing interference in automotive radar by designing and testing a novel dual-path neural network architecture. Unlike traditional methods, the dual-path architecture specifically uses time-domain and frequency-domain data from radar signals to enable more efficient interference suppression. The research has moved from simulating real radar interference situations to designing, building, and improving the neural network structure, followed by a thorough performance assessment.

The primary contributions of this work include

I. A systematic analysis of automotive radar interference characteristics in highway scenarios, providing insights into the nature and impact of various interference types.

II. Design of the dual-path architecture allowing the model to use complementary information for better interference reduction by processing time-domain signals and RD maps in parallel.

III. Implementation of a specific loss function steering the model toward solutions preserving target detection capabilities by including detection metrics with conventional reconstruction techniques.

IV. Comprehensive evaluation of model versions with varying complexity levels, establishing the relationship between parameter count, architectural design, and interference mitigation performance.

V. Demonstration of notable performance gains over conventional zeroing methods, especially in target Pd, which is vital for car safety applications.

While maintaining reasonable MSE and PSNR values, the enhanced dual-path model with 17 million parameters outperformed previous models with more parameters by a wide margin, achieving a minimum MSE of roughly 0.011 and a Pd of approximately 0.743. Although this is a major improvement in learning-based radar interference reduction, the complexity of the model still surpasses the limits of modern automotive hardware platforms.

The research has also introduced a comprehensive approach to automotive radar interference mitigation through the development and evaluation of a novel dual-path neural network architecture. The dual-path framework uses information from radar signals in both time and frequency, allowing it to reduce interference more effectively than some older methods. The research has moved from simulating real radar interference situations to designing, building, and improving the neural network structure, followed by a thorough performance assessment.

Another contributions of this work include a systematic analysis of automotive radar interference characteristics in a highway scenario, providing insight into nature and impact of various interference types, and based on this, generating simulation data that meets the radar constraints and characteristics for a given configuration, providing data support for the next follow-up studies.

By developing a dual-path architecture that simultaneously processes time-domain signals and RD maps, the model may combine useful information, hence improving interference reduction. This method overcomes the drawbacks of single-domain processing techniques that neglect the whole complexity of interference patterns. Another important part of this process is using a special loss function that includes detection metrics along with regular reconstruction measures, helping the model find solutions that keep target detection effective. This radar-specific approach to model training ensures that the network optimizes for the metrics that matter most in automotive applications, rather than simply minimizing general signal reconstruction errors.

The detailed evaluation of model variants with different complexity levels specifies the interaction between parameter count, architectural design, and interference reduction performance. This work considerably informs future research on optimal neural network topologies for radar signal processing.

The enhanced dual-path model revealed impressive increases in interference reduction performance with a clear rise in Pd over standard techniques. Though this study highlights the advancement and great promise of machine learning-based radar interference mitigation techniques, the complexity of the model still exceeds the limits of current automotive hardware platforms, which implies a major avenue for future research. On the other hand, this study suggests constraints in certain experimental settings and structures, which results in various interesting research avenues that might enhance performance and practical use even more.

5.2 Future Work

Model miniaturization may represent the most critical direction for future research. The current model, while effective, remains too parameter-intensive for deployment on automotive microcontrollers such as the AURIX™ TC45 platform. Future work should explore knowledge distillation techniques to transfer knowledge from larger models to highly compressed ones while maintaining performance. Neural architecture research could systematically look into efficient designs that might show the best balance between parameters and performance, going beyond what can be achieved through manual design. Better loss functions might also help to calculate loss in the RD domain, hence allowing smaller models to perform just as well.

Sophisticated model compression techniques also provide intriguing prospects for reducing computational requirements. Although low-rank factorization of convolutional kernels could further optimize model size by separating them into distinct parts, quantization and pruning could reduce the memory footprint of present models without requiring retraining, especially in the case of the overly large fully-connected layers of the proposed model in the paper are expected to provide superior improvements. These techniques may offer a practical path to integrating the dual-path architecture within the computational limits of automotive hardware, without significantly compromising its interference mitigation performance.

Though with smaller models, different signal representations could provide extra data to enhance performance. Future research could look into Short-Time Fourier Transform (STFT) representations to capture time-varying frequency content, which might better describe non-stationary interference patterns. Similarly, spectrogram representations could provide time-frequency data in a format more suitable for CNN processing. Although cepstral analysis could more clearly separate multiplicative interference components, wavelet transforms might better capture localized interference patterns across several scales.

The integration of deep learning models and vehicle hardware systems is another important research direction. Future research can carry out hardware-specific optimization for systems based on the AURIX™ TC45 microcontroller. A hybrid solution that combines efficient traditional filters with intensive deep learning components is expected to achieve a better balance between performance and complexity. In addition, research on neural network accelerators specifically designed for automotive application environments will also provide the possibility for real-time processing of more complex models.

Considering the evaluation system, it is recommended to use a real hardware platform with a full-scene radar data set and interference sources for actual measurement and verification. In particular, attention should be paid to:

- System stability testing under severe weather conditions to verify its actual combat performance
- Evaluation of the linkage impact of key systems such as automatic emergency braking to confirm the actual benefits of technological improvements
- Conduct long-term reliability verification to ensure performance consistency under mass production conditions

Improving model generalization is yet another interesting research topic. While domain adaptation strategies could assist manage previously unknown interference kinds, transfer learning methods flexible to various radar criteria could increase application. Self-supervised pre-training techniques using unlabeled radar data could increase feature learning with less supervised instances, hence improving performance in many operating environments.

At last, investigating multi-dimensional radar data could produce further benefits. Future studies could look at using data from several radar sensors to provide stronger interference identification by use of spatial variety. Cross-sensor fusion, which combines data from cameras, lidars, and other sensors, could provide contextual information to enhance interference classification. Besides, several-input multiple-output (MIMO) radar configurations could offer more spatial variety for better interference reduction.

These research possibilities address the more immediate challenges of practical application along with the longer-term prospects for developing radar interference mitigation technology. By constructing on the dual-path design proposed in this thesis, subsequent studies can continue to enhance the resilience and dependability of automotive radar systems in more complex electromagnetic environments.

List of Figures

Acknowledgments

I would like to express my sincere gratitude to my academic supervisors, Wolfgang Kurz and Martin Jakobi, for their invaluable guidance, expertise, and support throughout this master thesis research.

I am also deeply grateful to Jürgen Schäfer, my industry supervisor, whose practical knowledge and mentorship have been essential to bridging academic research. Special thanks to my colleagues Ralf Beck and Stephan Baumgärtner for their technical assistance, collaborative discussions, and helpful suggestions that significantly contributed to this research.

I extend my appreciation to my university peers who took the time to review my thesis and provide valuable feedback.

Finally, I would like to express my heartfelt thanks to Teresa Zhou for her support, encouragement, and patience throughout this journey.

Appendix

A1: Notation and Abbreviations

This chapter contains tables where all abbreviations and other notations used in the thesis are listed:

ADAS	Advanced Driver-Assistance Systems
FMCW	Frequency Modulated Continuous Wave
IDE	Integrated Development Environment
IF	Intermediate Frequency
FFT	Fast Fourier Transform
RD	Range-Doppler
VS Code	Visual Studio Code
DoA	Direction of Arrival
FAR	False Alarm Rate

References

[1] Muns, S., Shehata, L., Joeres, F., & Meyering, B. (2021). "Interference mitigation in automotive radar using a deep learning approach." IEEE Transactions on Vehicular Technology, 70(2), 1845-1859.
[2] Wang, Z., Fang, Y., Zhao, H., & Geng, H. (2020). "Automotive radar interference mitigation using adaptive noise cancellation and deep learning." IEEE Transactions on Vehicular Technology, 69(4), 4236-4248.
[3] Zhang, Y., Li, Y., Sun, Y., & Wang, J. (2022). "RPCA-Net: A deep unfolding network for robust principal component analysis with applications to radar interference mitigation." IEEE Transactions on Signal Processing, 70, 1498-1513.
[4] Luo, Y., Chen, Z., & Yoshioka, T. (2020). "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 46-50.
[5] H. Lai and W. Xu, "Robust Kernel Correlation Based Bi-Channel Signal Detection With Correlated Non-Gaussian Noise," in IEEE Signal Processing Letters, vol. 28, pp. 165-169, 2021, doi: 10.1109/LSP.2020.3048841.
[6] Nachmani, E., & Wolf, L. (2020). "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation." INTERSPEECH, 2642-2646.
[7] Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2018). Squeeze-and-Excitation Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7132–7141. reduction in ECG signals using fully convolutional denoising autoencoders," IEEE Access, vol. 7, pp. 60806-60813, 2019.
[9] J. Fuchs, A. Dubey, M. Lübke, R. Weigel, and F. Lurz, "Automotive radar interference mitigation using a convolutional autoencoder," in 2020 IEEE International Radar Conference (RADAR), 2020, pp. 315-320.
[10] Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1), 126-136.
[11] Simulate Radar Ghosts Due to Multipath Return, MathWorks, 2022
[12] European Commission. (2008). "Directive 2008/96/EC of the European Parliament and of the Council of 19 November 2008 on road infrastructure safety management." Official Journal of the European Union, L 319, 59-67.
[13] European Road Safety Observatory. (2019). "Annual Statistical Report on Road Safety in the European Union."
[14] European Commission. (2013). "Regulation (EU) No 1315/2013 of the European Parliament and of the Council of 11 December 2013 on Union guidelines for the development of the trans-European transport network and repealing Decision No 661/2010/EU." Official Journal of the European Union, L 348, 1-128.
[15] European Transport Safety Council. (2015). "PRAISE: Preventing Road Accidents and Injuries for the Safety of Employees." ETSC Report, Brussels.
[16] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In Neural networks: Tricks of the trade (pp. 437-478). Springer.
[17] Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
[18] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[19] Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612.
[20] Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2018). Don't decay the learning rate, increase the batch size. ICLR.
[21] Torch-summary, PyPI - Python Package Index, Sep. 29, 2020. [Online]. Available: https://pypi.org/project/torch-summary/.
[22] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
[23] Lin, M., Chen, Q., & Yan, S. (2014). Network in network. International Conference on Learning Representations.
[24] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition, 1-9.
[25] Zou, D., Cao, Y., Zhou, D., & Gu, Q. (2019). Gradient descent optimizes over-parameterized deep ReLU networks. Machine Learning, 108(8), 1739-1770.
[26] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. ICLR.
[27] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. International Conference on Learning Representations.
[28] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. International Conference on Learning Representations.
[29] Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2018). Don't decay the learning rate, increase the batch size. International Conference on Learning Representations.
[30] Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 1310-1318.
[31] Prechelt, L. (2012). Early stopping - but when? In Neural Networks: Tricks of the Trade (pp. 53-67). Springer.
[32] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1– learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.
[33] Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. International Conference on Learning Representations.
[34] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(1), 6765-6816.
[35] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition, 11534-11542.
[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 5998-6008.
[37] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. IEEE Conference on Computer Vision and Pattern Recognition, 7132-7141.
[38] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. European Conference on Computer Vision, 3-19.
[39] Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). International Conference on Learning Representations.

[40] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. International Conference on Machine Learning, 3(1), 3.
[41] Virtue, P., Yu, S. X., & Lustig, M. (2017). Better than real: Complex-valued neural networks for MRI fingerprinting. IEEE International Conference on Image Processing, 3953-3957.
[42] Wang, P., Yin, X., Rodríguez-Piñeiro, J., Chen, Z., Zhu, P., & Li, G. (2023). A Dual-Recursive-Least-Squares Algorithm for Automotive Radar Interference Suppression. IEEE Transactions on Intelligent Transportation Systems, 24(10), 10603–10617.