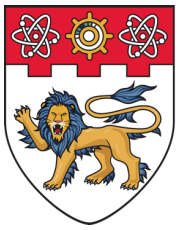
DEEP LEARNING TECHNIQUES FOR ROBUST NON-COOPERATIVE SPECTRUM SENSING IN COGNITIVE RADIO NETWORKS



NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Su Zhengyang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

DEEP LEARNING TECHNIQUES FOR ROBUST NON-COOPERATIVE SPECTRUM SENSING IN COGNITIVE RADIO NETWORKS

Su Zhengyang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

12-12-2024

Date

Su Zhengyang

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

12-12-2024

Date

Associate Prof. Teh Kah Chan

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journals / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as **Zhengyang Su**, Kah Chan Teh, Sirajudeen Gulam Razul, and Alex Chichung Kot, "Deep non-cooperative spectrum sensing over Rayleigh fading channel," IEEE Transactions on Vehicular Technology, vol. 71, no. 4, pp. 4460-4464, Apr. 2022.

The contributions of the co-authors are as follows:

- A/Prof Teh provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by A/Prof Teh, Prof. Kot and Dr. Razul.
- I co-designed the spectrum sensing system and deep learning network with A/Prof Teh.
- I performed all the numerical experiments with code programming and data analysis at the Center of Information Sciences and Systems (CISS).

Chapter 4 is published as **Zhengyang Su**, Kah Chan Teh, Yihang Xie, Sirajudeen Gulam Razul, Alex Chichung Kot, "Signal enhancement aided end-to-end deep learning approach for joint denoising and spectrum sensing," IEEE Transactions on Vehicular Technology, vol. 73, no.3, pp. 4424 - 4428, Mar. 2024.

The contributions of the co-authors are as follows:

- A/Prof Teh provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by A/Prof Teh, Prof. Kot and Dr. Razul.
- I co-designed the spectrum sensing system and deep learning network with A/Prof Teh.
- I co-designed the fading channel model and OFDM system with Dr. Xie.
- I performed all the numerical experiments with code programming and data analysis at the Center of Information Sciences and Systems (CISS).

Chapter 5 is published as **Zhengyang Su**, Kah Chan Teh, Sirajudeen Gulam Razul, and Alex Chichung Kot, "Robust channel invariant deep non-cooperative spectrum sensing," IEEE Wireless Communications Letters, vol. 12, no. 3, pp. 436-440, Mar. 2023.

The contributions of the co-authors are as follows:

- A/Prof Teh provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by A/Prof Teh, Prof. Kot and Dr. Razul.
- I co-designed the spectrum sensing system and deep learning network with A/Prof Teh.
- I performed all the numerical experiments with code programming and data analysis at the Center of Information Sciences and Systems (CISS).

Chapter 6 is published as **Zhengyang Su**, Kah Chan Teh, Yihang Xie, Sirajudeen Gulam Razul, Alex Chichung Kot, "DC4S: A dual-contrast self-supervised learning framework for robust and efficient spectrum sensing with limited labeled data," IEEE Transactions on Cognitive Communications and Networking (Early Access), DOI: 0.1109/TCCN.2025.3544400.

The contributions of the co-authors are as follows:

- A/Prof Teh provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by A/Prof Teh, Prof. Kot and Dr. Razul.
- I co-designed the spectrum sensing system and deep learning network with A/Prof Teh.
- I co-designed the fading channel model and OFDM system with Dr. Xie.
- I performed all the numerical experiments with code programming and data analysis at the Center of Information Sciences and Systems (CISS).

12-12-2024

Date

Su Zhengyang

Acknowledgements

This research is supported by the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. I am profoundly indebted to my supervisor A/Prof. Teh Kah Chan, for his invaluable guidance, unwavering encouragement, and expert technical support throughout this endeavor. I also extend my sincere gratitude to Prof. Alex C. Kot and Dr. Sirajudeen s/o Gulam Razul for their insightful contributions and diverse perspectives, which have significantly enriched this work. I am also deeply grateful to Associate Professor Wen Bihan and Dr. Satoshi Tsutsui for the opportunity to participate in their medical AI-related research group. Despite the challenges of balancing this work with my primary focus on spectrum sensing, their mentorship has broadened my research horizons and enhanced my interdisciplinary skills, which have proven instrumental in my overall doctoral journey.

I wish to express my appreciation to the members of the Centre for Information Sciences and Systems for their collegial support and intellectual stimulation, which have been instrumental in the completion of this thesis.

My heartfelt thanks go to my family for their constant love, encouragement, and understanding. Their unwavering support and sacrifices have been the foundation of my academic pursuits, and I am deeply grateful for their patience and faith in me.

I am particularly thankful to my girlfriend, who is also pursuing her doctoral degree. Our shared experiences as Ph.D. candidates have fostered a deep mutual understanding and respect. Her presence has been a source of strength and motivation, helping me maintain balance and perspective in my academic endeavors.

I would like to acknowledge my friends and colleagues for their camaraderie, intellectual discourse, and moral support. Their diverse backgrounds and perspectives

have not only enriched my research but have also contributed to my personal growth.

I am grateful to every individual I have encountered during my doctoral studies, particularly those who supported me through my baptism. This profound experience has deepened my faith and added a meaningful dimension to my personal development alongside my academic growth.

In conclusion, I extend my deepest appreciation to all who have contributed to my doctoral journey. The knowledge gained, relationships forged, and experiences shared during this transformative period will be cherished always.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

—Alan Turing

Abstract

The objective of this thesis is to exploit recent advances in deep learning-based techniques for spectrum sensing in cognitive radio networks. Firstly, we propose a deep learning-based approach for denoising and non-cooperative spectrum sensing under Rayleigh fading channels, where the model learns to sequentially denoise the signal and perform spectrum sensing, thereby improving detection accuracy, particularly under low signal-to-noise ratio (SNR) conditions. However, as the detection performance is highly affected by the denoising stage, we introduce a joint learning strategy that simultaneously performs denoising and spectrum sensing for orthogonal frequency-division multiplexing (OFDM) systems, optimizing the trade-off between the two tasks and further enhancing performance. Following that, to enhance the robustness and adaptability of spectrum sensing in dynamic wireless environments, we develop a contrast learning-based supervised spectrum sensing technique that extracts channel-invariant features, enabling the model to adapt to new environments without extensive retraining. To tackle the issue of limited labeled data, we extend our work by proposing a dual contrast self-supervised learning-based spectrum sensing (DC4S) framework. This approach leverages unlabeled data to learn meaningful representations, reducing the dependency on labeled samples and enabling effective spectrum sensing even in scenarios where labeled data are scarce. Additionally, it is designed to be robust against different sub-types of tapped-delay line (TDL) channels and certain OFDM signal hyper-parameters. Overall, the proposed methods progressively address the challenges of high-level noise, dynamic environments, and limited labeled data. This thesis illustrates the potential of deep learning-based techniques in enhancing the performance and robustness of spectrum sensing in cognitive radio networks, contributing to the development of more efficient and reliable wireless communication systems.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathbb{C}^{a\times b}$	Complex-valued matrix space of size $a \times b$
$\mathbb{R}^{a\times b}$	Real-valued matrix space of size $a \times b$
$f(\,\cdot\,)$	Network mapping function
$\mathrm{P_r}\{\,\cdot\,\}$	Probability of an event
$\lvert\,\cdot\,\rvert^2$	Squared magnitude
$\lvert\,\cdot\,\rvert$	Absolute value or magnitude
a^T	Transpose of matrix a
$\prod_k^B$	Product over index k from 1 to B
$\sum_k^B$	Sum over index k from 1 to B
$\forall$	For all
$\mathrm{floor}(\,\cdot\,)$	Floor function
$O(\,\cdot\,)$	Big O notation
$\int_{-\infty}^{+\infty} du$	Integral over the real line
$h^*(\,\cdot\,)$	Complex conjugate of function h
$\triangleq$	Defined as
$\circ$	Function compostion
$\rightarrow$	Maps to
$\lvert\lvert\,\cdot\,\rvert\rvert_F$	Frobenius norm
$\lvert\lvert\,\cdot\,\rvert\rvert_1$	L1 norm
$\lvert\lvert\,\cdot\,\rvert\rvert_2$	L2 norm
$\arg\min_{\boldsymbol{\theta}e}$	Argument that minimizes w.r.t. $\boldsymbol{\theta}_e$
$\sim$	Distributed as
$CN(a, b)$	Complex normal distribution
$\cap$	Intersection

$\emptyset$	Empty set
$\mathfrak{Re}\{\cdot\}$	Real part of a complex vector
$\mathfrak{Im}\{\cdot\}$	Imaginary part of a complex vector
$\mathbb{1}$	Indicator function
$\mathcal{N}_i$	the index set of the neighbors of agent i

Acronyms

5G	Fifth Generation
Adam	Adaptive Moment Estimation
ANN	Artificial Neural Network
AP	Affinity Propagation
BCED	Blindly Combined Energy Detection
CAE	Convolutional Auto-Encoder
CAV	Covariance Absolute Value
CDF	Cumulative Distribution Function
CLDNN	Convolutional Long Short-term Deep Neural Network
CNN	Convolutional Neural Network
CP	Cyclic Prefix
CRN	Cognitive Radio Network
CSCG	Circularly Symmetric Complex Gaussian
CSI	Channel State Information
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
ED	Energy Detection
FC	Fully-Connected Layer
FFT	Fast Fourier Transform
FLOPs	Floating-Point Operations
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
i.i.d	Independent and Identically Distributed
IFFT	Inverse Fast Fourier Transform
KS	Kolmogorov-Smirnov

LoS	Line-of-Sight
LSTM	Long-Short Term Memory
LTE-M	LTE-Machine-to-Machine
MAP	Maximum A Posteriori
MED	Maximum-Eigenvalue Detection
MFD	Matched-Filter Detection ML
Machine Learning	
MLP	Multi-Layer Perceptron
MRC	Maximum Ratio Combining
MSA	Multi-head Self-Attention
MSE	Mean-Square Error
NLoS	Non-Line-of-Sight
NLP	Natural Language Processing
NR	New Radio
NT-Xent	Normalized Temperature-scaled Cross-Entropy Loss
NU	Noise Uncertainty
OFDM	Orthogonal Frequency-Division Multiplexing
PDF	Probability Density Function
PSNR	Peak Signal-to-Noise Ratio
PU	Primary User
QoS	Quality of Service
ReLU	Rectified Linear Unit
ResNet	Residual Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SA	Self-Attention
SCAE	Stacked Convolutional Auto-Encoder
SCF	Spectral Correlation Function
SNR	Signal-to-Noise Ratio
SPET	Simplified Predicted Eigenvalue Threshold
SSIM	Structural Similarity Index Measure
STFT	Short-Time Fourier Transform
SU	Secondary User
SVM	Support Vector Machine
TDL	Tapped-Delay Line

UAV	Unmanned Aerial Vehicle
VAE	Variational Auto-Encoder

Chapter 1

Introduction

1.1 Motivation

The rapid growth of wireless devices and services has led to an unprecedented demand for spectrum resources. However, the traditional static spectrum allocation policy has resulted in spectrum scarcity and underutilization. The cognitive radio network (CRN) has emerged as a promising paradigm to address this issue by enabling dynamic spectrum access [1]. In CRNs, secondary users (SUs) can opportunistically access the licensed spectrum bands when they are not being used by the primary users (PUs), who are the licensed users of the spectrum, significantly improving spectrum utilization efficiency [2].

Spectrum sensing is a crucial component of CRNs, as it allows SUs to detect the presence or absence of PUs in a given frequency band. Accurate and robust spectrum sensing is essential to ensure the seamless coexistence of PUs and SUs and to prevent harmful interference [3–5]. Non-cooperative spectrum sensing, where SUs independently perform sensing without collaboration, is particularly challenging due to factors such as low signal-to-noise ratio (SNR), dynamic channel conditions. For methods that rely on supervised learning, the need for labeled data further complicates the development of robust sensing frameworks. While many traditional spectrum sensing methods can operate without labeled data, addressing this limitation is critical for advancing deep learning-based approaches, which form the core of this thesis.

Traditional spectrum sensing methods, such as energy detection (ED) [6] and cyclostationary feature detection [7], have been extensively studied. However, these methods have several limitations. ED suffers from performance degradation in low SNR regimes and is sensitive to noise uncertainty (NU) [8, 9]. Cyclostationary feature detection requires prior knowledge of PU signals and has high computational complexity [10]. Moreover, these methods rely on hand-crafted features and require careful parameter tuning, which limits their sensing performance.

Machine learning (ML) and deep learning (DL) have recently emerged as powerful tools for spectrum sensing, offering the potential to overcome the limitations of traditional methods [11, 12]. ML offers substantial benefits by optimizing sensing strategies through data-driven insights, while DL facilitates the automatic extraction of pertinent features from raw data, thereby enhancing the accuracy and robustness of sensing decisions. By leveraging deep neural network (DNN), it is possible to learn complex mappings between the received signals and the corresponding spectrum occupancy status, even in challenging environments. Various DL architectures, such as DNN, convolutional neural networks (CNNs) [13], recurrent neural networks (RNNs) [14], have been extensively explored for spectrum sensing. Each architecture brings distinct advantages, such as capturing spatial dependencies with CNNs and temporal dependencies with RNNs, thus contributing to the overall effectiveness and adaptability of spectrum sensing systems.

Despite the promising potential of DL for spectrum sensing, several challenges remain. A significant issue is the presence of high-level noise in received signals, which can substantially degrade sensing performance. Although denoising techniques based on DL have been proposed to mitigate the impact of noise [15–17], their effectiveness in the context of spectrum sensing has not been verified. Additionally, manual feature extraction methods, such as energy statistics and covariance matrices, may limit the performance of deep learning networks. There is a critical need to develop DL methods that operate on raw IQ data and perform effectively under low-SNR scenarios.

Another challenge is the performance degradation of DL-based spectrum sensing models when applied to new environments or channel conditions that differ from the training data. This domain shift problem necessitates the retraining of models using labeled data from the new environment, which is often impractical in real-world scenarios. To address this issue, techniques such as transfer learning

and modified channel-invariant feature extraction modules are being explored to mitigate sensitivity to channel condition shifts and improve model generalization.

Furthermore, supervised DL-based spectrum sensing methods typically require large amounts of labeled training data, which can be costly and time-consuming to obtain, especially in dynamic and complex environments. Self-supervised learning techniques, which leverage unlabeled data to learn meaningful representations, have shown promise in reducing the reliance on labeled data. Exploring the effectiveness of self-supervised learning frameworks for spectrum sensing and its potential to enhance performance with limited labeled data are also an important research direction.

In summary, DL has emerged as a powerful paradigm for spectrum sensing, offering the potential for accurate, robust, and adaptable detection of PUs in CRNs. This thesis aims to contribute to this research area by proposing novel solutions to the aforementioned problems. By developing DL-based denoising and sensing techniques, joint optimization frameworks, domain-generalized models, and self-supervised learning approaches, this research seeks to enhance the accuracy, reliability, and adaptability of non-cooperative spectrum sensing. The proposed techniques have the potential to enable more efficient and effective spectrum utilization, ultimately benefiting the ever-growing wireless communication systems.

1.2 Objectives

This thesis aims to address the challenges and limitations of non-cooperative spectrum sensing in CRNs by leveraging advanced DL techniques. The primary objectives of this research are as follows:

- **DL-based two-stage non-cooperative spectrum sensing:** The objective is to propose a DL-based non-cooperative spectrum sensing approach under Rayleigh fading channel that overcomes several limitations of existing traditional and DL-based methods, and obtains superior detection performance.
- **Joint optimization framework for denoising and spectrum sensing in OFDM systems:** The objective is to design a unified DL model that

simultaneously learns to denoise the received signals and perform spectrum sensing, thereby improving the overall sensing performance and reducing the dependency on the quality of the denoising stage. The proposed system will be evaluated using orthogonal frequency division multiplexing (OFDM) signals with tapped-delay line (TDL)-E channel.

- **Channel invariant spectrum sensing over unknown environments:** The objective is to investigate the use of modified loss functions to enable the model to extract channel-invariant features and maintain its performance across different channel conditions. The proposed approach aims to enhance the model's robustness and generalizability under different channel conditions.

- **Self-supervised learning for spectrum sensing with limited labeled data:** The objective is to develop a self-supervised pre-learning strategy that leverages unlabeled data to learn meaningful representations of OFDM communication signals. The effectiveness of pre-training the model using self-supervised learning techniques and fine-tuning it with a limited amount of labeled data will be evaluated and compared to fully supervised learning approaches.

By achieving these objectives, this thesis seeks to contribute significantly to the development of robust and efficient non-cooperative spectrum sensing techniques using deep learning. The proposed methods are expected to enhance the accuracy, adaptability, and practicality of spectrum sensing in CRNs, ultimately leading to improved spectrum utilization and management.

1.3 Contributions and Thesis Organization

The main contributions of this thesis are as follows:

DL-based two-stage non-cooperative spectrum sensing. Existing traditional and DL-based spectrum sensing methods suffer from various limitations such as low accuracy in low SNR environments, high computational complexity, and vulnerability to noise uncertainty. To overcome these limitations, a novel two-stage spectrum sensing method is proposed that combines DL-based denoising stage and

spectrum sensing stage. The proposed approach effectively mitigates the impact of noise on the sensing decision by reconstructing the noisy signals through a stacked convolutional auto-encoder (SCAE). Numerical results show that the probability of detection at a fixed probability of false alarm has great improvement.

Joint optimization framework for denoising and spectrum sensing in OFDM systems. Unlike previous DL-based two-stage approaches, a joint optimization framework for denoising and spectrum sensing is introduced, which simultaneously learns to denoise the received signals and perform spectrum sensing using a unified DL model. By jointly optimizing the denoising and sensing tasks, sensing performance is improved and the dependency on the quality of the denoising stage is reduced.

Channel invariant spectrum sensing over unknown environments. The domain shift problem in DL-based spectrum sensing is addressed by developing an environment-robust DL model. The proposed model is designed to extract channel-invariant features, enabling it to shorten the gaps of its performance between training environments and testing environments, without the need for extensive retraining. This contribution enhances the practicality and scalability of DL-based spectrum sensing in real-world CRNs.

Self-supervised learning for spectrum sensing with limited labeled data. To explore the scenario of spectrum sensing where labeled data are scarce, a self-supervised pre-training framework called DC4S is proposed, which leverages unlabeled data to learn meaningful representations of OFDM-QPSK signals. By pre-training the model using self-supervised learning techniques and fine-tuning it with a limited amount of labeled data, competitive performance can be achieved compared to fully supervised learning approaches.

The remainder of this thesis is organized as follows.

- **Chapter 2** presents a comprehensive literature review of spectrum sensing techniques, with both traditional methods and DL-based methods. It discusses the state-of-the-art approaches and identifies the research gaps that this thesis aims to address.

- **Chapter 3** introduces the proposed two-stage spectrum sensing method, which combines DL-based denoising and spectrum sensing techniques, including the architecture, methodology, experimental setup, and results, demonstrating the effectiveness of the proposed approach in mitigating the impact of noise on the sensing decision.

- **Chapter 4** describes the joint optimization framework for denoising and spectrum sensing in OFDM systems, including the encoder-decoder architecture, modified loss function, experimental results and analysis.

- **Chapter 5** focuses on the domain shift problem in DL-based spectrum sensing. It includes the network architecture, channel invariant feature extraction methodology, experimental results, and analysis across different testing channel conditions without the need for extensive retraining.

- **Chapter 6** demonstrates the details of self-supervised pre-training framework for spectrum sensing in scenarios where labeled data are scarce. It includes the proposed DC4S structure, methodology, training process, and analysis for experimental results.

- **Chapter 7** draws a conclusion for the thesis and introduces potential future research directions.

Chapter 2

Literature Review

2.1 Spectrum Sensing and Cognitive Radio Networks

The CRN has emerged as a promising paradigm to address the increasing demand for wireless spectrum resources and alleviate spectrum scarcity. The concept of cognitive radio, first introduced by Mitola and Maguire [1], envisions intelligent wireless communication systems that can adapt to their environment and utilize the available spectrum efficiently. CRNs enhance spectrum utilization by allowing unlicensed users, referred to as SUs, to opportunistically access licensed spectrum bands when they are not occupied by the licensed users, known as PUs [18].

The key enabling technologies for CRNs include spectrum sensing, spectrum management, spectrum mobility, and spectrum sharing [19]. Spectrum sensing is a crucial component of CRNs, enabling SUs to detect the presence or absence of PUs in a given frequency band. Accurate and reliable spectrum sensing is essential to ensure the seamless coexistence of PUs and SUs and to prevent harmful interference. Spectrum management involves selecting the best available spectrum bands for SUs based on their quality of service (QoS) requirements. Spectrum mobility enables SUs to vacate the occupied spectrum band and switch to another available band when a PU is detected. Spectrum sharing addresses the coordination and allocation of spectrum resources among multiple SUs to optimize overall network performance.

Despite the potential benefits of CRNs, several challenges need to be addressed for their practical deployment. One of the primary challenges is designing efficient and robust spectrum sensing techniques [20]. Traditional spectrum sensing methods, such as energy detection and cyclostationary feature detection, have limitations in sensitivity, complexity, and robustness to noise and interference. Moreover, the performance of spectrum sensing techniques can be significantly affected by high-level noise, noise uncertainty [21], and fading.

To overcome these challenges, deep learning techniques have shown promising results in spectrum sensing. The following sections of this literature review will delve deeper into specific research works related to traditional and predominantly deep learning-based spectrum sensing techniques, which form the core of this thesis.

2.2 Traditional Spectrum Sensing Techniques

Traditional spectrum sensing techniques are fundamental to cognitive radio systems, enabling SUs to identify unoccupied spectrum bands for opportunistic access. Among these techniques, ED is widely recognized for its simplicity and low computational complexity. ED operates by comparing the received signal energy with a predefined threshold to determine the presence or absence of a PU. However, ED suffers from several limitations, such as its susceptibility to noise uncertainty and its requirement for accurate noise power estimation, particularly in low SNR levels [22]. Similarly, maximum-eigenvalue detection (MED) captures correlation features to offer great detection capabilities on highly correlated signals but still requires accurate prior knowledge of noise characteristics [23]. Additionally, a wavelet-based approach for efficient spectrum sensing of wideband channels has been proposed, which exploits the multi-resolution analysis capability of wavelets [24]. However, it is sensitive to noise in low SNR scenarios, timing synchronization errors, and requires high sampling rates to achieve accurate edge detection in the power spectral density, leading to increased hardware complexity and cost. To address the limitations of ED, a simplified predicted eigenvalue threshold (SPET) was proposed, which outperforms ED in the presence of impulsive noise. However, SPET is not applicable to a wide range of systems where the noise is not impulsive [25].

To enhance the performance of spectrum sensing, various advanced traditional methods have been proposed. Feature-based approaches such as cyclostationary feature detection exploit the inherent periodicity of modulated signals to achieve better performance [26]. Higher-order moments-based detection also leverages signal periodicity for improved detection performance [27]. Additionally, correlation feature detection enhances detection performance by exploiting the correlation between the received signal and a reference signal [28]. Phase difference distribution-based approaches further improve robustness by capitalizing on the phase difference between the received signal and a reference signal [29]. Despite their improved performance, these methods entail relatively higher computational complexities, hindering their efficiency in real-time spectrum sensing scenarios. Besides, totally-blind detection methods such as blindly combined energy detection (BCED) [30] and covariance absolute value (CAV) detection [31] have been developed to offer robustness against fluctuating noise power without requiring prior knowledge of noise characteristics. Although these methods provide resilience, they often slightly compromise performance compared to semi-blind methods [5]. Furthermore, matched-filter detection (MFD) is known to be the optimum detection method when the PU's signal is known [32]. This method correlates the received signal with a known template of the PU's signal to maximize the SNR. However, matched-filter detection requires perfect knowledge of the PU's signal characteristics, such as modulation type, which is often unavailable in real-world scenarios.

In summary, traditional spectrum sensing techniques offer various approaches to detect the presence of PUs in cognitive radio networks. While ED is widely regarded as the simplest and most computationally efficient method, it suffers from poor performance in low SNR environments and is sensitive to noise uncertainty. More advanced techniques have been developed to address the limitations of ED. For example, cyclo-stationary feature detection, while robust to noise uncertainty, is computationally demanding due to its reliance on spectral correlation analysis and requires prior knowledge of the signal's cyclo-stationary properties. Matched filter detection offers optimal performance by correlating the received signal with a known reference signal. However, it requires full knowledge of the PU signal and channel information, making it impractical in scenarios where the information is unavailable. Covariance-based and maximum-eigenvalue detection methods are often used in multiple-antenna systems that rely on statistical properties of the received signal. Covariance-based detection involves moderate computational

requirements, while maximum-eigenvalue detection is more computationally intensive due to the eigenvalue decomposition of covariance matrices. These advanced methods improve detection performance but come at the cost of higher complexity and, in some cases, increase reliance on prior knowledge. Table 2.1 provides a concise comparison of these techniques, summarizing their field of applicability, computational complexity, and information requirements to facilitate an intuitive understanding of their strengths and limitations.

TABLE 2.1: Comparison among traditional spectrum sensing methods

Sensing Technique	**Applicability**	**Complexity**	**Prior information**
Energy detection	No limit	Low	Not Required
Cyclo-stationary feature detection	No limit	High	Required
Matched Filter detection	No limit	High	Required
Covariance-based detection	Multiple Antenna	Moderate	Not Required
Maximum-eigenvalue detection	Multiple Antenna	High	Not Required

2.3 Deep Learning-based Spectrum Sensing Techniques

Deep learning-based approaches have revolutionized the field of spectrum sensing by enabling the extraction of complex features and patterns from raw signal data. These techniques leverage the power of deep neural networks to learn hierarchical representations and make accurate decisions regarding the presence or absence of primary users. The application of deep learning to spectrum sensing has opened up new avenues for research and has led to significant advancements in the field. In this section, we explore three key areas of deep learning-based spectrum sensing: supervised learning approaches, domain adaptation and generalization methods, and learning techniques without extensive labels.

Supervised learning approaches have been extensively investigated for spectrum sensing, utilizing labeled training data to learn discriminative features and decision boundaries. These methods have demonstrated promising results in enhancing sensing accuracy and robustness compared to traditional techniques. However, the

reliance on large amounts of labeled data presents challenges in practical deployment scenarios. Gathering extensive labeled datasets can be time-consuming and expensive, particularly in dynamic and complex environments.

To overcome the limitations of supervised learning, domain adaptation and transfer learning techniques have been explored for spectrum sensing. These approaches aim to transfer knowledge learned from a source domain to a target domain with limited labeled data. By leveraging similarities between domains, transfer learning facilitates the adaptation of deep learning models to new environments or signal characteristics, thereby reducing the need for extensive data collection and labeling efforts in the target domain. This significantly enhances the model's robustness and generalizability in varied wireless communication environments.

Furthermore, the scarcity of labeled data has driven the exploration of learning approaches that can function without extensive labels, such as unsupervised learning, self-supervised learning, and semi-supervised learning. These methods have shown promise in diminishing the dependence on labeled data while still achieving competitive sensing accuracy. Unsupervised learning techniques aim to discover hidden patterns in unlabeled data, self-supervised learning leverages pretext tasks to learn useful representations from unlabeled data, and semi-supervised learning combines a small amount of labeled data with a large amount of unlabeled data to improve model performance.

In the following subsections, we delve deeper into each of these areas, demonstrating the key achievements and challenges associated with supervised learning, domain adaptation, and learning approaches without extensive labels in the context of deep learning-based spectrum sensing.

2.3.1 Supervised Learning Approaches

In recent years, DL has emerged as a powerful tool for spectrum sensing in cognitive radio networks. DL-based approaches have shown great potential in improving the performance of spectrum sensing by leveraging the ability of deep neural networks to learn and extract complex features from raw signal data. This subsection provides an overview of the state-of-the-art DL-based supervised spectrum sensing techniques, focusing on various architectures such as DNN, CNN and RNN.

In [33], a hybrid spectrum sensing scheme integrating classical energy detection, likelihood ratio test statistics, and artificial neural networks (ANN) was proposed, which leverages the strengths of these methods to achieve rapid and accurate detection of PU signals. In [34], the authors proposed a DNN-based approach for spectrum sensing that utilizes the divergence of two phase difference distributions as the input. While these initial deep learning approaches showed potential for improving detection performance, they relied on conventional signal processing techniques for feature extraction. Furthermore, attention has progressively turned to CNNs for achieving higher signal detection accuracy. For instance, in [13], a deep cooperative sensing framework based on CNNs was proposed, demonstrating improved performance compared to traditional cooperative sensing methods. In [35], the covariance matrix was utilized as the input feature of the neural network, leading to the development of the CM-CNN method to address the spectrum sensing problem. Similarly, in [36], a CNN-based framework is employed to discern PU activity patterns without the necessity for predefined statistical models, showcasing enhanced detection accuracy over conventional methods. Besides, other features of PU signals have also been investigated. For example, in [37], a CNN model employing spectral correlation function (SCF) was proposed for spectrum sensing, which is an effective characterization of cyclostationarity property. In [38], a deep short-time Fourier transform CNN (STFT-CNN) for spectrum sensing was introduced, leveraging the time-frequency representation of signals to enhance feature extraction. Building on these CNN-based approaches, in [39], the application of CNNs was expanded through the integration of deep residual networks (ResNet) and transfer learning, which directly processes raw I/Q samples and aims to enhance the robustness of spectrum sensing across varying wireless environments.

To further improve the sensing performance, several studies have explored the integration of CNNs with other deep learning architectures. In [40], the integration of CNNs with long-short term memory (LSTM) networks was explored, exploiting their ability to extract spatial and temporal features from the sensing data, thereby learning PU activity patterns to significantly boost probability of detection, even in scenarios with noise uncertainty. Similarly, in [41], an enhanced CNN-RNN (ECRNN) approach was proposed for sensing available spectrum in cognitive radio networks. In [42], the DetectNet based on convolutional long-short term memory fully connected deep neural network (CLDNN) was proposed, without the need for prior channel state information and noise level knowledge. Similarly, In [43],

DLSenseNet was designed, where parallel CNNs extract signal features, followed by LSTM blocks and fully connected layers. In [44], a combination of 1D CNN, LSTM, and fully connected layers was used to establish a DL-based spectrum sensing model. Additionally, ensemble learning, which combines multiple deep learning models, has been explored to enhance spectrum sensing. In [45], an ensemble DL-based cooperative spectrum sensing method was proposed with a stacking fusion center, aiming to improve the robustness and accuracy of sensing decisions. However, the requirement for target domain data during the training phase limited its practical deployment in dynamic wireless environments.

The application of DL for spectrum sensing in specific wireless systems, such as orthogonal frequency-division multiplexing (OFDM), has also been investigated. In [46], the spectrum sensing problem was viewed as an SNR-related multi-class classification issue within an OFDM system, which employs a naive Bayesian classifier (NBC) and a class-reduction prediction method to enhance the sensing accuracy. In [47], a DL-based spectrum sensing method was proposed for OFDM systems, utilizing CNNs to learn the features of the OFDM signal. They demonstrated the effectiveness of their approach through simulations and experiments. Exploiting the cross-correlation property of the cyclic prefix of OFDM signals, another study proposed a method to sense the full-duplex OFDM system via a matching network (MN), though this scheme is applicable only to certain types of signals [48]. Besides, in [49], a robust stacked autoencoder-based spectrum sensing method (SAE-SS) was proposed. Moreover, in [50], CNNs have been utilized to analyze the cyclic spectrum of OFDM signals by transforming them into grayscale images for improved feature extraction and signal detection. Similarly, in [37], a spectrum sensing and signal identification approach based on deep learning and the spectral correlation function specifically targeted OFDM signals. While these approaches demonstrated effectiveness for OFDM systems by leveraging specific signal properties like cyclic prefix and subcarrier features, their strong dependency on OFDM-specific characteristics inherently limited their generalization ability to other modulation schemes or communication systems.

DL-based spectrum sensing has also been explored in the context of emerging wireless networks, such as space-air-ground integrated networks. In [51] a DL-based

spectrum sensing framework was proposed for space-air-ground integrated networks, considering the unique challenges and requirements of these networks. Additionally, in [52], a low-complexity DL approach for spectrum sensing in satellite-based systems was proposed to reduce computational overhead. In the context of unmanned aerial vehicles (UAVs), various DL-based spectrum sensing and sharing approaches have been investigated, addressing the specific needs of UAV communication systems [53–55]. Wideband spectrum sensing has also been a focus of research [56–61]. For instance, in [60], a real-time wideband spectrum sensing software/hardware framework named DeepSense, based on CNN, was proposed to enhance real-time detection capabilities. While achieving real-time processing, DeepSense faces challenges in memory constraints when deployed on edge devices. Furthermore, a parallel CNN-based wideband spectrum sensing approach was introduced, featuring low latency and the ability to generalize across different protocols such as LTE-M uplink transmissions, 5G NR downlink transmissions, and WiFi transmissions [61].

In conclusion, the application of DL to spectrum sensing has demonstrated significant improvements over traditional methods. However, existing methods do not adequately address the impact of heavy noise on sensing performance, particularly in low SNR scenarios. As illustrated in Chapter 3, we propose a two-step framework, which utilizes a stacked convolutional autoencoder (SCAE) for signal enhancement, followed by a hybrid CNN-SA-GRU (H-CSG) network for spectrum sensing [62]. This approach outperforms existing DL-based spectrum sensing methods, and the SCAE component is compatible with other DL architectures. However, it necessitates separate training phases for the denoising and detection processes, making the detection performance heavily dependent on the efficacy of the denoising stage. To address these challenges, as illustrated in Chapter 4, we introduce an end-to-end joint denoising and spectrum sensing (JDSS) network that optimizes the balance between denoising and detection, thereby prioritizing overall sensing accuracy[63].

While DL-based spectrum sensing methods have significantly improved probability of detection by optimizing model designs, they predominantly rely on supervised learning strategies, which needs large numbers of labeled data to ensure robust model performance. In real-world radio environments, acquiring signal samples may be straightforward, yet the availability of high-quality labels is often limited

by economic and security considerations. Besides, the manual labeling of these signals is impractical and prone to errors due to the labor-intensive nature of the task and the risk of mislabeling. In the following subsections, we will delve into how transfer learning is applied to enhance the deep learning model's robustness and how to reduce the dependency on labeled samples for training.

2.3.2 Domain Adaptation and Transfer Learning in Spectrum Sensing

Generally, most of DL-based spectrum sensing methods outperform traditional feature-based methods in scenarios where they are trained. However, the environment of electromagnetic wave propagation is considerably diverse, which means the received signal distribution is highly impacted. Hence, the sensing model trained well under one environment may work poorly in another one. One approach to address the challenge of the environment variations is to gather sufficient labeled samples in various contexts, which is difficult to realize in practice. An alternative approach focuses on developing models that can generalize across different domains or adapt to new environments with minimal retraining. In this context, domain adaptation techniques, such as transfer learning, have emerged as promising methods to enhance the adaptability of spectrum sensing models to new scenarios with limited labeled data. Alternatively, domain generalization techniques, which aim to extract channel-invariant features, offer a complementary approach. These methods strive to develop robust models capable of maintaining consistent performance across diverse environments without the need for explicit adaptation.

The foundation of transfer learning lies in the ability to transfer knowledge acquired from one domain to another, as discussed in the seminal survey [64]. Building upon this concept, researchers have explored various transfer learning techniques to improve spectrum sensing performance in dynamic wireless environments. One of the applications of transfer learning in spectrum sensing is in the detection of signals in ambient backscatter communications. In [65], a deep transfer learning method was proposed, which leverages knowledge from a source domain with abundant labeled data to enhance signal detection in a target domain with limited labeled samples. Their approach demonstrates the effectiveness of transfer learning in improving the detection performance and reducing the need for extensive

labeled data in the target domain. In [66], transfer learning was utilized to non-cooperative spectrum sensing, where the characteristics of the primary user's signal are unknown. It highlights the potential of transfer learning in addressing the challenges of non-cooperative spectrum sensing and reducing the reliance on labeled data. Adversarial learning has also been explored in combination with transfer learning to enhance spectrum sensing performance. In [67], a spectrum sensing approach based on adversarial transfer learning was proposed, where a generative adversarial network (GAN) is employed to align the feature distributions of the source and target domains. By minimizing the domain discrepancy, their method achieves improved spectrum sensing accuracy in the target domain. Similarly, in [68], the authors introduced adversarial learning strategy for dealing with performance degradation with untrained testing SNR sets, but it requires training SNR labels to extract the SNR-invariant features. In [69], the authors proposed a deep transfer cooperative sensing (DTCS) approach to enhance the adaptation ability, but it needs a pre-determined target domain during training and the algorithm operates on energy level for each SU, which could ignore more important features extracted from raw signals. Similarly, in [70], the use of adversarial transfer learning for automatic modulation classification was investigated, which demonstrates its effectiveness in adapting to new modulation schemes. In [71], a novel CNN-based deep learning model was proposed and transfer learning technique was employed for spectrum sensing and modulation recognition, which is able to adapt the model to new scenarios and improve its generalization capacity. The integration of multiple deep learning architectures with transfer learning has also shown promising results in spectrum sensing. In [72], a CNN-RNN based transfer learning approach was proposed for spectrum sensing in cognitive radio networks. By combining the spatial feature extraction capabilities of CNNs with the temporal modeling abilities of RNNs, their method achieves improved sensing performance and adaptability to new environments. However, transfer learning approaches often require access to data from the target domain or assume a predefined set of target domains, which may not always be feasible in real-world spectrum sensing scenarios. Besides, most existing methods focus on adapting to specific channel conditions rather than learning truly channel-invariant features, limiting their generalization capabilities. To address these limitations, domain generalization techniques have emerged as a promising alternative, aiming to develop models that can generalize well across various unseen domains without requiring explicit adaptation or access

to target domain data.

In conclusion, while transfer learning has shown great promise in enhancing the performance and adaptability of deep learning-based spectrum sensing in cognitive radio networks, domain generalization techniques offer a complementary approach to address the challenges of environmental variations. Domain generalization methods aim to learn domain-invariant features that can maintain consistent performance across diverse and unseen environments without the need for explicit adaptation. Motivated by the limitations of existing domain adaptation approaches and the potential of domain generalization, we propose an end-to-end spectrum sensing network named ER-SNet [73], as illustrated in Chapter 5. This novel approach is designed to improve the model's generalization ability for untrained communication environments by extracting robust and channel-invariant features.

2.3.3 Learning Without Extensive Labels: Unsupervised, Self-Supervised, and Semi-Supervised Approaches

In the realm of spectrum sensing for cognitive radio networks, the availability of extensive labeled training data has been a persistent challenge. Acquiring large amounts of labeled data can be time-consuming, costly, and impractical in dynamic wireless environments where the characteristics of the signals and the behavior of the primary users may change over time. Moreover, relying solely on supervised learning approaches, which require labeled data, may limit the adaptability and scalability of spectrum sensing systems. To address these challenges, there has been a growing interest in exploring learning approaches that can operate effectively with limited or no labeled data. These approaches, namely unsupervised learning, semi-supervised learning, and self-supervised learning, have the potential to revolutionize spectrum sensing by enabling the discovery of underlying patterns and structures in the data, reducing the dependence on labeled samples, and facilitating the adaptation to new and unseen environments.

Unsupervised learning approaches have gained significant attention in the field of spectrum sensing for cognitive radio networks, as they enable the detection of primary users' signals without the need for extensive labeled training data. These approaches aim to discover hidden patterns and structures in the received

signal data, allowing for efficient and adaptive spectrum sensing in dynamic wireless environments.

Several studies have explored unsupervised learning approaches for spectrum sensing. For example, a comparison of K-means clustering, Gaussian mixture models (GMM), support vector machines (SVM), and weighted K-nearest neighbor classifiers paved the way for spectrum sensing based on learning approaches [11]. Another study introduced a generative adversarial learning framework for spectrum sensing, leveraging the power of GANs to improve sensing performance [74]. In addition, a variational autoencoder (VAE)-based approach was proposed, where the VAE is trained to learn a compact representation of the covariance matrix of received signal data, capturing the underlying statistical characteristics of the signals and enabling detection without labeled training samples [75]. Building on the concept of unsupervised learning, researchers have explored its application in cooperative spectrum sensing scenarios. An unsupervised two-stage learning framework for cooperative spectrum sensing was introduced, where the first stage involves unsupervised feature learning using autoencoders, and the second stage employs clustering algorithms for decision fusion [76]. Clustering algorithms have also been extensively explored in unsupervised learning-based spectrum sensing. For example, in [77, 78], the robust features of received signal samples, such as eigenvector or eigenvalues, are clustered using the k-means algorithm or GMM to conduct the spectrum sensing task. Further advancements in unsupervised clustering approaches have combined deep learning with clustering techniques to learn discriminative features and make accurate sensing decisions [79, 80]. However, these methods often require prior knowledge of the number of clusters and face initialization challenges. To address these limitations, an approach utilizing a sparse autoencoder to learn useful representations of sensing data and the affinity propagation (AP) algorithm to cluster the representations and determine the channel state was proposed [81].

While unsupervised learning approaches have shown promising results in spectrum sensing, they often rely solely on the inherent structure and patterns in the received signal data. Other learning paradigms, such as self-supervised learning and semi-supervised learning, can be leveraged to further enhance the performance of spectrum sensing models.

Semi-supervised learning approaches leverage a small amount of labeled data in addition to the unlabeled data to guide the learning process. By incorporating limited labeled samples, semi-supervised learning can provide additional supervision and improve the accuracy of spectrum sensing models. However, when the amount of labeled data are extremely limited, the initial model trained on labeled samples may not provide reliable pseudo-labels for unlabeled data, leading to error propagation through the training process. For instance, SSDNN [82] relies on high-confidence samples for model updating, but with very scarce labeled data, the confidence measure itself becomes unreliable. Besides, the performance of semi-supervised learning heavily depends on the quality of labeled samples. If the limited labeled data are not representative of the overall data distribution, the model may learn biased representations. On the other hand, self-supervised learning, which is a paradigm tailored for leveraging unlabeled data to train deep neural networks, has also become an increasingly popular framework due to its ability to mitigate the constraints associated with extensive data labeling. Self-supervised learning employs pretext tasks to generate self-supervision, enabling the model to learn meaningful representations from the data itself without requiring explicit labels. This approach has gained traction across various domains, including computer vision (CV) and natural language processing (NLP). In CV field, tasks like image colorization [83], jigsaw puzzle solving [84], and rotation prediction [85] have been employed to generate self-supervision. One commonly used framework in self-supervised learning is contrastive learning, which aims to distinguish between positive and negative samples. In [86], a straightforward yet powerful framework for the contrastive learning of visual representations, named SimCLR, demonstrated notable success on the ImageNet dataset [87]. SimCLR learns input image representations by maximizing the agreement between the latent features of two different augmentations of the same input. Inspired by the simplicity and proven effectiveness of SimCLR, we have integrated this framework into our self-supervised learning strategy, specifically tailored to tackle the challenges of spectrum sensing. In NLP field, Word2Vec paper [88], introduced two groundbreaking architectures for generating continuous word representations, focusing on pretext tasks that either predict the central word from its context or predict the surrounding words based on the central word. In [89], an innovative approach BERT that pretrains deep bidirectional representations from unlabeled text is presented. It enhances language representation learning by predicting randomly

masked words in a sentence, demonstrating remarkable development in language representation learning.

The intense demand for labeled samples in conventional DL approaches has recently prompted researchers to employ self-supervised learning in the field of wireless communication, necessitating careful design of network architecture, encoder, and data augmentation strategies. For instance, in [90], a contrastive learning framework BYOL was also proposed and the residual network was designed to extract rich features from unlabeled samples. Additionally, in [91], short-time Fourier neural network (STFNet) [92] was introduced within the SimCLR framework to learn representations by aligning time-domain and frequency-domain representations of the same sample closer together, while distancing them from representations of different signals. In our previous work on automatic modulation classification [93], a beta-VAE-based data augmentation module was incorporated into the self-supervised learning process to preserve the identity of the sample while generating diverse views. However, the existing literature on spectrum sensing with self-supervised learning predominantly explores data augmentation from various perspectives, such as time and frequency domains, while often overlooking the intrinsic dynamics of signals. Moreover, these studies generally do not assess the robustness of the models across diverse radio environments, including variations in signal modulations, oversampling rates, and channel conditions. Therefore, there is considerable scope for enhancing spectrum sensing performance through the self-supervised learning paradigm. To address these limitations, we propose a robust and efficient self-supervised learning framework for non-cooperative spectrum sensing named DC4S [94], as illustrated in Chapter 6.

2.3.4 Challenges and Limitations of Deep Learning-based Spectrum Sensing

While deep learning-based approaches have demonstrated significant advantages over traditional methods in spectrum sensing, it is crucial to acknowledge and understand their inherent limitations and challenges. These challenges not only affect the practical deployment of deep learning models but also influence their reliability and adoption in real-world cognitive radio networks. This section discusses several

key limitations that need to be carefully considered when developing deep learning solutions for spectrum sensing.

Firstly, while deep learning models can achieve superior detection performance, they require extensive computational resources for training and inference. This is particularly challenging in spectrum sensing applications where real-time processing is often required.

Secondly, the "black box" nature of deep learning models presents unique challenges in spectrum sensing applications. Unlike traditional methods such as energy detection that have clear theoretical foundations, deep learning approaches make it difficult to interpret why certain signals are classified as present or absent. This lack of interpretability becomes particularly critical in scenarios where missed detection could lead to harmful interference with primary users.

Thirdly, we found that deep learning models for spectrum sensing are sensitive to hyperparameter settings, from network architecture choices to training parameters. This tuning process is time-consuming and the optimal settings may not generalize well across different channel conditions or signal types.

Chapter 3

DL-based Two-Stage Non-Cooperative Spectrum Sensing

3.1 Introduction

In the era of rapidly growing wireless communication technologies, the demand for efficient spectrum utilization has become increasingly crucial. Cognitive radio networks have emerged as a promising solution to address the spectrum scarcity problem by enabling dynamic spectrum access. In these networks, unlicensed users, referred to as SUs, can opportunistically access the licensed spectrum bands when they are not occupied by the licensed users, known as PUs. However, to ensure seamless coexistence and prevent harmful interference to the PUs, accurate and reliable spectrum sensing is of paramount importance.

Traditional spectrum sensing techniques, such as ED [6], MED [23], and cyclostationary feature detection [26], have been extensively studied in the literature. However, these methods often face challenges in practical wireless communication scenarios, particularly in low SNR environments. The presence of noise, fading, and other channel impairments can significantly degrade the detection accuracy, leading to missed detection of PUs or false alarms. Moreover, these techniques often rely on prior knowledge of the signal and noise characteristics, which may not always be available in dynamic wireless environments.

To address these limitations, there has been a growing interest in applying deep learning techniques to the spectrum sensing problem. Deep learning has demonstrated remarkable success in various domains due to its ability to learn hierarchical representations and capture complex patterns from data. Recently, the advancements in deep learning have opened up new possibilities for tackling the spectrum sensing problem. Deep learning-based approaches have shown promising results in various wireless communication tasks, including modulation classification [95, 96], channel estimation [97, 98], and signal detection [99, 100]. These successes have motivated researchers to explore the application of deep learning techniques to enhance spectrum sensing performance. In [13], the authors designed a CNN-based approach for cooperative spectrum sensing. For the same scenario, the "DetectNet" based on convolutional long-short term memory network (CLDNN) was proposed in [42]. In order to extend the multiple CNNs across different sites, wireless environments and frequencies, the ResNet architecture and transfer learning methods were applied to strengthen its robustness [39]. However, existing deep learning-based spectrum sensing methods often overlook the impact of noise on the sensing performance, which can significantly degrade their effectiveness in low SNR scenarios.

In this chapter, our focus is on developing a novel two-stage deep learning-based spectrum sensing approach that incorporates denoising to enhance detection performance in low SNR environments. The proposed approach consists of a denoising stage and a sensing stage. Firstly, the autoencoder-based pre-processing stage aims to reconstruct clean signal samples from noisy observations, effectively reducing the impact of noise on the sensing performance. By training the autoencoder on a dataset containing both clean and noisy signal samples, it learns to map noisy inputs to their corresponding clean versions. The denoised signal samples are then fed into the subsequent sensing model for spectrum occupancy prediction. In the second stage, a hybrid architecture called H-CSG is introduced, which combines the strengths of CNNs, self-attention (SA) mechanisms, and gated recurrent units (GRUs). The CNN component captures spatial features, while the SA mechanism helps to focus on the most informative parts of the input sequence. GRUs, a variant of RNNs, are utilized to model temporal dependencies and capture the sequential nature of the signal samples. By incorporating denoising and leveraging the power of deep learning architectures, this method seeks to overcome the limitations of traditional spectrum sensing techniques and enhance the efficiency of cognitive radio

networks.

The rest of this chapter is organized as follows. Section 3.2 describes the problem formulation and signal model. Section 3.3 outlines the framework of the proposed two-stage approach, including the autoencoder-based pre-processing stage and the H-CSG sensing model. Simulation results and discussion are provided in Section 3.4, which covers dataset generation, complexity analysis, time-frequency analysis, performance evaluation and ablation study. Finally, Section 3.5 presents the conclusion of this work.

3.2 System and Signal Model

In this section, we assume a single-antenna cognitive radio scenario where the SU engages in spectrum sensing without any prior knowledge, a condition often termed as totally blind spectrum sensing. In such a scenario, the SU lacks knowledge regarding the PU's signal characteristics or channel state information, which introduces significant challenges in reliable signal detection.

Let $\mathbf{r}_m = [r_m(0), r_m(1), \ldots, r_m(L-1)] \in \mathbb{C}^{L\times 1}$ denote the complex baseband received signal at the SU, where m represents the m-th discrete-time sample from the PU transmitter and L is the sample length. The spectrum sensing problem can be formulated as a binary hypothesis testing problem, where the two hypotheses are defined as follows:

$$\begin{cases} H_0 : \mathbf{r}_m = \mathbf{w}_m, \\ H_1 : \mathbf{r}_m = h_m \cdot \mathbf{x}_m + \mathbf{w}_m, \end{cases} \tag{3.1}$$

where H_0 represents the absence of the PU signal, and H_1 indicates the presence of the PU signal. The noise vector $\mathbf{w}_m = [w_m(0), w_m(1), \ldots, w_m(L-1)] \in \mathbb{C}^{L\times 1}$ is assumed to be independent and identically distributed (i.i.d.) additive noise following a circularly symmetric complex Gaussian (CSCG) distribution with zero mean and variance σ_w^2. The signal vector $\mathbf{x}_m = [x_m(0), x_m(1), \ldots, x_m(L-1)] \in \mathbb{C}^{L\times 1}$ represents the m-th pure modulated signal from the PU without noise, and h_m denotes the complex channel gain between the PU and SU, which is assumed to follow a Rayleigh fading model. We consider a slow fading scenario where

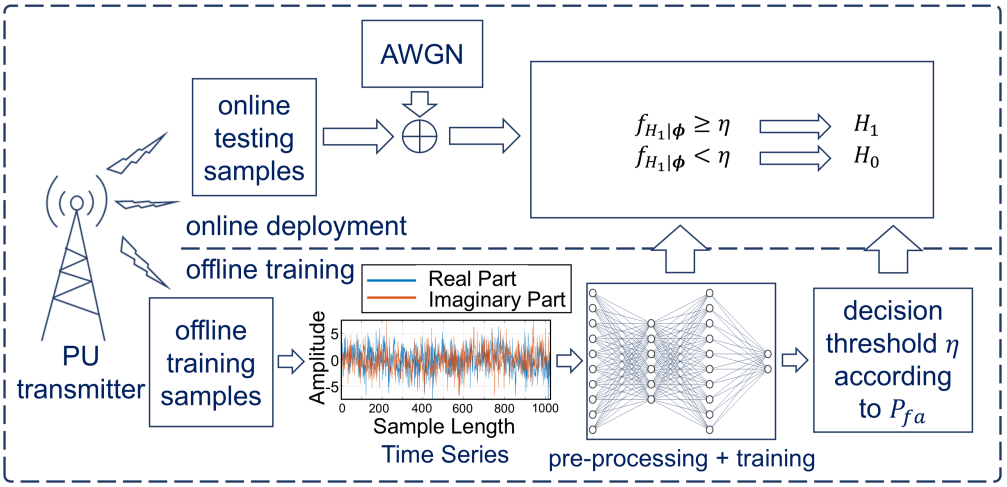


FIGURE 3.1: The proposed deep learning-based spectrum sensing workflow.

the channel gain remains constant over a sensing period but varies independently between different sensing periods [101].

Typically, two performance indicators are considered in spectrum sensing: the probability of detection P_d and the probability of false alarm P_{fa}. P_d represents the probability of correctly detecting the presence of the PU signal when it is actually present, while P_{fa} represents the probability of erroneously detecting the presence of the PU signal when it is absent. In the context of deep learning-based spectrum sensing, the output of the final activation layer of the neural network provides the probability of each hypothesis (i.e., H_0 and H_1). The decision rule for the binary hypothesis testing problem is based on comparing the output probability under hypothesis H_1 with a threshold η, which is calculated by a pre-determined P_{fa}. Mathematically, the performance metrics can be defined as follows:

$$\begin{cases} P_d = \mathrm{P_r}\left\{ f_{H_1|\boldsymbol{\phi}}(\mathbf{r}) > \eta \mid H_1 \right\}, \\ P_{fa} = \mathrm{P_r}\left\{ f_{H_1|\boldsymbol{\phi}}(\mathbf{r}) > \eta \mid H_0 \right\}, \end{cases} \tag{3.2}$$

where $f_{H_1|\boldsymbol{\phi}}(\mathbf{r})$ represents the output probability of label H_1 under the well-trained model weights $\boldsymbol{\phi}$ of our proposed deep learning-based spectrum sensing model $f(\cdot)$, and $\mathrm{P_r}\{\cdot\}$ denotes the probability of a specified event.

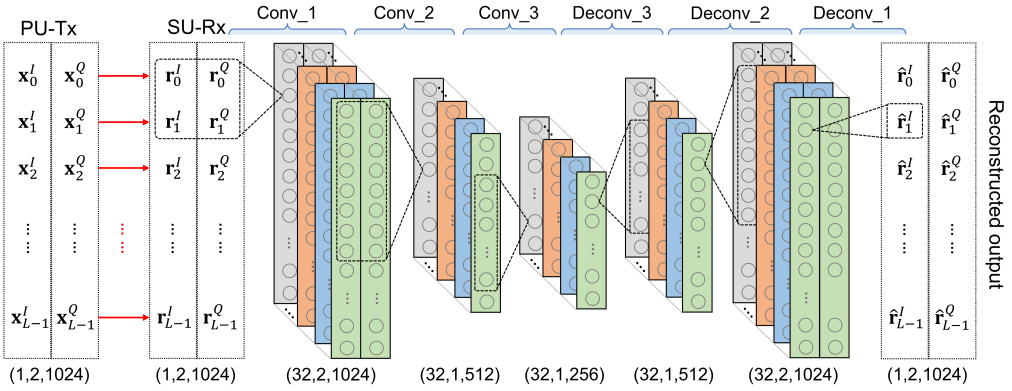


FIGURE 3.2: The proposed architecture of SCAE.

3.3 Network Design and Training Procedure

The proposed DL-based spectrum sensing workflow, as illustrated in Figure 3.1, comprises two main stages: offline training and online deployment. This approach is designed to be robust against various factors, such as modulation types and time offset. The first step involves introducing a stacked convolutional autoencoder (SCAE) network as a pre-processing method for signal reconstruction. Subsequently, the reconstructed dataset $\mathcal{D}^{Re}$, containing both real and imaginary parts, is fed into a Hybrid CNN-SA-GRU (H-CSG) network to determine the presence or absence of the PU based on an adaptively modified threshold. In the following subsections, we will delve into the details of the pre-processing method and the sensing network training process.

3.3.1 Autoencoder-based Pre-processing

Inspired by the unsupervised SCAE proposed in [102] for feature extraction, we employ a similar structure as a supervised method to denoise the radio signals by regressing the noisy data to clean data. The architecture and hyperparameters of the SCAE are depicted in Figure 3.2 and Table 3.1. The training process of the SCAE consists of the following steps:

Step 1. The first convolutional layer $Conv1$ and the last transposed convolutional layer $Deconv1$ are treated as the first convolutional autoencoder (CAE). By training this network, we obtain the weights $\mathcal{W}_{SCAE}^{(1)}$ of the hidden layer and the output features $\mathbf{z}_{SCAE}^{(1)}$ of the $Conv1$, which is denoted as the first encoding layer.

TABLE 3.1: Hyper-parameters of SCAE network

Layer	Stride	Kernel Size
Conv_1	(1,1)	32 @ (2,2)
Conv_2	(2,2)	32 @ (2,8)
Conv_3	(1,2)	32 @ (1,16)
Deconv_3	(1,2)	32 @ (1,16)
Deconv_2	(2,2)	32 @ (2,8)
Deconv_1	(1,1)	1 @ (2,2)

Step 2. The output features $\mathbf{z}_{SCAE}^{(k)}$ from the previous step are used as the input for the $(k+1)$-th CAE, i.e., the layers $Conv(k+1)$ and $Deconv(k+1)$. Through this process, we obtain the weights $\mathcal{W}_{SCAE}^{(k+1)}$ and output features $\mathbf{z}_{SCAE}^{(k)}$ of the $(k+1)$-th encoding layer, where $k = 1, 2$, as our SCAE structure consists of three independent CAEs.

Step 3. The initialized weights obtained from *Step 1* and *Step 2* are used to conduct fine-tuning for the entire SCAE network.

As the denoising process is a regression problem, we adopt the mean-square error (MSE) loss function, which is defined as follows:

$$\mathcal{L}_{SCAE}(\boldsymbol{\theta}) = \frac{1}{B}\left(\sum_{i=1}^{B} \left|g_{\boldsymbol{\theta}}(\mathbf{r}_i) - \mathbf{x}_i\right|^2\right), \tag{3.3}$$

where B represents the batch size, $g_{\boldsymbol{\theta}}$ is the transformation function, and $\boldsymbol{\theta}$ denotes the weights of the SCAE. We generate a dataset $\mathcal{D}^s$ containing both modulated signals with label 1 and noise-only signals with label 0 under various SNR conditions for training the SCAE. For the i-th sample, $g_{\boldsymbol{\theta}}(\mathbf{r}_i)$ represents the predicted output after the regression layer, and $\mathbf{x}_i$ is the corresponding clean signal without corruption. Importantly, for samples with label 1, $\mathbf{x}_i$ represents the clean modulated signal without noise corruption, while for samples with label 0, $\mathbf{x}_i$ is set to zero, representing the absence of a primary user signal.

The optimal network parameters $\boldsymbol{\theta}$ are determined using the adaptive moment estimation (Adam) optimizer on the loss function. Another dataset with random

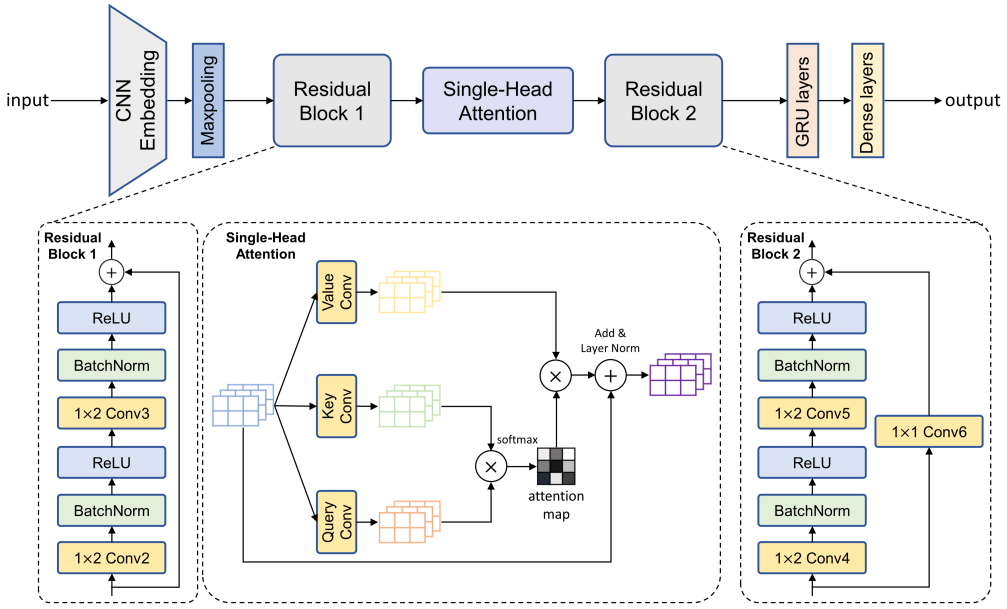


FIGURE 3.3: The proposed architecture of H-CSG network.

labels of 0 or 1, generated using the same method as $\mathcal{D}^s$, is created for evaluating the performance of the SCAE. The output of this dataset will be subsequently used for training the H-CSG network. After the denoising stage, we obtain a new dataset $\mathcal{D}^{Re}$ with reconstructed signal samples $\hat{\mathbf{r}}_i$, where $\hat{\mathbf{r}}_i = \{\hat{r}_i(0), \hat{r}_i(1), \ldots, \hat{r}_i(L-1)\} \in \mathbb{C}^{L\times 1}$. These reconstructed samples represent either denoised PU signals or noise reduction in the absence of a PU signal, depending on their original labels.

3.3.2 Proposed H-CSG Sensing Model and Training

In the proposed H-CSG sensing model, we treat the received signal as a $1\times 2\times 1024$ image and employ a CNN-based architecture to process it. However, simply stacking CNN modules can lead to the gradient vanishing problem, where the information disappears as the depth of the network increases. To mitigate this issue, we incorporate residual blocks instead of simple stacked CNN modules. Residual blocks add the input to the output of several layers, allowing the gradient to propagate unhindered and facilitating the training of deep networks. The use of CNN layers is motivated by their proven capability to extract robust spatial features from the signal, which enhances pre-processing and sets a strong foundation for subsequent temporal modeling by GRU layers and attention-based refinement. In addition to the CNN modules, we utilize GRU layers to model temporal dependencies and capture the sequential nature of the signal samples. The GRU layers were

chosen for their ability to effectively capture temporal dependencies in sequential data with relatively lower computational complexity compared to LSTM layers. The architecture of the proposed H-CSG network is illustrated in Figure 3.3. It consists of one CNN embedding layer, one max-pooling layer, two residual blocks, a SA module [103] placed between the residual blocks, followed by two GRU layers and two fully-connected (FC) layers. The hyperparameters of the H-CSG network are detailed in Table 3.2.

TABLE 3.2: Hyper-parameters of H-CSG network

Layer Name	**H-CSG Hyper-Parameters Setting**	
CNNs	**Stride**	**Kernel size**
CNN Embedding Layer	(1,1)	64 @ (2,2)
Residual CNN Block1: Conv2,3	(1,1)	64 @ (1,2)
Residual CNN Block2: Conv4	(2,2)	128 @ (1,2)
Residual CNN Block2: Conv5	(1,1)	128 @ (1,2)
Residual CNN Block2: shortcut Conv6	(2,2)	128 @ (1,1)
Self-attention: Value Conv	(1,1)	64 @ (1,2)
Self-attention: Key, Query Conv	(1,1)	32 @ (1,2)
GRUs	**Units Number**	
GRU Layer1	64	
GRU Layer2	32	
Fully Connected Layers	**Units Number**	
Dense Layer1	16	
Dense Layer2	2	
Output: Score Vector (Dimension: Batch $\times$ 2)		

The main objective of incorporating the SA module is to focus on the most informative regions of the input signal that exhibit strong dependencies. The SA module has three branches named "query", "key", and "value". Let $x^{W\times H\times 64}$ be the input to the SA module. We apply three 2×2 convolutional layers with different filter sizes of 32, 32, and 64 on the input to obtain outputs $x_1^{(W\times H)\times 32}$, $x_2^{(W\times H)\times 32}$, and $x_3^{(W\times H)\times 64}$, respectively. The SA coefficient map can be obtained as follows:

$$c_1 = \text{softmax}\left(x_1 \cdot {x_2}^T\right), \tag{3.4}$$

where T denotes matrix transpose. By computing the dot product between x_1 and ${x_2}^T$, the self-attention mechanism effectively extracts the complex dependencies among the input signal components. This capability is particularly beneficial in high-order modulation schemes where each symbol carries multiple bits, allowing the model to capture intricate relationships that may influence the signal's representation across the spectrum. The feature attention map is obtained by multiplying the "value" (x_3) and the weights given by the coefficient map:

$$\hat{x} = c_1 \cdot x_3. \tag{3.5}$$

With the help of residual connection, the information propagation is enhanced. Therefore, the final output of the SA module is

$${\mathcal{A}_{SA}}^{W\times H\times 64} = x + \hat{x}. \tag{3.6}$$

To train the H-CSG network, we label the k-th input time series signal $\hat{\mathbf{r}}_k$ as $(\hat{\mathbf{r}}_k, b_k)$, where b_k is the corresponding sample label. For the binary classification problem, the output of the softmax function is given by:

$$f_{\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right) = \begin{bmatrix} f_{\boldsymbol{\phi}|H_0}\left(\hat{\mathbf{r}}_k\right) \\ f_{\boldsymbol{\phi}|H_1}\left(\hat{\mathbf{r}}_k\right) \end{bmatrix}, \tag{3.7}$$

subject to the constraint:

$$f_{\boldsymbol{\phi}|H_0}\left(\hat{\mathbf{r}}_k\right) + f_{\boldsymbol{\phi}|H_1}\left(\hat{\mathbf{r}}_k\right) = 1, \tag{3.8}$$

where $\boldsymbol{\phi}$ represents the trained weights of the proposed H-CSG network, and $f(\,\cdot\,)$ denotes the output expression of the network model. The term $f_{\boldsymbol{\phi}|H_i}\left(\hat{\mathbf{r}}_k\right)$ represents the predicted probability of the hypothesis $H_i (i = 0, 1)$. Following the maximum-likelihood rule, we adopt a general objective function:

$$\ell_{H-CSG} = \prod_{k}^{B} \left(f_{\boldsymbol{\phi}|H_0}\left(\hat{\mathbf{r}}_k\right)\right)^{1-b_k} \left(f_{\boldsymbol{\phi}|H_1}\left(\hat{\mathbf{r}}_k\right)\right)^{b_k}. \tag{3.9}$$

During the process of obtaining the optimal model parameters, we minimize the loss function instead of maximizing the likelihood function. The categorical cross-entropy loss function is defined as:

$$\mathcal{L}_{CE}(\boldsymbol{\phi}) = \frac{1}{B}\sum_{k}^{B} - \left[b_k \cdot \log\left(f_{\boldsymbol{\phi}|H_1}\left(\hat{\mathbf{r}}_k\right)\right) + (1-b_k) \cdot \log\left(1 - f_{\boldsymbol{\phi}|H_1}\left(\hat{\mathbf{r}}_k\right)\right)\right]. \quad (3.10)$$

To satisfy the transmission requirements based on the IEEE 802.22 standard, we need to maintain a constant probability of false alarm P_{fa}. Since the output of the regression layer is the posterior probability of each hypothesis condition with equal prior probabilities, a decision is made according to the maximum a posteriori (MAP) rule:

$$PU \text{ is } \begin{cases} \text{active}, & f_{H_1|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right) > f_{H_0|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right), \\ \text{idle}, & f_{H_1|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right) < f_{H_0|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right), \end{cases} \quad (3.11)$$

where $f_{H_0|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right)$ and $f_{H_1|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right)$ are the output probabilities of each hypothesis from the well-trained H-CSG network.

By comparing $f_{H_0|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right)$ and $f_{H_1|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right)$, the PU state can be determined, but it cannot ensure a constant probability of false alarm. Hence, a decision threshold η is determined to replace the above decision condition. We adopt the Monte-Carlo scheme to randomly pick N samples $\{\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{N-1}\}$ under the *idle* condition and rearrange them in an increasing order based on $f_{H_1|\boldsymbol{\phi}}\left(\hat{\mathbf{r}}_k\right)$. The reordered N samples are denoted as $\{\widetilde{\mathbf{r}}_0, \widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_{N-1}\}$, where

$$\forall 1 < m < n < N, f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_m\right) < f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_n\right). \quad (3.12)$$

The sequence of output probabilities under H_1 for the reordered N samples is given by:

$$\left\{f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_0\right), f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_1\right), \ldots, f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_{N-1}\right)\right\}, \quad (3.13)$$

where $f_{H_1|\boldsymbol{\phi}}(\widetilde{\mathbf{r}}_0)$ is the smallest and $\widetilde{\mathbf{r}}_0$ is the i-th sample of the reordered sequence. The decision threshold is determined as:

$$\eta = f_{H_1|\boldsymbol{\phi}}\left(\widetilde{\mathbf{r}}_{\text{floor}\left(N*\left(1-P_{fa}\right)\right)}\right), \tag{3.14}$$

where the function floor$(\cdot)$ denotes rounding down to the nearest integer. Thus, we can control the actual P_{fa} to be consistently smaller than the desired P_{fa}, and the PU state can be determined by

$$\text{PU is } \begin{cases} \text{active}, & f_{H_1|\boldsymbol{\phi}}(\mathbf{r}_i) > \eta, \\ \text{idle}, & f_{H_1|\boldsymbol{\phi}}(\mathbf{r}_i) < \eta. \end{cases} \tag{3.15}$$

In conclusion, the proposed two stage spectrum sensing methodology, encompassing SCAE network for denoising and H-CSG network for sensing, is concisely summarized in **Algorithm 1**.

3.4 Simulation Results and Discussion

In this section, we present the simulation results and discuss the performance of our proposed deep learning-based spectrum sensing approach. We begin by describing the dataset generation process. Next, we provide a complexity analysis of the proposed model to evaluate its computational efficiency. Additionally, we perform a time-frequency analysis to assess the denoising efficacy of the method. We then compare the receiver operating characteristic (ROC) curves of various spectrum sensing techniques, including our proposed method, under both noise uncertainty (NU) and non-NU scenarios. Furthermore, we assess the probability of detection (P_d) at a fixed false alarm rate (P_{fa}) and compare our approach with existing conventional and deep learning-based spectrum sensing algorithms. We also investigate the robustness and extensibility of our proposed scheme against untrained modulation types and antenna configurations to demonstrate its generalization capability. Finally, we conduct an ablation study to examine the contribution of individual components in our framework to overall performance.

Algorithm 1 SCAE-H-CSG Spectrum Sensing Training Algorithm.

Initialization: Received signals $\mathbf{r}$ from SU, transmitted signals $\mathbf{x}$ without corruption, labels for training the sensing model and DL network hyper-parameters and initial model weights $\boldsymbol{\phi}$, $\boldsymbol{\theta}$.

Phase: 1. Training of denoising network SCAE

Input: Received signals $\mathbf{r}$, ground truth clean signal $\mathbf{x}$, and SCAE parameters $\boldsymbol{\theta}$.
while epochs $\in$ predefined number, stop criterion not met **do**
    Shuffle the training dataset, initialize parameters $\boldsymbol{\theta}$,
    and partition into mini-batches of size B.
    for each mini-batch i **do**
        Compute the MSE loss $\mathcal{L}_{SCAE}$ by Equation (3.3).
        Compute the gradient of all weights.
        Update weights based on the Adam optimizer.
    end for
end while
Save the trained SCAE weights, reconstruct the received signals to form dataset $\mathcal{D}^{Re}$.

Phase: 2. Training of H-CSG sensing network

Input The reconstructed signals $\hat{\mathbf{r}}_i$, and their corresponding ground truth hypotheses H_0 or H_1.
Shuffle the training dataset, and split with min-batch size B.
while iterations $\in$ predefined number, stop criterion not met **do**
    Shuffle the training dataset, reinitialize parameters $\boldsymbol{\phi}$,
    and partition into mini-batches of size B.
    for each mini-batch i **do**
        Compute the CE loss $\mathcal{L}_{CE}$ by Equation (3.10).
        Compute gradients of all weights.
        Update weights based on the Adam optimizer.
    end for
end while
Assess the performance of denoising and detection, test robustness of the trained SCAE-H-CSG network.
return Well-trained parameters of SCAE denoising model and H-CSG sensing model.

3.4.1 Dataset Generation

Our dataset is built upon the widely used RadioML2016.10a dataset [104], which is commonly employed for modulation classification tasks. We consider 6 modulation types, namely, Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 8-Phase Shift Keying (8PSK), 4-Level Pulse Amplitude Modulation (PAM4), 16-Quadrature Amplitude Modulation (QAM16), and 64-Quadrature Amplitude Modulation (QAM64), to cover a diverse range of modulation schemes. The dataset consists of baseband I/Q signal vectors and considers common radio communication impairments such as frequency offset, time delay, and sample rate drift. In this dataset, a root raised cosine filter is employed for pulse shaping. We re-synthesize this publicly available dataset to generate modulated signals with an increased sampling length of 1024. The dataset parameters are summarized in Table 3.3.

TABLE 3.3: Dataset Parameters

Dataset Settings	**Value**
Training samples (per SNR per modulation)	3000
Validation samples (per SNR per modulation)	1000
Evaluation samples (per SNR per modulation)	1000
SNR range	-18∼-5dB
Roll-off factor	0.4
Sample rate	25000 Hz
Maximum sample rate drift	500 Hz
Maximum frequency offset	0.02×sample rate
Maximum time delay	1 symbol

Our training dataset consists of a total of 420,000 signal samples across Rayleigh channels, which are represented by multiplying the clean signal $\mathbf{x}_k$ with a complex Rayleigh fading coefficient h_k containing both amplitude and phase information. The received vector $\mathbf{r}_k$ in the dataset can be expressed as:

$$\mathbf{r}_k = (h_k \cdot \mathbf{x}_k) \cdot b_k + \mathbf{w}_k, \tag{3.16}$$

where $\mathbf{x}_k$ represents the clean modulated signal, b_k and $\mathbf{w}_k$ denote the label of the k-th sample and the noise vector, respectively. The additive white Gaussian noise (AWGN) vectors are generated based on the desired SNR level and follow a CSCG distribution. To create the dataset for our non-cooperative spectrum sensing problem, we assign labels (0 or 1) to the signal samples based on the pre-determined ground truth PU activity. As our primary focus is on the detection performance for weak signals, we consider an SNR range from -18 dB to -5 dB with a 1 dB increment. This SNR range allows us to evaluate the performance of our proposed method under challenging low-SNR conditions.

The input complex-valued signals are divided into real and imaginary parts, i.e., $\mathbf{r}_{\text{in}} = [\mathbf{r}_{\text{real}}; \mathbf{r}_{\text{imag}}]$. The denoised signals with both real and imaginary parts are then fed into the pre-processing network as input, with a dimension of $1 \times 2 \times 1024$. For the received complex-valued signals, data energy normalization is performed as follows:

$$\mathbf{r} = \frac{\mathbf{r}_{\text{in}}}{\sqrt{\sum |\mathbf{r}_{\text{in}}|^2}}, \tag{3.17}$$

where $\mathbf{r}$ is the normalized signal, and $|\cdot|$ denotes the complex modulus.

3.4.2 Complexity Analysis

In this subsection, we analyze the computational complexity of our proposed SCAE-H-CSG model and compare it with other deep learning-based spectrum sensing methods.

For the SCAE network, which consists of several convolutional and deconvolutional layers, the computational complexity for one signal sample can be expressed as $O\left(\sum_{l=1}^{D} C_{l-1} \cdot C_l \cdot K_l \cdot M_l\right)$, where D represents the total number of convolutional and deconvolutional layers, C_l denotes the number of channels in the l-th layer, K_l is the kernel size of the l-th convolutional layer, and M_l represents the output feature map size of the l-th convolutional layer [105]. For deconvolutional layers, the complexity follows the same expression, but M_l represents the input feature map size of the l-th deconvolutional layer. By combining the complexities of both convolutional and deconvolutional layers, we can compute the overall computational complexity of the SCAE network for one sample as $O\left(\sum_{l=1}^{6} C_{l-1} \cdot C_l \cdot K_l \cdot M_l\right)$. Based on the

parameter settings of our SCAE network, the floating-point operations (FLOPs) for one sample is approximately 55.54 million.

In the H-CSG network, the complexities of the CNN and residual CNN modules are similar to that of the SCAE network. The computational complexity of the GRU layers for one sample can be expressed as $O\left(\sum_{l=1}^{P} 3H_l \cdot (E_l + H_l)\right)$ [106], where P represents the number of GRU layers, H_l denotes the number of hidden units in the l-th GRU layer, and E_l represents the embedding size of the l-th GRU layer. Lastly, the computational complexity of the dense layer is given by $O(N_{in} \cdot N_d)$, where N_{in} represents the input dimension and N_d represents the number of neurons in the dense layer. Considering both the SCAE pre-processing stage and the H-CSG network, the total FLOPs for one sample in one iteration is approximately 148.81 million.

Table 3.4 presents a comparison of the FLOPs and the number of parameters for each model. It is worth noting that the computational complexity of the ED method is negligible compared to the deep learning-based schemes.

TABLE 3.4: Computation complexities of each DL-based spectrum sensing methods

Model	**FLOPs (million)**	**No. of Parameters (million)**
SCAE-H-CSG	148.81	0.207
DetectNet [42]	36.58	32.18
ResNet [39]	51.93	0.434
CNN [13]	38.4	16.8

* Note that the computation complexity of ED is negligible compared to other DL-based schemes.

3.4.3 Time-Frequency Analysis of Denoising Efficacy

To assess the efficacy of the proposed SCAE-based denoising approach, we employ time-frequency analysis, which provides insights into both the temporal and spectral characteristics of the signals before and after denoising. Time-frequency representations, such as the spectrogram, offer a visual and quantitative means

to evaluate the performance of denoising algorithms by highlighting the energy distribution of the signal across both time and frequency domains.

The spectrogram is derived from the squared modulus of the Short-Time Fourier Transform (STFT), which is defined as:

$$STFT_r(t,\nu) = \int_{-\infty}^{+\infty} r(u)h^*(u-t)e^{-j2\pi\nu u}du, \tag{3.18}$$

where $r(u)$ represents the input signal, $h(u-t)$ denotes the window function centered at time t, and ν represents the frequency component. The spectrogram, denoted as $S_r(t,\nu)$, is obtained by computing the squared modulus of the STFT:

$$S_r(t,\nu) = \left| \int_{-\infty}^{+\infty} r(u)h^*(u-t)e^{-j2\pi\nu u}du \right|^2. \tag{3.19}$$

Figure 3.4 presents the time-frequency analysis of an example QPSK signal at an SNR of 0 dB, showcasing the denoising performance of the proposed SCAE approach. The spectrograms depict the signals at different stages: (a) before SCAE denoising, (b) after SCAE denoising, and (c) the reference clean signal.

In Figure 3.4(a), the spectrogram reveals a significant presence of noise across the frequency spectrum, evidenced by the erratic and dispersed energy patterns. This visual clutter indicates a high level of signal distortion and spectral leakage, obscuring the underlying signal characteristics. After applying the proposed SCAE-based denoising approach, the time-frequency plot in Figure 3.4(b) demonstrates a substantial reduction in noise. The energy distribution is more focused and aligned with the expected characteristics of a clean QPSK signal, demonstrating the SCAE's effectiveness in suppressing noise while retaining the integrity of the signal's spectral properties. In Figure 3.4(c), the spectrogram of the clean QPSK signal serves as a benchmark for comparison. It exhibits minimal noise and a clear, concentrated energy distribution, which is the ideal outcome of an effective denoising process.

To further demonstrate the robustness and generalizability of the SCAE-based denoising approach, we present additional examples of its impact on different modulation schemes. Figure 3.5 and Figure 3.6 showcase the denoising performance of the SCAE model on randomly selected 8PSK and QAM64 signals, respectively.

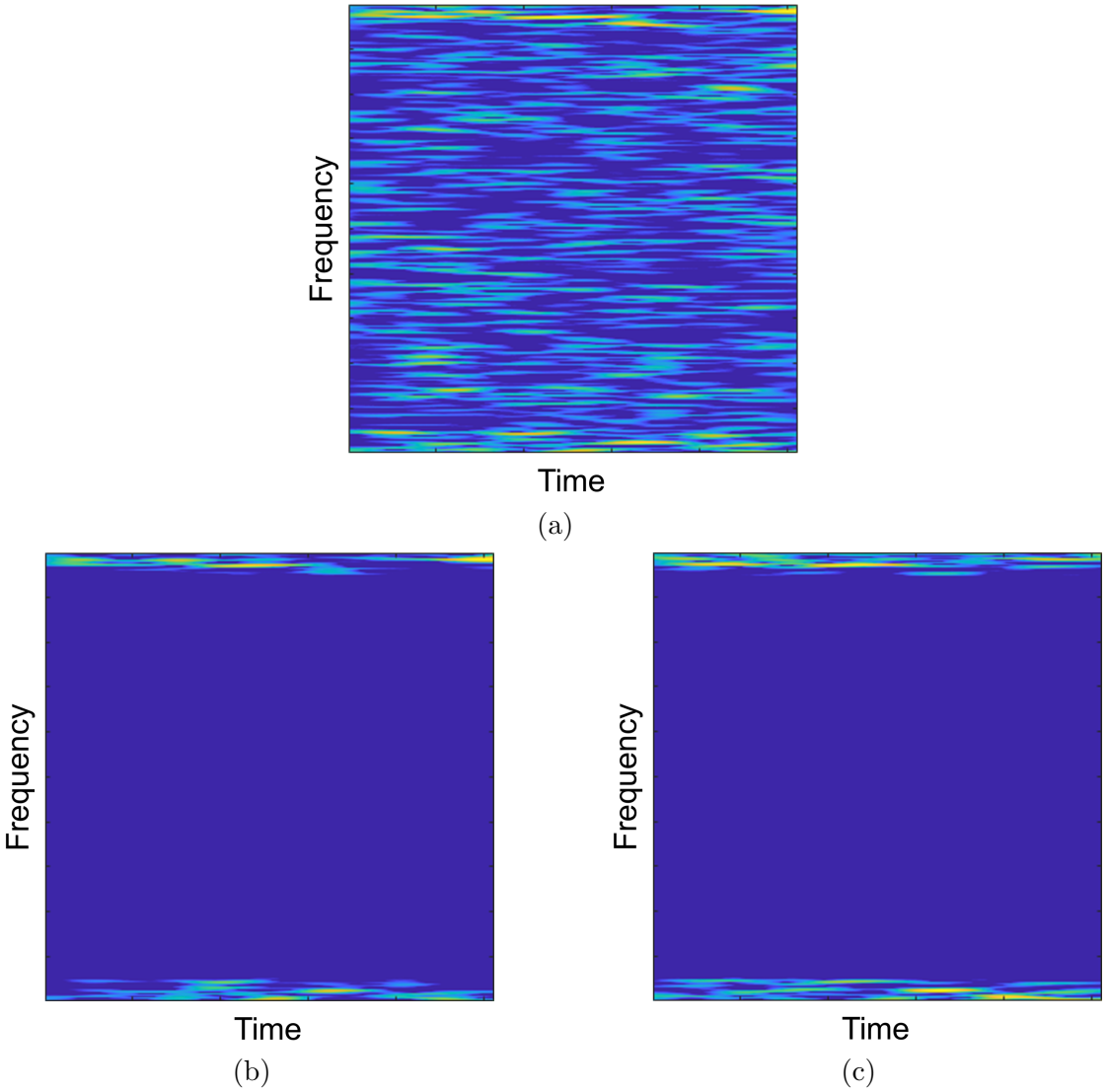
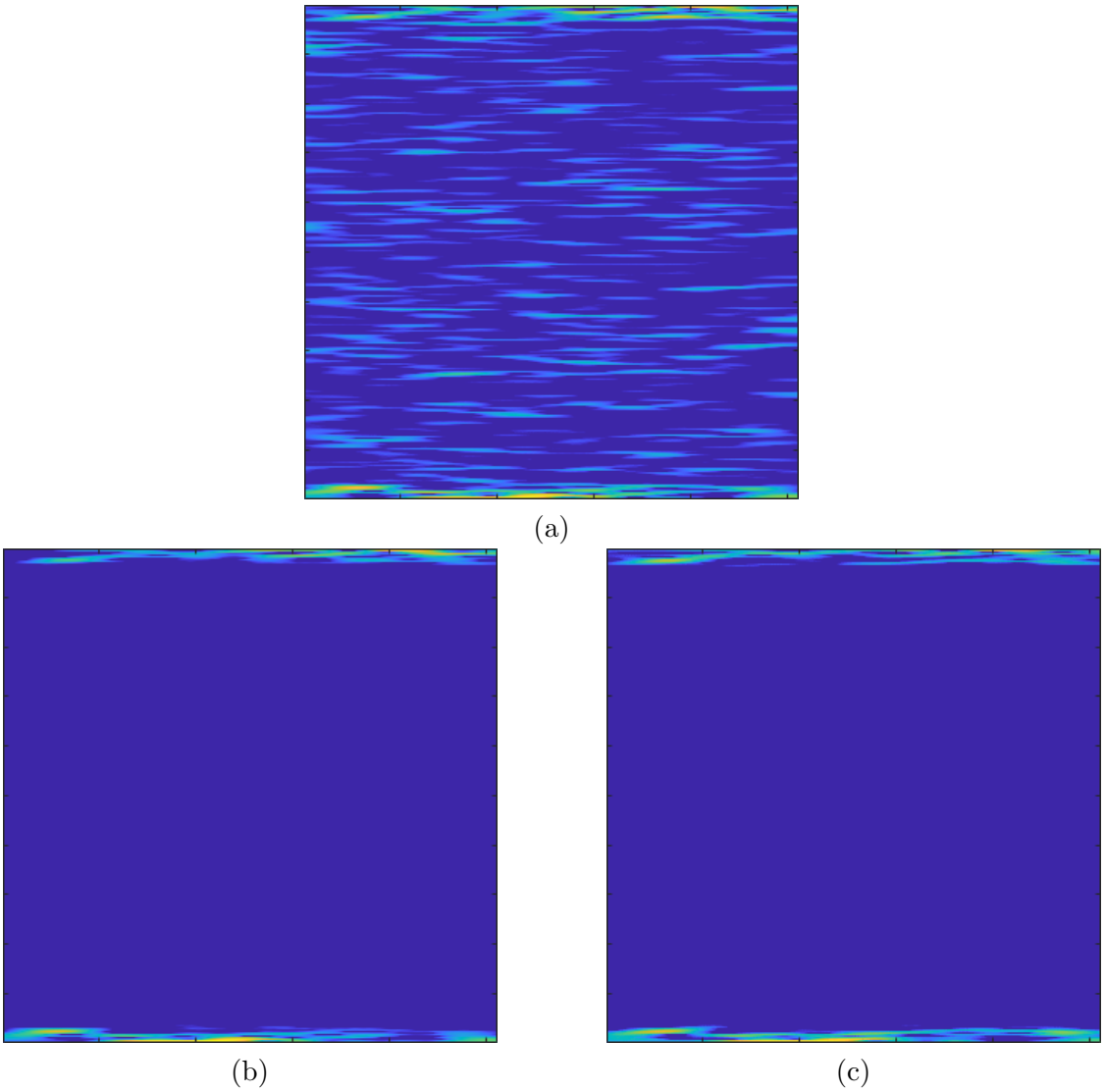


FIGURE 3.4: Time-frequency analysis of a QPSK signal. (a) Before applying SCAE; (b) After applying SCAE; (c) Clean reference QPSK signal.

In both cases, the spectrograms of the denoised signals exhibit a significant reduction in noise compared to their noisy counterparts. The SCAE model effectively suppresses the noise while retaining the essential spectral characteristics of the signals.

To quantitatively assess the efficacy of the SCAE denoising process, we employed the structural similarity index measure (SSIM), a metric that evaluates the perceived quality of a processed image compared to its original counterpart [107]. SSIM values range from 0 to 1, where 0 indicates no structural correlation between the processed and original images, suggesting significant structural differences, and 1 indicates perfect similarity, implying that the images are nearly identical in terms



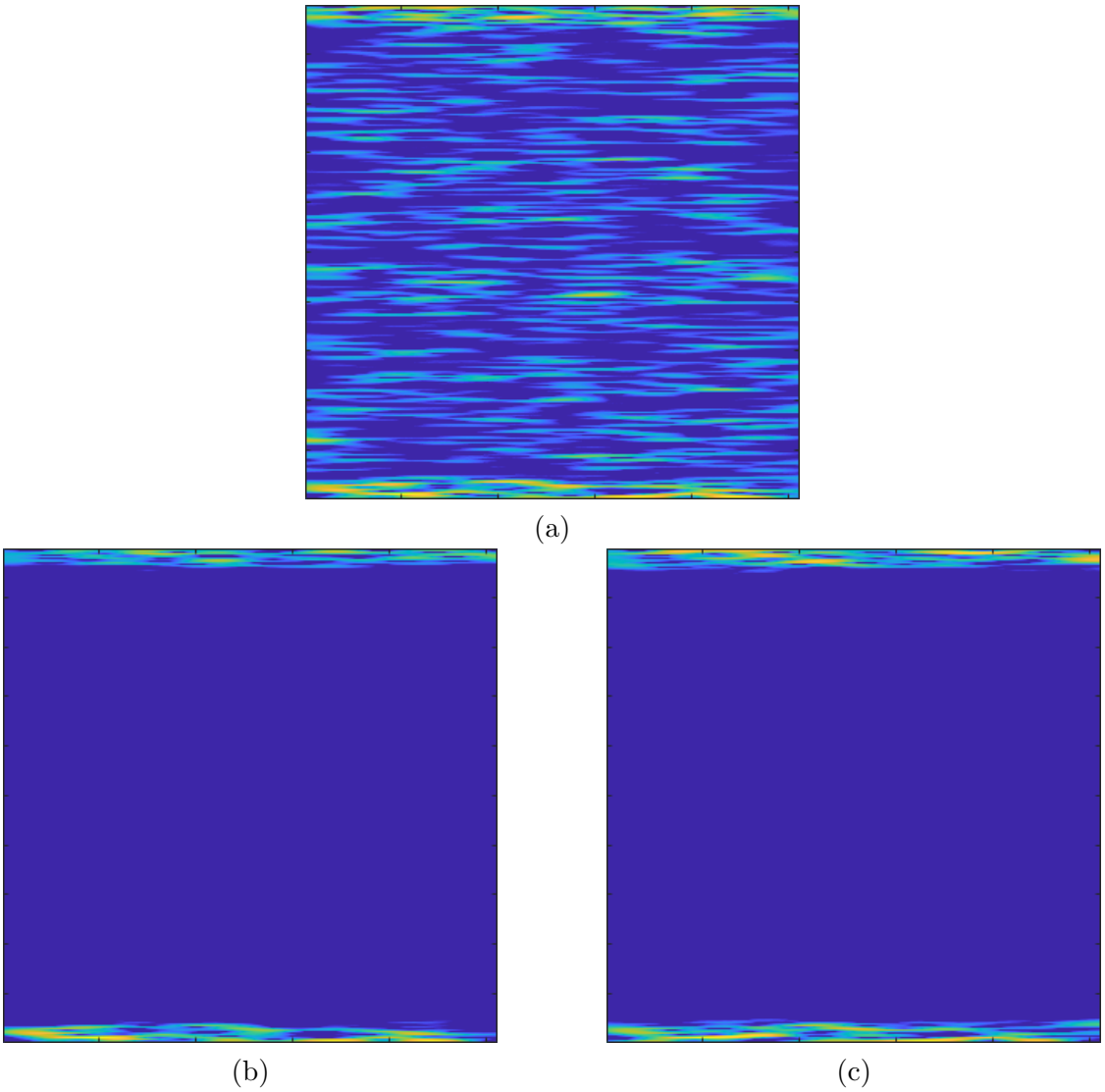
(a)

(b) (c)

FIGURE 3.5: Time-frequency analysis of an 8PSK signal. (a) Before applying SCAE; (b) After applying SCAE; (c) Clean reference 8PSK signal.

of structural content and visual quality. Unlike traditional metrics such as MSE or peak signal-to-noise ratio (PSNR), SSIM considers variations in luminance, contrast, and structural information, such as edge similarity and high-frequency content, which provides a more perceptually relevant assessment [108, 109].

Table 3.5 presents the SSIM values for above three example signals: QPSK, 8PSK, and QAM64. The table compares the SSIM between the noisy signals and clean signals, as well as between the reconstructed signals and clean signals. It demonstrates the significant improvement in signal quality achieved by the SCAE model. For QPSK signals, the SSIM increased from 0.1442 to 0.9498, indicating a substantial enhancement in structural similarity. Similar improvements are observed



(a)

(b)

(c)

FIGURE 3.6: Time-frequency analysis of a QAM64 signal. (a) Before applying SCAE; (b) After applying SCAE; (c) Clean reference QAM64 signal.

for 8PSK and QAM64 signals. These high SSIM values for the reconstructed signals across different modulation schemes underscore the SCAE model's ability to effectively denoise signals while preserving their essential characteristics.

It is worth noting that in some instances, a small portion of the useful signal energy may be inadvertently removed during the denoising process. However, this minor loss of energy does not significantly impact the overall spectrum sensing performance. The primary objective of the denoising stage is to enhance the signal quality by reducing the noise level, thereby facilitating more accurate and reliable spectrum sensing decisions. The consistent improved performance as demonstrated

TABLE 3.5: SSIM Comparison for Signal Reconstruction

Set	SSIM (Noisy vs. Clean)	SSIM (Reconstructed vs. Clean)
QPSK	0.1442	0.9498
8PSK	0.3563	0.9761
QAM64	0.1904	0.9643

by both the spectrogram visualizations and the SSIM metrics, validates the effectiveness of our SCAE-based denoising approach.

3.4.4 Performance Evaluation

In this section, we present a comprehensive evaluation of the proposed H-CSG network's performance in comparison to existing conventional and deep learning-based approaches for spectrum sensing over Rayleigh fading channels. The comparative methods include ED [110], CNN [13], ResNet [39], and DetectNet [42]. To ensure a fair comparison, all schemes utilize the same dataset comprising signals modulated using six different modulation schemes, each with a sample length of 1024. The optimal hyperparameters for each model were determined through extensive simulations.

Figure 3.7 illustrates the ROC curves for various spectrum sensing approaches at an SNR of -14 dB. The results demonstrate that the proposed H-CSG model significantly outperforms other models across all P_{fa} levels. Notably, when P_{fa} is set to 10%, the P_d of the H-CSG scheme exceeds 91%, which is approximately 13% higher than DetectNet (around 78%). This substantial improvement in detection performance underscores the effectiveness of our proposed approach in challenging low SNR conditions. To assess the robustness of the proposed scheme against SNR variations and NU, we evaluated the performance under conditions where the noise power fluctuates over time. In this scenario, the estimated noise power $\hat{\sigma}_w^2$ is drawn from the interval $\left[\frac{1}{\varepsilon}\sigma_w^2, \varepsilon\sigma_w^2\right]$, where σ_w^2 represents the actual noise power, ε denotes the NU factor. The ROC curves of different spectrum sensing approaches in the NU scenario are also shown in Figure 3.7, where "NU-2dB" represents the case of 2dB NU, i.e., $\varepsilon = 1.58$. We can observe our proposed model is robust to SNR. In contrast, the performance of ED deteriorates significantly in the presence of NU. It is worth noting that DL-based methodologies such as CNN, ResNet, and DetectNet

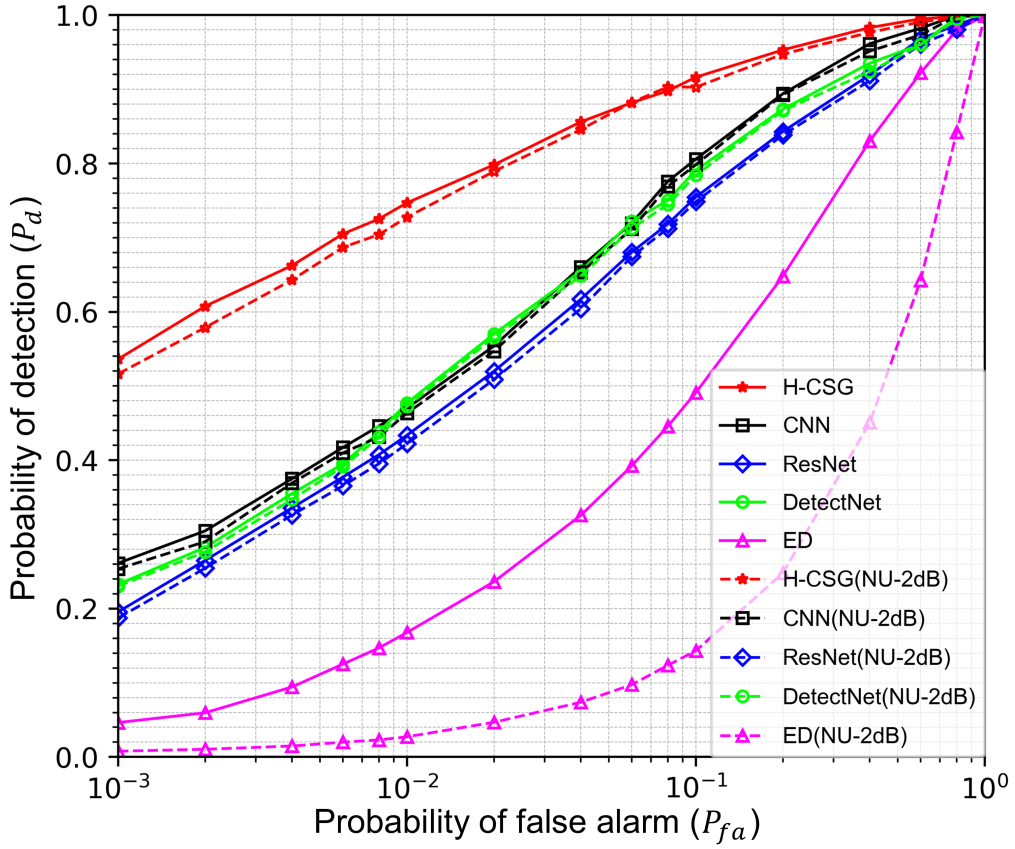


FIGURE 3.7: The ROC curves of different approaches at SNR=-14 dB.

do not require prior noise information for PU signal detection, which contributes to their resilience against NU.

Figure 3.8 presents the detection probability of all neural network models as a function of SNR, with P_{fa} fixed at 10%. The proposed H-CSG network demonstrates remarkable performance, achieving approximately 55% detection accuracy over Rayleigh channels at an extremely low SNR of -18 dB. In comparison, the ED approach is difficult to detect useful signals precisely in this challenging scenario. The impact of the SCAE pre-processing step is illustrated by the red dotted curve, showing a significant improvement of around 10% in P_d at low SNR levels compared to the H-CSG model without pre-processing. This improvement highlights the effectiveness of the SCAE in enhancing signal quality for subsequent detection. Furthermore, MFD assumes perfect knowledge of signal and channel characteristics. As expected, MFD achieves the best detection performance across all SNR ranges, serving as a benchmark for "optimal" performance in this context. However, its reliance on perfect channel estimation, equalization, and signal synchronization makes it impractical in real-world cognitive radio environments.

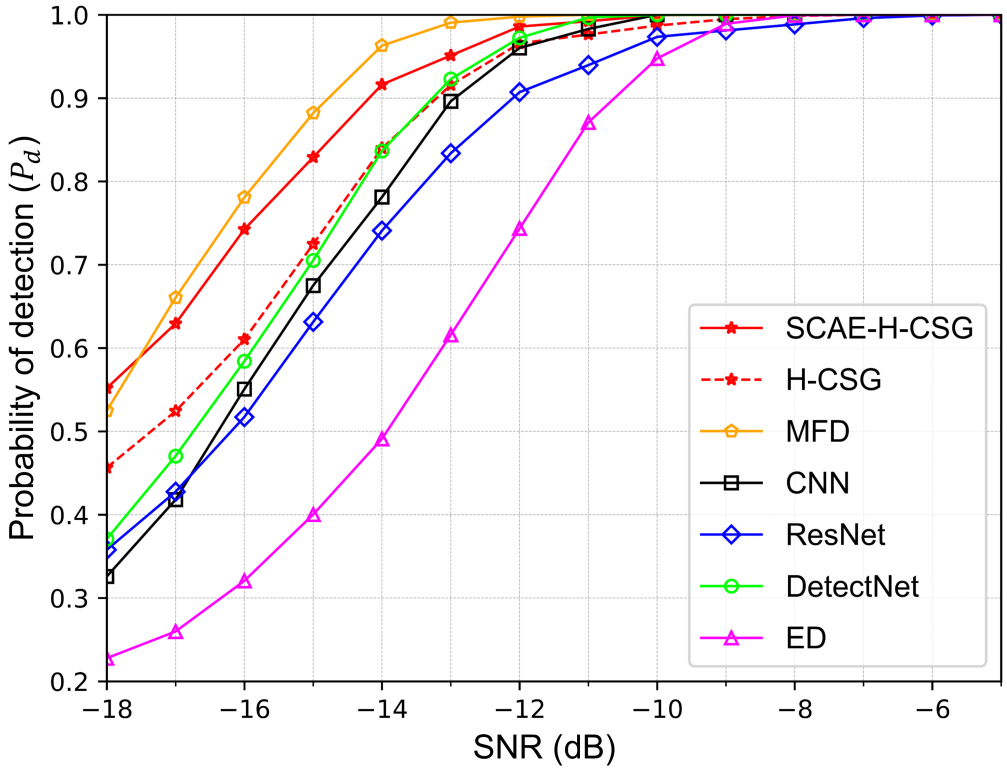


FIGURE 3.8: The probability of detection of different approaches at P_{fa}=10%.

The proposed deep learning-based method bypasses these steps, offering a more feasible and robust solution under dynamic and non-cooperative conditions. Additionally, we have examined high SNR scenarios (above -5 dB) and found that the P_d consistently approaches 100% for the proposed method. These results collectively demonstrate that the H-CSG algorithm can reliably detect spectrum occupancy across a wide range of SNR conditions, including scenarios with noise uncertainty.

3.4.5 Robustness and Extensibility Analysis

To evaluate the robustness of our proposed DL-based spectrum sensing approach, we conduct two experiments. First, we assess the performance of the SCAE-H-CSG model in detecting untrained modulation types. Second, we extend the original SISO-based SCAE-H-CSG model to a SIMO configuration and analyze its performance under various channel conditions.

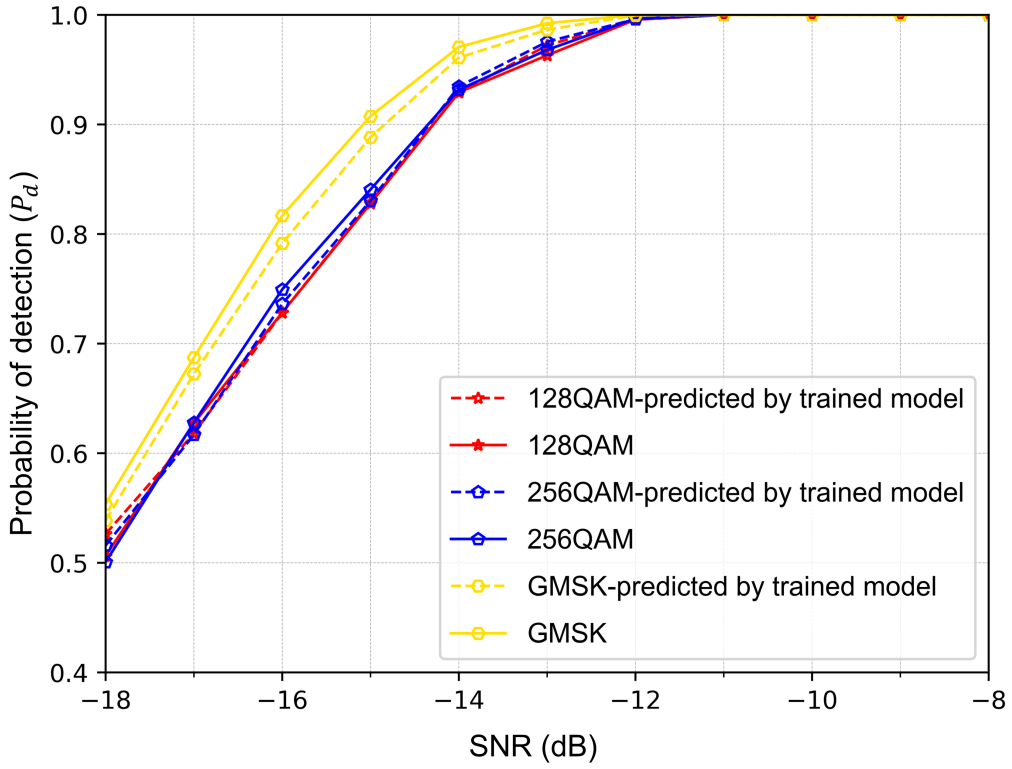


FIGURE 3.9: Robustness of the proposed scheme against untrained modulation types.

3.4.5.1 Robustness against modulation types

We have conducted experiments using modulation schemes that were not included in the training dataset. Specifically, we tested the model's performance on Gaussian Minimum Shift Keying (GMSK), 128-Quadrature Amplitude Modulation (128QAM), and 256-Quadrature Amplitude Modulation (256QAM) signals. Figure 3.9 presents the P_d as a function of SNR for these untrained modulation types at a fixed $P_{fa} = 10\%$. The results demonstrate remarkable consistency between the performance of the model when retrained on these unseen modulations (solid lines) and the performance predicted by the model trained on our original dataset $\mathcal{D}^s$ (dashed lines). The consistent performance across these diverse and previously unseen modulation schemes underscores the robustness and generalizability of the proposed H-CSG network. This characteristic is crucial for practical spectrum sensing applications, where the specific modulation scheme of the primary user may be unknown or variable. The ability of the model to maintain high detection accuracy without retraining for new modulation types demonstrates its potential

for deployment in dynamic spectrum environments where adaptability to various signal types is essential.

3.4.5.2 Extensibility on SIMO system

Secondly, although the SCAE-H-CSG model is originally designed for SISO systems, its architecture and learning capabilities make it adaptable to SIMO scenarios. In the second experiment, we consider a SIMO system with N_r receive antennas, where the received signal vectors $\mathbf{y}_m \in \mathbb{C}^{N_r \times L}$ can be expressed as:

$$\mathbf{y}_m = \begin{bmatrix} \mathbf{r}_m^0 \\ \mathbf{r}_m^1 \\ \vdots \\ \mathbf{r}_m^{N_r-1} \end{bmatrix} = \begin{bmatrix} h_m^0 \\ h_m^1 \\ \vdots \\ h_m^{N_r-1} \end{bmatrix} \cdot \mathbf{x}_m + \begin{bmatrix} \mathbf{w}_m^0 \\ \mathbf{w}_m^1 \\ \vdots \\ \mathbf{w}_m^{N_r-1} \end{bmatrix}, \tag{3.20}$$

where h_m^i represents the complex channel gain and AWGN between the transmit antenna and the i-th receive antenna of the m-th signal sample, and $\mathbf{w}_m^i$ is the noise at the i-th receive antenna.

To combine the received signals from multiple antennas, we employ maximum ratio combining (MRC), which is a widely used diversity combining technique that maximizes the SNR at the receiver. In our analysis, we assume perfect channel state information (CSI) at the receiver. Under this ideal condition, the MRC combiner output $\mathbf{y}_m^{MRC}$ can be expressed as:

$$\mathbf{y}_m^{MRC} = \sum_{i=0}^{N_r-1} {h_m^i}^* \cdot \mathbf{r}_m^i, \tag{3.21}$$

where ${h_m^i}^*$ represents the complex conjugate of the channel gain for the i-th receive antenna, and $\mathbf{r}_m^i$ is the received signal at the i-th receive antenna. This assumption of perfect CSI allows us to analyze the theoretical upper bound of the system performance. MRC effectively weights the received signals based on their respective channel gains, giving higher weights to stronger signals and lower weights to weaker signals, thereby maximizing the overall SNR.

To evaluate the performance of the SCAE-H-CSG model in the SIMO configuration, we consider two channel models, including independent and correlated channels. In

the correlated channel model, the channel vector $\mathbf{h}$ is generated using a correlation matrix $\mathbf{R} \in \mathbb{C}^{N_r \times N_r}$, which captures the spatial correlation between the receive antennas. The correlated channel vector can be obtained as:

$$\mathbf{h} = \mathbf{R}^{1/2} \cdot \mathbf{h}_{\text{iid}}, \tag{3.22}$$

where $\mathbf{h}_{\text{iid}} \in \mathbb{C}^{N_r \times 1}$ is an i.i.d. channel vector, and $\mathbf{R}^{1/2}$ is the square root of the correlation matrix $\mathbf{R}$. The exponential correlation model is adopted to characterize the spatial correlation between the receive antennas. The (i, j)-th element of the correlation matrix $\mathbf{R}$ is given by:

$$R_{i,j} = \rho^{|i-j|}, \tag{3.23}$$

where ρ is the correlation coefficient, which quantifies the degree of correlation between the receive antennas.

In our experiments, we consider a SIMO system with $N_r = 2$ receive antennas and evaluate the performance of the SCAE-H-CSG model under independent channels and correlated channels with correlation coefficients of 0.1, 0.5, and 0.9. The detection performance is assessed in terms of the P_d at a fixed P_{fa} of 10%.

Figure 3.10 presents the probability of detection curves for the SIMO configuration under different channel conditions. The results demonstrate that the SIMO system with MRC combining achieves a performance improvement of up to nearly 3dB SNR when compared to the SISO case. This gain aligns with the theoretical improvement in SNR when employing the MRC combining method at the receiving end. The independent channel scenario yields the best performance, capitalizing on the full diversity gain offered by the multiple receive antennas. As the correlation coefficient increases, the detection performance gradually degrades. This behavior is expected, as higher correlation reduces the effective diversity gain. Nevertheless, even in the presence of strong spatial correlation where correlation coefficient is 0.9, the SIMO configuration still outperforms the SISO case, highlighting the benefits of multiple receive antennas in spectrum sensing and simultaneously validate the extensibility of the proposed scheme from SISO to SIMO systems.

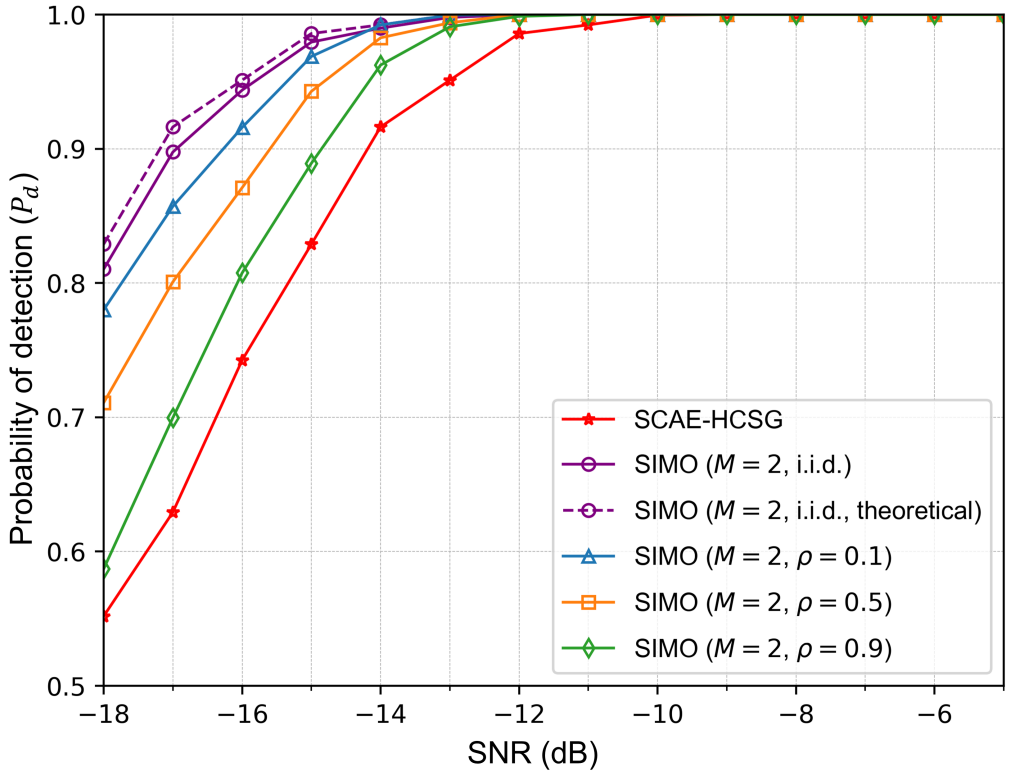


FIGURE 3.10: Extensibility of the proposed scheme on different antenna configurations and channel conditions.

3.4.6 Ablation Study

In this section, we present an in-depth ablation study to investigate the impact of various network components and architectural choices on the performance of our proposed deep learning-based spectrum sensing model. The primary objective of this study is to gain insights into the contribution of each component and to validate the effectiveness of our design decisions.

We perform two sets of ablation experiments, each focusing on different aspects of our proposed model. The first experiment aims to assess the impact of key architectural components, including residual connections, the GRU module, the self-attention mechanism, and the pre-processing stage utilizing the SCAE network. By systematically removing or modifying these components, we can quantify their individual contributions to the model's performance and identify the most critical elements of our architecture. The second set of experiments is designed to evaluate the compatibility of the SCAE on other DL architectures.

In the first experiment, we begin by establishing a baseline network architecture, which consists of a simple two-layer CNN extracted from our proposed H-CSG network. This baseline serves as a starting point for our experiments, allowing us to isolate and evaluate the effects of individual components on the overall sensing performance. Table 3.6 presents the probability of detection P_d achieved by different network configurations at various SNR levels, ranging from -18 dB to -13 dB. It is observed that the baseline CNN achieves a P_d of 27.58% at -18 dB SNR, which gradually improves to 78.6% as the SNR increases to -13 dB. Next, we increase the number of CNN layers and incorporate residual connections, resulting in a deeper residual CNN configuration. The addition of CNN layers and residual connections significantly enhances the sensing performance across all SNR levels. For instance, at -18 dB SNR, the P_d improves from 27.58% to 36.21%. To further investigate the impact of temporal modeling, we introduce the GRU module into the residual CNN architecture, resulting in the H-CSG configuration without SA module. The incorporation of GRU enhances the model's ability to capture temporal dependencies in the signal, leading to improved sensing performance. Moreover, the H-CSG configuration, which integrates the self-attention mechanism, further boosts the sensing performance. The self-attention mechanism allows the model to focus on the most informative parts of the input sequence, enhancing its discriminative capabilities. Finally, we incorporate the SCAE pre-processing stage into the H-CSG configuration, resulting in the complete SCAE-H-CSG model. The SCAE pre-processing stage plays a crucial role in denoising the received signals, enhancing the overall sensing performance. The SCAE-H-CSG model achieves the highest P_d values across all SNR levels, significantly outperforming all other configurations.

TABLE 3.6: probability of detection (%) of different components in SCAE-H-CSG model under each SNR level.

Component	**-18dB**	**-17dB**	**-16dB**	**-15dB**	**-14dB**	**-13dB**
Baseline CNN	27.58	35.63	40.58	54.13	65.79	78.60
Residual CNN	36.21	43.65	55.60	68.38	79.01	89.78
Residual CNN + GRU	40.51	49.24	59.71	70.66	82.66	91.29
H-CSG	45.63	52.42	61.04	72.46	83.98	91.56
SCAE-H-CSG	**55.18**	**62.93**	**74.25**	**82.89**	**91.63**	**95.11**

In the second experiment, we investigate the efficacy and compatibility of the SCAE as a pre-processing method for spectrum sensing. Figure 3.11 illustrates the comparative performance of our proposed H-CSG network and DetectNet, both with and without SCAE pre-processing, across a range of SNR values. The results demonstrate that the incorporation of SCAE as a pre-processing step leads to performance improvements for both architectures. For the H-CSG network, the addition of SCAE results in a consistent enhancement in detection performance across all SNR levels, with the improvement being particularly pronounced in the low SNR regime. The H-CSG with SCAE outperforms its counterpart without SCAE by approximately 1.5 dB at lower SNRs. Similarly, DetectNet also benefits from the SCAE pre-processing, showing improved detection performance across the SNR range. These findings demonstrate the efficacy and compatibility of SCAE as a pre-processing method. The performance improvements observed across different network architectures underscore the versatility of SCAE in enhancing signal quality for spectrum sensing tasks. Notably, while both H-CSG and DetectNet benefit from SCAE pre-processing, the H-CSG network shows a more significant performance gain, particularly in the challenging low SNR regime, which demonstrates the synergistic effect of integrating SCAE within the H-CSG framework. However, this performance improvement comes with an increase in computational complexity. The inclusion of the SCAE stage adds 55.54 million FLOPs to the model, bringing the total computational complexity of H-CSG to 148.81 million FLOPs. Similarly, DetectNet, which has a baseline complexity of 36.58 million FLOPs, also experiences an increase due to SCAE. While this additional complexity enhances detection performance, particularly in low SNR scenarios, it introduces challenges for real-time applications in resource-constrained environments. For such scenarios, DetectNet without SCAE could serve as a viable alternative, balancing lower complexity with moderate performance. Thus, the choice of model should consider the specific application requirements, balancing computational efficiency and sensing accuracy.

In conclusion, the ablation study provides valuable insights into the design choices and component interactions within our proposed DL-based spectrum sensing model. By systematically evaluating the impact of residual connections, GRU, self-attention, and SCAE pre-processing, we can validate the effectiveness of each component and their collective contribution to the model's superior performance. Furthermore, the comparison of DetectNet with and without SCAE pre-processing highlights

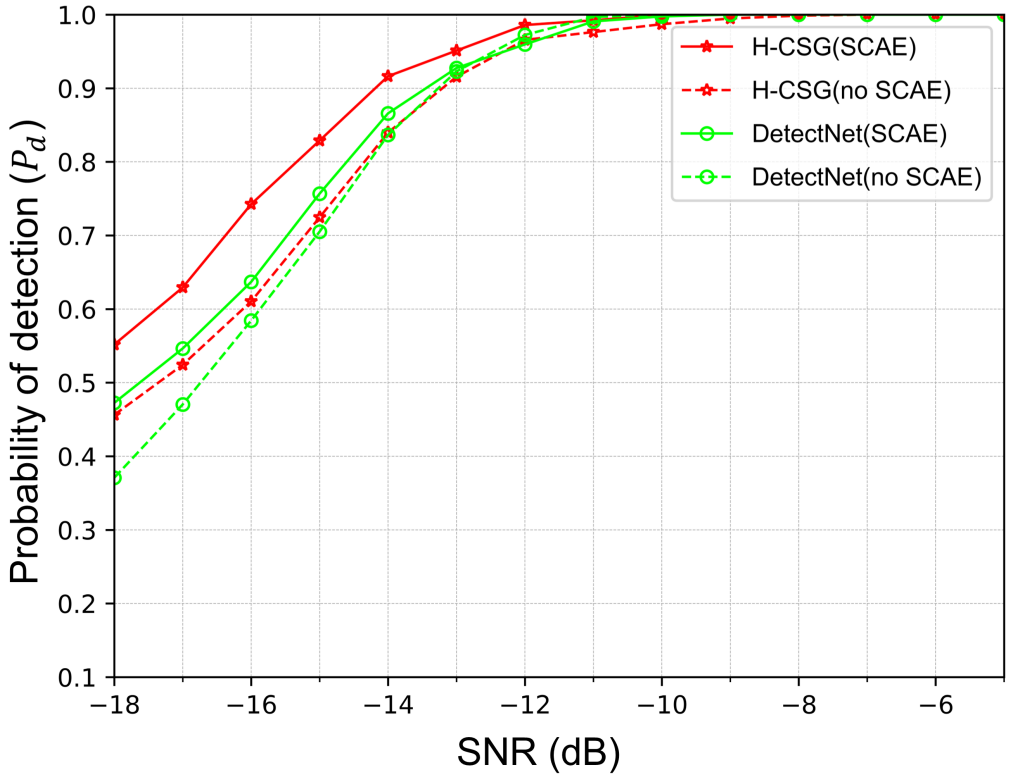


FIGURE 3.11: Compatibility of SCAE on DetectNet.

the importance of the denoising stage in enhancing sensing performance. However, the inclusion of SCAE introduces additional computational complexity, which may limit its applicability in resource-constrained environments. For such scenarios, models without SCAE, such as DetectNet alone, may serve as computationally efficient alternatives with moderate performance. Therefore, this study underscores the importance of balancing computational efficiency with sensing accuracy when designing spectrum sensing systems for real-world applications.

3.5 Conclusion

In this chapter, we have explored the application of deep learning techniques to address the challenging problem of non-cooperative spectrum sensing in scenarios where the desired modulated signal is buried below the noise floor. We have proposed a novel two-stage approach, consisting of a SCAE pre-processing step followed by a H-CSG sensing network. This innovative architecture is designed to be robust to varying SNR conditions and untrained modulation types, enabling reliable spectrum sensing in a wide range of practical situations. The proposed SCAE

pre-processing stage has demonstrated its effectiveness in efficiently compressing and encoding the input signal data, as evidenced by the improved sensing performance in the subsequent stage, even under challenging low SNR conditions. By reconstructing the denoised signal from the compressed representation, the SCAE enhances the signal quality and facilitates improved detection performance in the subsequent sensing network. The H-CSG network, which combines the strengths of CNNs, SA mechanisms, and GRUs, has shown remarkable ability to extract correlated information from each channel and signal data bit. This comprehensive feature extraction capability allows the H-CSG network to make accurate spectrum sensing decisions, even in challenging low SNR environments.

Through extensive experimentation, we gain several valuable insights about the practical implementation of deep learning for spectrum sensing. Although our model is initially trained on specific modulation schemes, it illustrates robust performance when tested on untrained modulations like GMSK and 128QAM. This generalization capability suggests that the model learns fundamental signal characteristics rather than modulation-specific features. The extension from SISO to SIMO systems further validates the architecture's flexibility in leveraging spatial diversity without requiring significant modifications. Additionally, the compatibility of SCAE with other architectures like DetectNet also demonstrates the broader applicability of our denoising approach in signal processing tasks.

However, it is important to acknowledge the limitations of the current two-stage approach. While the SCAE pre-processing step effectively denoises the input signal, it introduces additional computational complexity and may not fully exploit the potential synergies between the denoising and sensing tasks. Additionally, the flat Rayleigh fading channel model used in this chapter simplifies real-world scenarios and may not fully highlight the advantages of deep learning-based methods over traditional techniques in dynamic environments. Hence, in subsequent chapters, we will evaluate the proposed methods under more complex channel models, including TDL-E channels, which incorporate multipath fading and time variations. Moreover, the separate training of the SCAE and H-CSG networks may lead to suboptimal performance, as the two stages are not jointly optimized, and the sensing network may be highly affected by the performance of denoising stage. To address these limitations and further enhance the spectrum sensing performance,

in the next chapter, we will explore a unified deep learning framework that combines the denoising and sensing tasks into a single end-to-end network. By jointly optimizing the denoising and sensing objectives, we aim to develop a more efficient and effective approach that can directly learn the mapping from noisy input signals to accurate spectrum sensing decisions. This unified framework has the potential to streamline the spectrum sensing pipeline, reduce computational complexity, and improve the overall performance by leveraging the inherent correlations between the denoising and sensing tasks.

In conclusion, the work presented in this chapter has demonstrated the immense potential of deep learning techniques in tackling the challenging problem of non-cooperative spectrum sensing in low SNR environments. The proposed SCAE-H-CSG model has showcased superior performance and robustness compared to traditional and other DL approaches. However, the limitations of the two-stage approach motivate us to explore an end-to-end deep learning framework in the next chapter, aiming to further enhance the efficiency and effectiveness of spectrum sensing in cognitive radio networks.

Chapter 4

Joint Optimization Framework for Denoising and Spectrum Sensing

4.1 Introduction

In the previous chapter, we explored a two-stage deep learning approach for non-cooperative spectrum sensing in cognitive radio networks, which consists of a denoising stage using a SCAE followed by a sensing stage using H-CSG network. While this approach demonstrated superior performance compared to traditional methods, it has certain limitations. The separate training of the denoising and sensing stages may lead to suboptimal performance, as the sensing network's performance is highly dependent on the denoising stage. Moreover, the two-stage approach introduces additional computational complexity and may not fully exploit the potential synergies between the denoising and sensing tasks.

In the context of signal modulation and channel modeling, the previous chapter focused on single carrier modulation schemes such as QPSK, BPSK, and QAM, with the Rayleigh channel represented by a complex channel gain h, which follows a CSCG distribution. However, to align our research with the latest developments in wireless communications, this chapter shifts the focus to OFDM, which has become the dominant modulation scheme in modern wireless communication systems due to its robustness against frequency-selective fading and its ability to support high data rates. Moreover, we adopt the fifth generation (5G) tapped-delay line (TDL) channel model, specifically the TDL-E model, to characterize the temporal and

spatial characteristics of wireless channels in various environments, as it has been widely adopted in the industry and research community.

To address the limitations of the two-stage approach and further enhance spectrum sensing performance, we propose a joint denoising and spectrum sensing (JDSS) network that combines the denoising and sensing tasks into a single end-to-end deep learning framework. By jointly optimizing the denoising and sensing objectives, we aim to develop a more efficient and effective approach that can directly learn the mapping from noisy input signals to accurate spectrum sensing decisions. The main contributions of this chapter are summarized as follows:

- We design an end-to-end non-cooperative spectrum sensing scheme with signal enhancement, which enables joint learning of denoising and detection.
- The network's bottleneck consists of several self-attention blocks, capable of capturing intrinsic correlations between the subcarriers of OFDM signals under the 5G TDL channel environment.
- We introduce an efficient total loss function, which incorporates both l_1 loss and multi-resolution STFT, resulting in improved performance and optimization.
- The proposed JDSS can achieve superior sensing performance by integrating the denoising step into the neural network, outperforming other baseline schemes, including SCAE-H-CSG [62], DetectNet [42], CNN [111], SAE [49], ED [110], and autocorrelation-based detection [112].

The proposed JDSS network offers several advantages over the previous two-stage approach. By jointly optimizing the denoising and sensing objectives, the network can learn more efficient and effective representations that are tailored specifically for the spectrum sensing task. This unified framework streamlines the spectrum sensing pipeline, reduces computational complexity, and improves overall performance by leveraging the inherent correlations between the denoising and sensing tasks.

The remainder of this chapter is organized as follows. Section 4.2 presents the system model and problem formulation. Section 4.3 describes the proposed JDSS network architecture and training strategy, including dataset generation and loss

function design. Section 4.4 discusses the performance evaluation, followed by hyper-parameter analysis, ablation study, and complexity analysis. Finally, Section 4.5 concludes the key contributions of this work.

4.2 System Model and Problem Illustration

This chapter builds upon the non-cooperative spectrum sensing framework introduced in Chapter 3, extending the research to more advanced modulation techniques and channel models that are relevant to modern wireless communication systems, particularly 5G networks. While both chapters consider a scenario where a SU equipped with a single antenna aims to detect the presence of PU signals, the main difference lies in the modulation scheme employed by the PU and the channel model used to characterize the wireless propagation environment.

In this chapter, the PU utilizes OFDM signals, which is a widely adopted modulation scheme in modern wireless communication systems, such as 5G networks, due to its robustness against frequency-selective fading and its ability to support high data rates. OFDM facilitates parallel symbol modulation across N_{sc} subcarriers, with QPSK employed to map the m^{th} randomly generated bit sequence A_m to the complex symbol sequence $C_{m,k}$. The sample $x_{m,n}$ of the m^{th} baseband time-domain OFDM block is given by:

$$x_{m,n} = \sum_{k=0}^{N_{sc}\zeta-1} \frac{1}{\sqrt{N_{sc}\zeta}} e^{j\frac{2\pi kn}{N_{sc}\zeta}} C_{m,k}, \quad -N_c \leq n \leq N_{sc}\zeta - 1, \tag{4.1}$$

where $N_{sc} = 256$ denotes the total number of subcarriers, $\zeta = 4$ is the oversampling factor, resulting in an OFDM block length of $N_{sc}\zeta = 1024$, and $N_c = N_{sc}\zeta/8 = 128$ is the length of the cyclic prefix (CP).

To accurately model the wireless channel in a 5G communication scenario, we adopt the TDL channel model, specifically the TDL-E model, as specified in the 3GPP TR 38.901 release [113]. The TDL-E model incorporates both line-of-sight (LoS) and non-line-of-sight (NLoS) components, making it more representative of real-world 5G channels compared to the simpler Rayleigh channel model used in Chapter 3. The channel coefficients for each tap are generated using a fading

model equation that complies with the TR38.901 release. The baseband discrete-time signal traversing this channel can be expressed as:

$$h_{m,n} = \sum_{l=1}^{L=14} g_{m,l} x_{m,n-l}, \tag{4.2}$$

where l denotes the tap index, and $g_{m,l}$ represents time-dependent coefficients of the l^{th} tap. Besides, $g_{m,l}$ remains invariant within a single OFDM block, but exhibits temporal fluctuations. Moreover, a typical spectrum sensing system can be modeled as a binary classification problem as follows:

$$\begin{cases} H_0 : r_{m,n} = w_{m,n}, \\ H_1 : r_{m,n} = h_{m,n} + w_{m,n}, \end{cases} \tag{4.3}$$

where $r_{m,n}$ denotes the received sample of the m^{th} OFDM vector at the SU end. Note that $w_{m,n}$ denotes the i.i.d. complex AWGN with zero mean and variance σ_w^2.

For conventional spectrum sensing approaches, such as ED and autocorrelation-based detection, the test statistics Υ_{ED} and Υ_{AC} of the m^{th} received signals are given by:

$$\begin{aligned} \Upsilon_{\text{ED}} &= \frac{1}{N_{sc}\zeta} \sum_{n=0}^{N_{sc}\zeta-1} |r_{m,n}|^2, \\ \Upsilon_{\text{AC}} &= \frac{\frac{1}{N_c}\sum_{n=-N_c}^{-1} \bar{R}_{m,n}}{\frac{1}{N_{sc}\zeta}\sum_{n=0}^{N_{sc}\zeta-1} |r_{m,n}|^2}, \end{aligned} \tag{4.4}$$

where $\bar{R}_{m,n} = \mathfrak{Re}\left\{r_{m,n} r^*_{m,n+N_{sc}\zeta}\right\}$ is the real value of the n^{th} sample value product. The test statistics Υ_{ED} and Υ_{AC} are assessed against their respective thresholds η_{ED} and η_{AC}, which are determined through Monte Carlo simulations.

For DL-based spectrum sensing schemes, let the m^{th} OFDM vectors $\mathbf{x}_m$ and $\mathbf{r}_m$ be defined as $\mathbf{x}_m \triangleq [x_{m,0}, \ldots, x_{m,N_{sc}\zeta-1}]$ and $\mathbf{r}_m \triangleq [r_{m,0}, \ldots, r_{m,N_{sc}\zeta-1}]$, representing the transmitted signal from the PU and the received signal at the SU after discarding the CP, respectively. The decision rule for the binary hypothesis testing problem is formulated as:

$$f_{H_1|\boldsymbol{\theta}}(\mathbf{r}_m) \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{4.5}$$

where $f_{H_1|\boldsymbol{\theta}}(\,\cdot\,)$ denotes the likelihood of hypothesis H_1, conditional on the well-trained parameters $\boldsymbol{\theta}$ via the network mapping function $f(\,\cdot\,)$. The state of the m^{th}

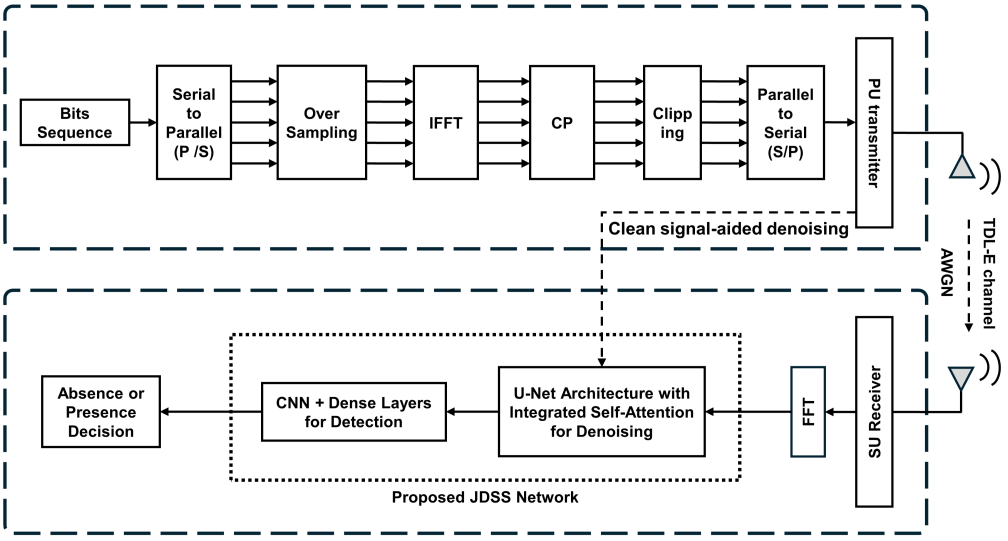


FIGURE 4.1: The overall diagram of the OFDM signal generation and JDSS training process.

PU signal can be determined by comparing $f_{H_1|\boldsymbol{\theta}}(\mathbf{r}_m)$ with the detection threshold η. Conventionally, the probability of detection P_d or the probability of missed detection P_m, and the probability of false alarm P_{fa} are used to assess the sensing performance:

$$\begin{cases} P_d = 1 - P_m = \mathrm{P_r}\left\{ f_{H_1|\boldsymbol{\theta}}(\mathbf{r}) > \eta \mid H_1 \right\}, \\ P_{fa} = \mathrm{P_r}\left\{ f_{H_1|\boldsymbol{\theta}}(\mathbf{r}) > \eta \mid H_0 \right\}, \end{cases} \tag{4.6}$$

where the notation $\mathrm{P_r}\{\cdot\}$ represents the probability of an event.

4.3 Joint Denoising and Spectrum Sensing Scheme

In this section, we will first present the architecture of the proposed JDSS network, outlining the function of each core module. Furthermore, the designed loss function and training procedures are introduced, followed by an illustration of the online detection strategy for identifying PU signals.

4.3.1 JDSS Architecture

Figure 4.1 illustrates the overall training process of the proposed JDSS network, including the generation of OFDM signals. The bits sequence undergoes serial-to-parallel conversion, modulation, oversampling, inverse fast Fourier transform

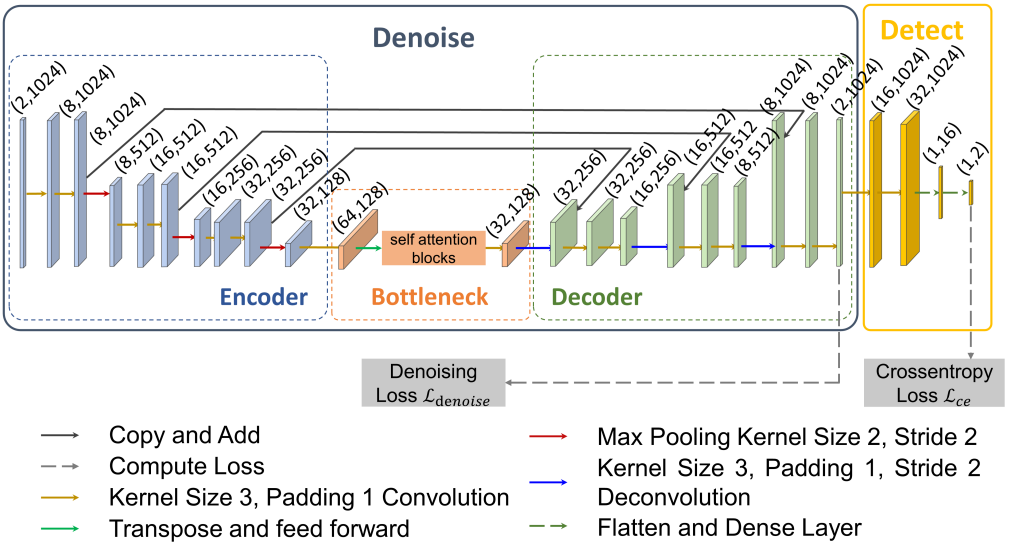


FIGURE 4.2: The proposed JDSS network architecture.

(IFFT), cyclic prefix addition, and clipping before being converted back to serial form for transmission. The generated OFDM signal is then used to train the JDSS network, which consists of a U-Net architecture with integrated self-attention for denoising and a CNN-based detection network. The entire network is trained end-to-end using a carefully designed loss function that balances the denoising and detection tasks, enabling efficient and effective spectrum sensing in low SNR environments.

The proposed JDSS network architecture is based on the U-Net design, which is specifically chosen for OFDM spectrum sensing for several key reasons. Firstly, U-Net's symmetric encoder-decoder structure with skip connections is particularly suitable for joint optimization of denoising and sensing tasks. These skip connections are crucial as they preserve detailed subcarrier information while allowing deeper feature extraction, enabling the network to maintain both high signal reconstruction quality and detection accuracy. Secondly, unlike the sequential processing nature of H-CSG architecture used in Chapter 3, which better suits single-carrier modulation, U-Net's structure is naturally suited to handle OFDM's parallel subcarrier characteristics through its hierarchical processing capabilities. The architecture, as depicted in Figure 4.2, consists of two main segments: the denoising section $\mathcal{E}$ and the sensing section $\mathcal{G}$. The denoising section is based on the U-Net architecture [114], comprising an encoder, a decoder, and a bottleneck. The encoder and decoder contain an equal number of layers and are interconnected via skip connections, facilitating the effective fusion of high-level and low-level features.

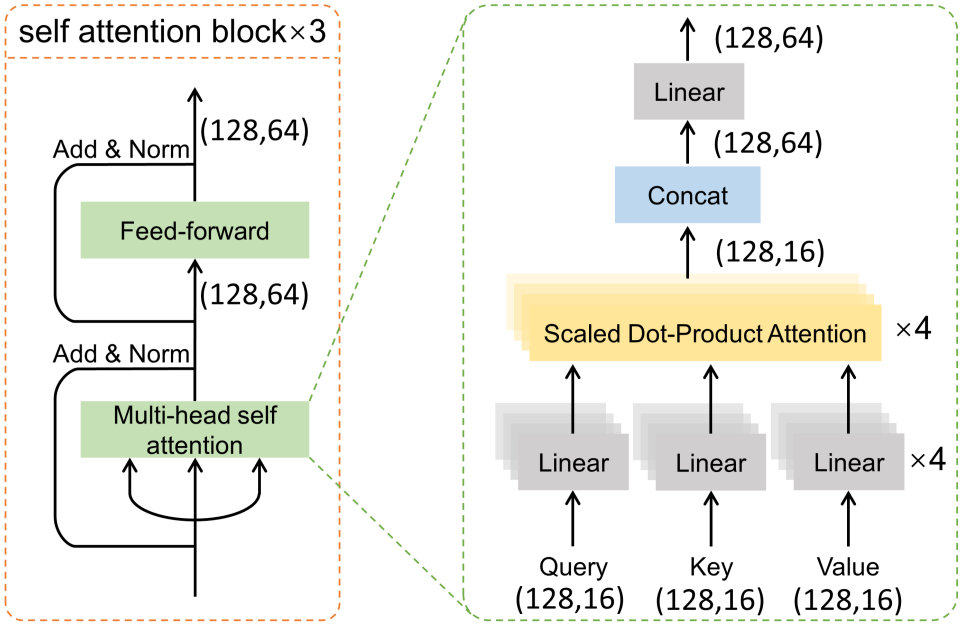


FIGURE 4.3: The bottleneck architecture.

Within the decoder, this integration is achieved through additive operations. The detection network, designed for the binary hypothesis testing task, is built upon two CNN layers and two dense layers. Additionally, the proposed network incorporates batch normalization layers after each CNN module, followed by rectified linear unit (ReLU) activation layers and dropout layers to improve training and generalization.

The bottleneck, serving as the bridge between the encoder and decoder, consists of two CNN layers and three self-attention blocks, as shown in Figure 4.3. Each self-attention block comprises a multi-head self-attention layer and a feed-forward layer, enveloped with skip connections and succeeded by layer normalization. The multi-head self-attention layer operates with 4 heads to capture complex dependencies and relationships within the input feature, while the feed-forward layer consists of an input and hidden layer channel sizes of 64 and 128, respectively. Subsequently, we will provide a detailed explanation of the multi-head self-attention procedure. The input feature is duplicated to create separate query, key, and value vectors, denoted as Q_i, K_i, and V_i, respectively, where $i = 1, \ldots, 4$ represents the head index. Each vector is then processed through a dense layer and scaled dot-product attention mechanism. The output of each head can be mathematically expressed as follows:

$$\text{head}_i = \text{Attention}_i\left(Q_i W_i^Q, K_i W_i^K, V_i W_i^V\right), \tag{4.7}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{16\times 16}$ denote the weight matrices of the first dense layer subsequent to the input. Let $\widetilde{Q_i}$, $\widetilde{K_i}$ and $\widetilde{V_i}$ represent the computations of $Q_i W_i^Q$, $K_i W_i^K$ and $V_i W_i^V$, respectively. Thus, Attention_i can be formulated as follows:

$$\text{Attention}_\text{i}\left(\widetilde{Q_i}, \widetilde{K_i}, \widetilde{V_i}\right) = \text{softmax}\left(\frac{\widetilde{Q_i}\widetilde{K_i}^T}{\sqrt{d_k}}\right)\widetilde{V_i}, \tag{4.8}$$

where $d_k = 16$ is the channel dimension of K_i. Hence, the output of multi-head self attention modules is expressed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_i\right) W^O, \tag{4.9}$$

where $W^O \in \mathbb{R}^{64\times 64}$ denotes the weight matrix of the dense layer that follows the Concat layer.

In the context of OFDM spectrum sensing, it is crucial to recognize that not all subcarriers contribute equally to the detection performance. Due to the inherent nature of OFDM, subcarriers experience diverse fading effects, resulting in varying levels of significance in their informational content, which necessitates the incorporation of a multi-head attention mechanism in our proposed JDSS network. By allocating disparate weights to individual subcarriers, the multi-head attention mechanism allows the model to dynamically focus on subcarriers that hold more salient information for spectrum sensing. Concurrently, it attenuates the influence of subcarriers that are less significant. This selective attention mechanism augments the performance of OFDM spectrum sensing, especially in the presence of fading introduced by the TDL-E channel.

4.3.2 Training Strategy

Generally, the training process f can be expressed as the composition of functions of denoising and detection, namely, $f = \mathcal{E} \circ \mathcal{G}$, where $\mathcal{E} : \mathbf{r} \to \hat{\boldsymbol{x}}$ aims to regress the input noisy signal $\mathbf{r}$ into a denoised signal $\hat{\boldsymbol{x}}$, and $\mathcal{G} : \hat{\boldsymbol{x}} \to \boldsymbol{d}$ is to determine whether the PU signal is present or absent.

4.3.2.1 Dataset Generation and Pre-Processing

The dataset for training and evaluating the proposed JDSS network is generated through the simulation of OFDM signals transmitted over a TDL-E channel, which provides an accurate representation of the wireless propagation environment in 5G communication scenarios.

For the wireless channel simulation, we employ the TDL-E model, which is characterized by 14 distinct paths, each defined by a time-varying complex gain and a fixed delay. The channel coefficients are generated in accordance with the distribution properties and parameters specified in Table 4.1. The maximum Doppler shift is configured at 148.18 Hz, corresponding to a mobile speed of 40 km/h, which is representative of typical urban mobility scenarios.

TABLE 4.1: TDL-E Channel Model Parameters

Parameters	**Value**
Carrier Frequency	4 GHz
Sampling Rate	32.75 MHz
Maximum Doppler Shift	148.18Hz
Mobile Speed	40 km/h
Delay Profile	TDL - E
Number of Paths	14

To provide a visual representation of the channel model, Figure 4.4 illustrates the magnitude of different paths for each frame, offering insight into the multipath characteristics of the simulated wireless environment.

The dataset is constructed by generating OFDM signals over the TDL-E channel within the SNR range of -18 dB to -5 dB, with an increment of 1 dB. For each distinct SNR level, the dataset comprises 10,000 OFDM training frames and 2,000 evaluation frames, each with a length of 1,024. To address the issue of class imbalance, an equal number of AWGN vectors are generated, corresponding to the H_0 hypothesis scenario with label 0. Besides, To facilitate the joint training of the denoising and detection sections, clean OFDM samples transmitted at the PU end are also collected. The constructed training dataset can be formulated as:

$$\{R, X, D\} = \{(\mathbf{r}_1, \mathbf{x}_1, d_1), \ldots, (\mathbf{r}_M, \mathbf{x}_M, d_M)\}, \tag{4.10}$$

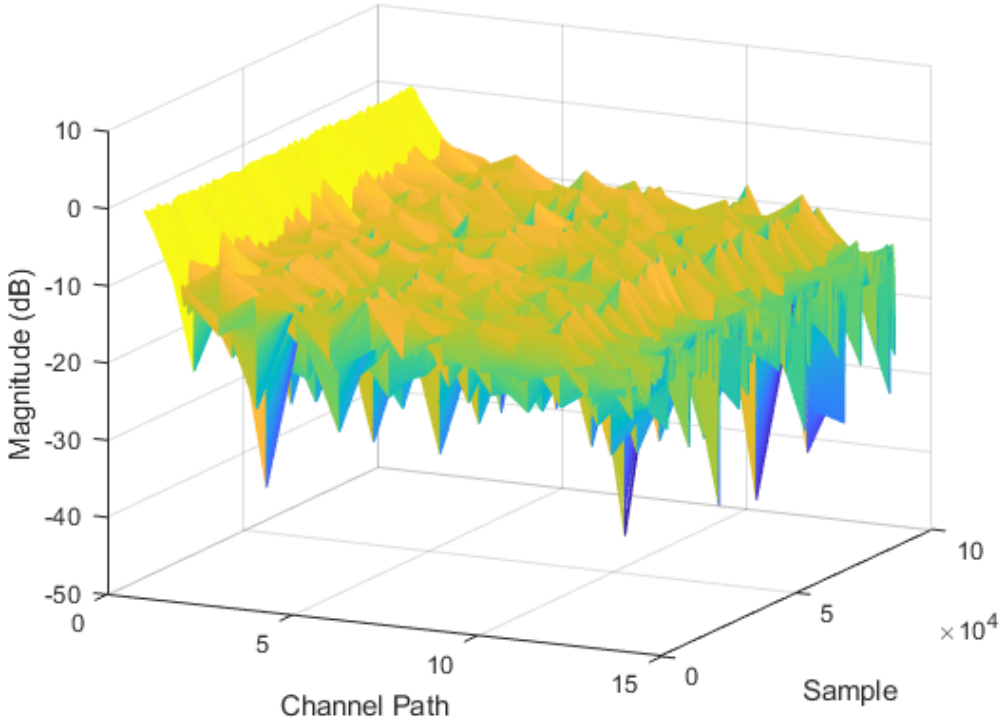


FIGURE 4.4: Visualization of TDL-E channel model regarding with the magnitude of different paths.

where $R = \{\mathbf{r}_m\}_{m=1}^{M}$ denotes the M received noisy OFDM signals or AWGN vectors, expressed by the label $d_m = 1$ or $d_m = 0$. Additionally, $X = \{\mathbf{x}_m\}_{m=1}^{M}$ represents the set of clean OFDM signals obtained directly from the PU transmitter without noise. Note that the corresponding clean samples $\mathbf{x}_m$ of AWGN vectors are designated as 0.

4.3.2.2 Loss Function

To facilitate joint training of the denoising and detection phases, we have devised a composite loss function. For the denoising segment, the loss function is comprised of two elements: the ℓ_1 loss relating to signal waveform and a multi-resolution STFT (Multi-STFT) loss computed between the clean and denoised signals. Note that the input to our network are OFDM signals in the frequency domain, which are derived from their time-domain representations using fast Fourier transform (FFT). Nevertheless, the computation of the denoising loss is performed in the time domain. We denote x as the clean signal and $\hat{x}$ as the denoised signal, which can be expressed as $\hat{\mathbf{x}} = \mathcal{E}(\mathbf{r}; \boldsymbol{\theta}_e)$, where $\boldsymbol{\theta}_e$ represents the weights of the denoising network.

Let $s(\mathbf{x};\boldsymbol{\nu}_i)$ and $s(\hat{\mathbf{x}};\boldsymbol{\nu}_i)$ be the STFT envelope of clean signal $\mathbf{x}$ and denoised signal $\hat{\mathbf{x}}$, with $\boldsymbol{\nu}_i$ representing the hyper-parameters of the i^{th} resolution STFT including the window size, the FFT bin, and the hop size. Subsequently, spectral convergence $\mathcal{L}_{sc}$ and magnitude loss $\mathcal{L}_{mag}$ are incorporated into the Multi-STFT loss, which is expressed as follows:

$$\begin{aligned}\text{Multi-STFT}(\mathbf{x},\hat{\mathbf{x}}) = \mathcal{L}_{sc} + \mathcal{L}_{mag} = \\ \sum_{i=1}^{I=3}\left(\frac{\|s\left(\mathbf{x};\boldsymbol{\nu}_i\right) - s\left(\hat{\mathbf{x}};\boldsymbol{\nu}_i\right)\|_F}{\|s\left(\mathbf{x};\boldsymbol{\nu}_i\right)\|_F} + \frac{1}{E}\left\|\log\frac{s\left(\mathbf{x};\boldsymbol{\nu}_i\right)}{s\left(\hat{\mathbf{x}};\boldsymbol{\nu}_i\right)}\right\|_1\right),\end{aligned} \tag{4.11}$$

where the notations $\|\cdot\|_F$ and $\|\cdot\|_1$ represent the Frobenius and l_1 norms, respectively. E denotes the number of elements in the magnitude $s(\mathbf{x};\boldsymbol{\nu}_i)$. The division and logarithm operations are performed elementwise. The first term employs the Frobenius norm to emphasize the convergence of large spectral components, which is crucial in early training phases. The second term, utilizing logarithmic scaling and L1 norm, helps in accurately capturing small-amplitude components that become important in later stages of training. [115–117] Thus, the overall denoising loss function can be expressed as

$$\mathcal{L}_{\text{denoise}}(\boldsymbol{\theta}_e) = \text{Multi-STFT}(\mathbf{x},\hat{\mathbf{x}}) + \|\mathbf{x}-\hat{\mathbf{x}}\|_1. \tag{4.12}$$

The L1 norm is chosen for both the magnitude loss in Equation (4.11) and the time-domain loss in Equation (4.12) due to its robustness to outliers. It helps maintain stable training when dealing with occasional large signal deviations while preserving important signal characteristics in both time and frequency domains. Additionally, given that our ultimate objective involves binary hypothesis classification, it is required to incorporate a cross-entropy loss component, as given by

$$\begin{aligned}\mathcal{L}_{\text{ce}}(\boldsymbol{\theta}_e,\boldsymbol{\theta}_g) = &\frac{1}{B}\sum_{m=1}^{B} -\left[d_m\cdot\log\left(f_{H_1|\boldsymbol{\theta}}\left(\mathbf{r}_m\right)\right) + \right. \\ &\left.\left(1-d_m\right)\cdot\log\left(1-f_{H_1|\boldsymbol{\theta}}\left(\mathbf{r}_m\right)\right)\right],\end{aligned} \tag{4.13}$$

where d_m represents the class label of the m^{th} sample $\mathbf{r}_m$. Therefore, the cumulative loss of the JDSS can be represented as follows:

$$\mathcal{L}(\boldsymbol{\theta}_e,\boldsymbol{\theta}_g) = \gamma_1\mathcal{L}_{\text{denoise}}(\boldsymbol{\theta}_e) + \gamma_2\mathcal{L}_{\text{ce}}(\boldsymbol{\theta}_e,\boldsymbol{\theta}_g), \tag{4.14}$$

where γ_1 and γ_2 are weighting factors of two loss components, and $\boldsymbol{\theta}_g$ denotes the weights of classification network $\mathcal{G}$. Hence, the objective is to identify the optimal weights $(\boldsymbol{\theta}_e^*, \boldsymbol{\theta}_g^*)$ of the JDSS network, which can be expressed by

$$\left(\boldsymbol{\theta}_e^*, \boldsymbol{\theta}_g^*\right) = \arg\min_{\boldsymbol{\theta}_e, \boldsymbol{\theta}_g} \mathcal{L}\left(\boldsymbol{\theta}_e, \boldsymbol{\theta}_g\right). \tag{4.15}$$

4.3.3 Online Prediction

For a more accurate assessment of detection performance, it is required to maintain a constant P_{fa}. As indicated in Equation (4.5), an optimal decision threshold η is required for the accurate detection of PU signals. To achieve this, we randomly select Z AWGN samples, $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_Z\}$, which are denoted as the hypothesis H_0. These samples are then sequentially arranged in the ascending order of $f_{H_1|\boldsymbol{\theta}}\left(\mathbf{r}_Z\right)$, resulting in $\{\widetilde{\mathbf{r}}_1, \widetilde{\mathbf{r}}_2, \ldots, \widetilde{\mathbf{r}}_Z\}$, which satisfy the following condition:

$$\forall 1 < i < j < Z, f_{H_1|\theta^*}\left(\widetilde{\mathbf{r}}_i\right) < f_{H_1|\theta^*}\left(\widetilde{\mathbf{r}}_j\right). \tag{4.16}$$

Hence, the reordered sequence corresponding to the Z output probabilities under the hypothesis H_1 is as follows:

$$\left\{f_{H_1|\boldsymbol{\theta}^*}\left(\widetilde{\mathbf{r}}_1\right), f_{H_1|\boldsymbol{\theta}^*}\left(\widetilde{\mathbf{r}}_2\right), \ldots, f_{H_1|\boldsymbol{\theta}^*}\left(\widetilde{\mathbf{r}}_Z\right)\right\}. \tag{4.17}$$

Lastly, the decision threshold η can be calculated as follows:

$$\eta = f_{H_1|\boldsymbol{\theta}^*}\left(\widetilde{\mathbf{r}}_{\text{floor}\left(Z*\left(1-P_{fa}\right)\right)}\right), \tag{4.18}$$

where the function floor $(\cdot)$ truncates a real number to the largest integer less than or equal to it. Consequently, the presence of PU signals can be determined using Equations (4.5) and (4.18).

In conclusion, the proposed spectrum sensing methodology that encompasses both training and testing procedures is concisely summarized in **Algorithm 2**.

Algorithm 2 Joint denoising and spectrum sensing (JDSS) scheme

Pre-processing: Reshape complex vectors of length 1024 into real and imaginary dimensions in a 2×1024 format.
Initialize: Training epochs $Iter_M$, batchsize B, weights of JDSS $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_g$, and a specific trade-off parameter γ_1, γ_2 in Equation (4.14).
1: **Repeat**
2: **Compute** the denoising loss $\mathcal{L}_{denoise}(\boldsymbol{\theta}_e)$, along with the classification loss $\mathcal{L}_{ce}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_h)$.
3: **Compute** $\mathcal{L}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_g)$ with a specific pair of γ_1 and γ_2;
4: **Update** the network's weights $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_h$ based on back propagation using gradient descent method *Adam*;
5: **Until** no improvement in *accuracy* or $epoch = Iter_M$;
6: **Return** the probability for each hypothesis, namely, $f_{H_0|\boldsymbol{\theta}}(\mathbf{r})$, $f_{H_1|\boldsymbol{\theta}}(\mathbf{r})$, along with the optimally trained parameters $\boldsymbol{\theta}_e^*$, $\boldsymbol{\theta}_h^*$ of the network.
Prediction: Evaluate samples under the SNR range of -18 dB to -5 dB using the decision threshold η, determined by Equation (4.18).

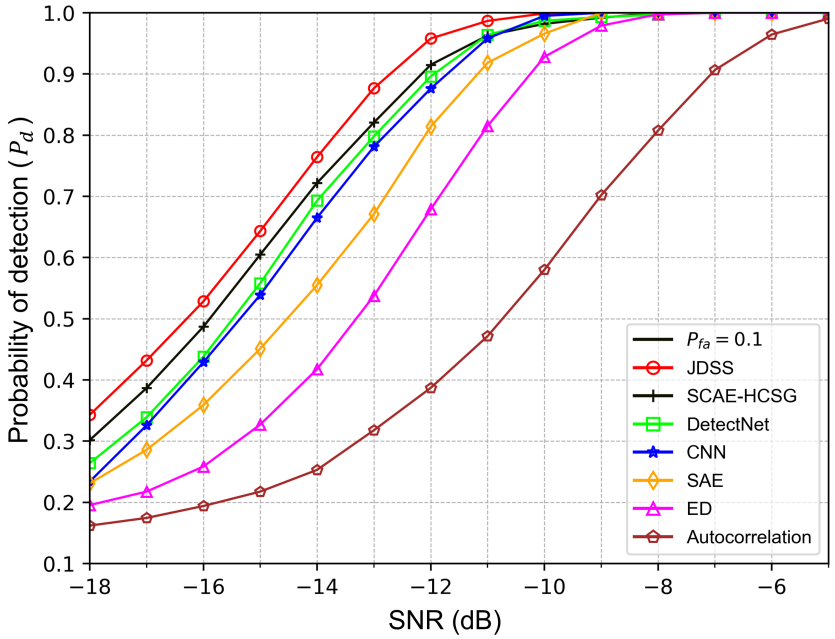
4.4 Simulation Results and Analysis

This section presents a comprehensive analysis of the proposed JDSS network. We begin with a performance evaluation of the proposed method compared to benchmark models. This is followed by a hyper-parameter analysis to assess the impact of trade-off factors on the network's performance. An ablation study is then conducted to evaluate the contribution of individual components of the JDSS architecture. Finally, we provide a complexity analysis of our proposed methodology alongside the benchmark models.

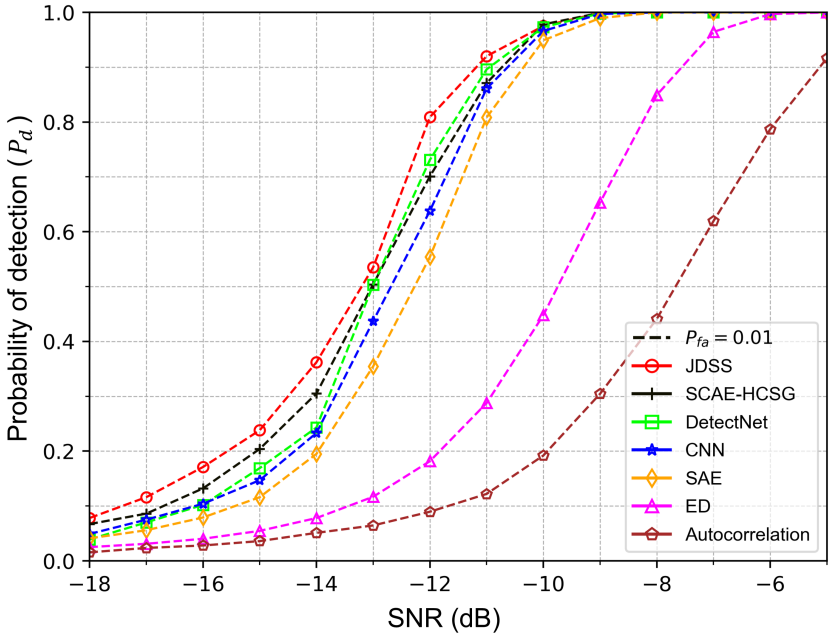
4.4.1 Performance Evaluation

In the experiments, we evaluate and compare the detection performance of JDSS to those of conventional spectrum sensing approaches, as well as deep learning-based techniques. These include the two-stage SCAE-H-CSG sensing scheme, DetectNet, CNN, SAE, ED and autocorrelation-based detection. To maintain a fair and unbiased assessment of each method's performance, the training datasets are kept identical across all methodologies. However, an exception was made for the autocorrelation-based detector, which necessitates the use of CP.

As illustrated in Figure 4.5, the probabilities of detection P_d of different spectrum sensing approaches are plotted against different SNR levels at P_{fa} of 0.1 and



(a) Detection performance at $P_{fa} = 0.1$.



(b) Detection performance at $P_{fa} = 0.01$.

FIGURE 4.5: Probabilities of detection of different approaches at (a) $P_{fa} = 0.1$; (b) $P_{fa} = 0.01$.

0.01, respectively. It is shown that our proposed JDSS consistently outperforms other techniques across different SNR levels. Particularly, at $P_{fa} = 0.1$, DetectNet, CNN, and SAE, which do not have a signal enhancement component, display inferior performance in comparison to JDSS and SCAE-H-CSG. Under the extremely low SNR conditions, the JDSS approach achieves nearly 35% in P_d, offering an

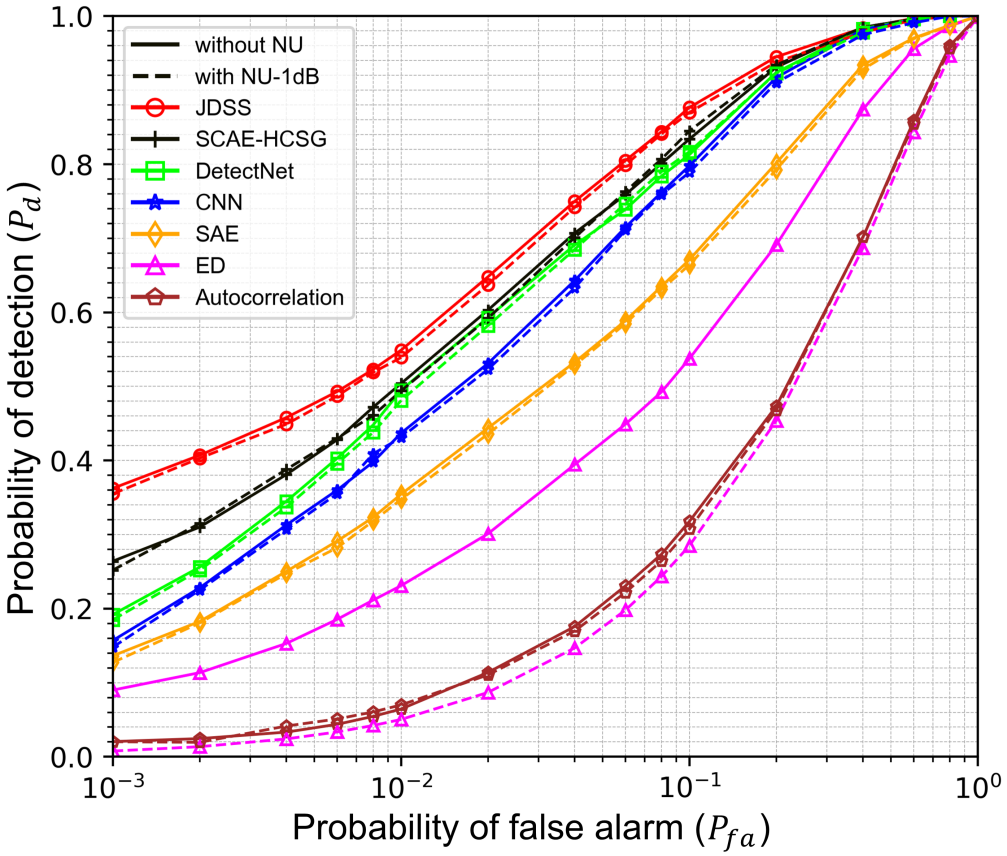


FIGURE 4.6: ROC curves for different approaches at SNR = -13dB: With and without NU.

appreciable improvement of 8% to 13% relative to DetectNet, CNN, and SAE. The integration of self-attention modules within the bottleneck effectively mitigates the fading effects induced by the TDL-E channel, demonstrating the strengths of our proposed method. In contrast to the mean-square error (MSE)-centric denoising loss function employed by SCAE-H-CSG, JDSS leverages a combination of STFT and l_1 waveform loss functions. This strategic selection provides approximately 5% improvement in P_d. Furthermore, when $P_{fa} = 0.01$, it can also be observed that the proposed JDSS performs better than the other approaches. The ED approach, characterized by its low computational complexity and ease of implementation, yields a P_d that is inferior to those DL-based spectrum sensing techniques. On the other hand, its performance is observed to surpass that of the autocorrelation-based detection method, which is inherently dependent on the CP length. Noting that with the prior knowledge of noise power σ_w^2, ED can outperform the detectors that are based on second-order statistics. For our simulations, a configuration of $N_c = N_{sc}\zeta/8$ is employed for the autocorrelation-based detection.

In Figure 4.6, we show the ROC curves of various methods at SNR=-13dB. The ROC curve provides an insightful visualization of the binary classification model's performance. This curve is constructed based on two key metrics, namely, P_d and P_{fa}. By controlling P_{fa} on a logarithmic scale along the x-axis, we can observe the corresponding variations in P_d along the y-axis. Notably, it reveals an evident superiority in the performance of the proposed JDSS, particularly under conditions characterized by lower P_{fa}. Specifically, when P_{fa} is set to 0.001, the JDSS achieves a P_d of around 35%. In contrast, other DL-based schemes yield P_d values ranging from 13% to 24%. Furthermore, we evaluate the robustness of these methodologies under conditions of NU. With an NU of 1dB (referred to as "NU-1dB" in Figure 4.6), the estimated noise power $\hat{\sigma}_w^2$ follows a uniform distribution across $\left[\frac{1}{\varepsilon}\sigma_w^2, \varepsilon\sigma_w^2\right]$, with ε representing the NU factor. Specifically, in the case of 1dB NU, the NU factor ε is calculated as $\varepsilon = 10^{\frac{1\text{dB}}{10}} \approx 1.26$. The ED-based approach is observed to be highly sensitive to NU, primarily due to its dependence on noise power estimation for blind detection, leading to worse performance. In contrast, while the autocorrelation-based detection scheme might underperform compared to ED in NU-free scenarios, it demonstrates robustness against NU, thanks to its reliance on second-order statistics. DL-based techniques, such as our proposed JDSS, showcase the resilience to SNR variations as they operate under a completely blind sensing scenario without any prior knowledge. These results demonstrate the efficiency and reliability of JDSS across different P_{fa} levels and in the presence of NU.

4.4.2 Hyper-Parameter Analysis

Figure 4.7 illustrates the impact of varying the ratio γ_1/γ_2 on the model's performance, represented by the mean P_d (%) on the y-axis. The x-axis shows the different values of γ_1/γ_2 explored in the analysis, ranging from 0 to 10. The hyper-parameter analysis reveals that the model's performance exhibits a non-monotonic behavior with respect to the ratio γ_1/γ_2. As the ratio increases from 0 to approximately 0.25, the model's performance improves, reaching a peak at $\gamma_1/\gamma_2 \approx 0.25$. This suggests that a balanced contribution from both weighting factors is beneficial for the model's performance. However, as the ratio continues to increase beyond 0.25, the model's performance starts to decline. This indicates that placing a disproportionately high emphasis on γ_1 relative to γ_2 can have a detrimental effect on

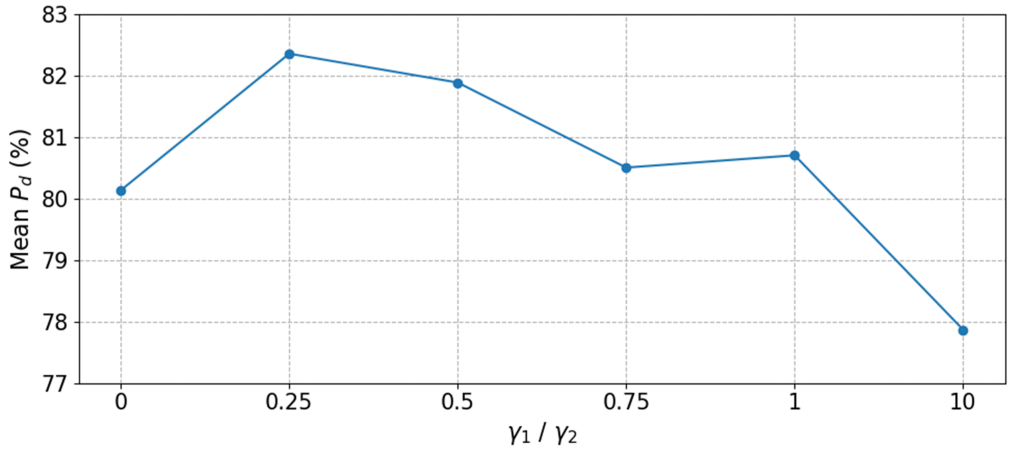


FIGURE 4.7: Hyper-parameter analysis of the JDSS network performance with respect to the ratio of weighting factors γ_1/γ_2, where γ_2 is fixed at 1.

the model's performance. The model's performance appears to be more sensitive to changes in the ratio γ_1/γ_2 when it is greater than 1. This is evident from the steeper slope of the curve in the region where $\gamma_1/\gamma_2 > 1$ compared to the region where $\gamma_1/\gamma_2 < 1$. This observation implies that the model's performance is more robust to variations in the weighting factors when γ_1 is smaller than γ_2.

Based on the hyper-parameter analysis, it can be concluded that the optimal ratio of weighting factors for the model is approximately $\gamma_1/\gamma_2 \approx 0.25$. This ratio strikes a balance between the contributions of γ_1 and γ_2, enabling the model to achieve its best performance. Deviating significantly from this optimal ratio, especially when γ_1 becomes much larger than γ_2, can lead to a degradation in the model's performance.

4.4.3 Ablation Study

To gain a deeper understanding of the individual component's contribution to the overall performance of the proposed JDSS network, we conduct an ablation study. This study involves evaluating the performance of different variants of the JDSS network by removing or modifying specific components and comparing their results with the proposed JDSS model. Table 4.2 presents the results of the ablation experiments under selected SNR levels. We consider four different model configurations: (1) U-Net with concatenation in the decoder part, (2) U-Net with SA module in the bottleneck and concatenation in the decoder part, (3) U-Net with element-wise

addition in the decoder part, and (4) the proposed JDSS model, which incorporates both SA in the bottleneck and element-wise addition in the decoder part.

TABLE 4.2: Ablation experiments on JDSS model under selected SNR level.

Component	**-18dB**	**-16dB**	**-14dB**	**-12dB**
U-Net (Concat)	27.37	41.03	65.23	91.33
U-Net + SA (Concat)	31.97	47.67	75	94.13
U-Net (Add)	33.63	48.87	74.4	94.93
JDSS	**34.29**	**52.84**	**76.43**	**95.8**

The choice of baseline is motivated by the widespread use of concatenation in the original U-Net architecture, where feature maps from the encoder are concatenated with the corresponding decoder feature maps. From the results, we observe that the integration of the self-attention module in the bottleneck (U-Net + SA (Concat)) leads to a noticeable improvement in performance when compared to the baseline, particularly at lower SNR levels. For instance, at an SNR of -18 dB, the U-Net + SA (Concat) model achieves a 4.6% improvement over the baseline U-Net (Concat) model, increasing from 27.37% to 31.97%. Similarly, at -16 dB, the performance improves from 41.03% to 47.67%. This highlights the importance of the self-attention mechanism in capturing the dependencies between different subcarriers and enhancing the network's ability to extract meaningful features for spectrum sensing.

Furthermore, replacing the concatenation operation with element-wise addition in the decoder part (U-Net (Add)) results in a further performance boost. The element-wise addition operation offers a computationally efficient alternative to concatenation by reducing the need to process additional feature maps. This operation enables the fusion of features from different stages of the network, allowing the decoder to benefit from both low-level and high-level feature representations. The comparable performance between the "add" and "concatenation" operations suggests that the fused features resulting from the element-wise addition capture sufficient discriminative information for effective spectrum sensing.

The proposed JDSS model, which incorporates both the self-attention module in the bottleneck and element-wise addition in the decoder part, achieves the best

performance across all SNR levels. At an SNR of -18 dB, the JDSS model achieves a 34.29% accuracy, surpassing the U-Net (Add) model by 0.66%. The performance of JDSS model remains superior as the SNR increases, achieving P_d of 52.84%, 76.43%, and 95.8% at SNR levels of -16 dB, -14 dB, and -12 dB, respectively. These results demonstrate the synergistic effect of combining these two components, leading to enhanced spectrum sensing accuracy in low SNR environments.

In summary, the ablation study highlights the individual contributions of the self-attention module and the element-wise addition operation in the JDSS network. The self-attention mechanism enhances the network's ability to capture dependencies between subcarriers, while the element-wise addition operation provides a computationally efficient way to fuse features from different stages of the network. The combination of these components in the proposed JDSS model results in superior performance for joint denoising and spectrum sensing, particularly in challenging low SNR scenarios. These findings underscore the effectiveness of the proposed architectural modifications in addressing the complexities of spectrum sensing in modern wireless communication systems.

4.4.4 Complexity Analysis

The computational complexity analysis for each DL-based spectrum sensing method is provided in Table 4.3, where l is the index of the network layer. For 1D CNN layers, u_l, k_l and m_l represent the number of channels, kernel size, and output length of the l^{th} layer, respectively. Let c_l be the input length of the deconvolutional layer, which shows similar complexity as that of 1D-CNN layers. Besides, d_l represents the number of neurons of the l^{th} dense layer. e_l and h_l denote the embedding size and the number of hidden units of the l^{th} GRU or LSTM layer. Furthermore, $D_{1\sim5}$ represent the number of 1D-CNN layers, deconvolutional layers, LSTM layers, GRU layers and dense layers, respectively. More intuitively, FLOPs is employed to measure the complexity. With approximately 35.979 million FLOPs, JDSS notably reduces complexity when contrasted with the SCAE-H-CSG method, which requires roughly 148.81 million FLOPs. Concurrently, JDSS exhibits better in P_d with comparable FLOPs to both DetectNet and CNN, which require nearly 36.58 million and 38.4 million FLOPs, respectively. SAE achieves the minimal

TABLE 4.3: Computational complexity formulas and FLOPs of the proposed scheme and baselines

Model	**Formula for Complexity**	FLOPs (in millions)
JDSS	$O\left(\sum_{l=1}^{D_1} u_{l-1}k_l m_l u_l + \sum_{l=1}^{D_2} u_{l-1} k_l c_l u_l + \sum_{l=1}^{D_5} d_{l-1}d_l\right)$	35.979
SCAE-H-CSG	$O\left(\sum_{l=1}^{D_1} u_{l-1}k_l m_l u_l + \sum_{l-1}^{D_4} 3\cdot(h_l + e_l)\, h_l + \sum_{l=1}^{D_5} d_{l-1}d_l\right)$	148.81
DetectNet	$O\left(\sum_{l=1}^{D_1} u_{l-1}k_l m_l u_l + \sum_{l-1}^{D_3} 4\cdot(h_l + e_l)\, h_l + \sum_{l=1}^{D_5} d_{l-1}d_l\right)$	36.58
CNN	$O\left(\sum_{l=1}^{D_1} u_{l-1}k_l m_l u_l + \sum_{l=1}^{D_5} d_{l-1}d_l\right)$	38.4
SAE	$O\left(\sum_{l=1}^{D_5} d_{l-1}d_l\right)$	2.593

2.593 million FLOPs as it utilizes only a few dense layers. However, this streamlined architecture also results in relatively inferior performance compared to other DL-based methods.

4.5 Conclusion

In this chapter, we have presented a novel joint denoising and spectrum sensing scheme that effectively determines the state of PUs by leveraging signal enhancement techniques. The proposed method combines the denoising and sensing tasks into a single end-to-end deep learning framework, enabling the network to learn efficient and effective representations tailored specifically for spectrum sensing. By incorporating a multi-resolution STFT loss with an l_1 waveform loss, the JDSS scheme outperforms conventional MSE regression denoising models, demonstrating significant improvements in both denoising and sensing performance. Moreover,

the joint optimization of denoising and spectrum sensing not only improves overall performance but also reduces computational complexity compared to the two-stage SCAE-H-CSG approach. Furthermore, the integration of multi-head self-attention modules in the bottleneck allows the network to focus on subcarriers that are minimally affected by fading, effectively addressing the challenges posed by the TDL-E channel model. This selective attention mechanism enhances the robustness of the JDSS scheme in the presence of diverse fading effects, ensuring reliable spectrum sensing performance in realistic 5G communication scenarios.

Our extensive experiments with the JDSS framework reveals several practical insights. The empirical studies show that the optimal ratio of weighting factors between denoising and detection objectives significantly impacts the model's performance. This finding suggests that while signal enhancement is important, excessive emphasis on denoising can actually degrade overall sensing performance. The ablation studies demonstrates that the incorporation of self-attention in the bottleneck notably improves the network's ability to capture dependencies between subcarriers. Additionally, element-wise addition in the decoder part outperforms concatenation operations, offering both improved performance and computational efficiency. These insights provide valuable guidance for future designs of joint optimization frameworks in spectrum sensing applications.

Despite the significant advancements demonstrated by the JDSS scheme, it is important to acknowledge that the performance of the proposed method relies on the availability of a sufficient amount of labeled data for training. In practical scenarios, collecting a large volume of labeled data can be challenging and time-consuming, particularly when considering the diverse range of channel conditions and signal characteristics encountered in real-world environments.

To address this limitation and further enhance the practicality of the JDSS scheme, we will focus on developing domain adaptation techniques that enable the network to generalize well to new environments without requiring extensive retraining. In the next chapter, we will explore a domain-generalized network that aims to extract channel-invariant features, allowing the model to maintain its performance in new environments without the need for retraining using labeled data from the target domain.

By leveraging domain generalization techniques, we aim to develop a spectrum sensing framework that can effectively adapt to varying channel conditions and signal characteristics, reducing the reliance on labeled data. This research direction holds significant promise for enhancing the efficiency and flexibility of spectrum sensing in dynamic wireless environments, ultimately contributing to the realization of intelligent and adaptive cognitive radio systems.

Chapter 5

Channel Invariant Spectrum Sensing over Unknown Environments

5.1 Introduction

In the previous chapter, we introduced the joint denoising and spectrum sensing scheme, which demonstrates significant advancements in spectrum sensing performance. However, the JDSS scheme's reliance on a substantial amount of labeled data for training poses a challenge in practical scenarios, where collecting such data across diverse channel conditions can be time-consuming and resource-intensive.

To address this limitation and enhance the practicality of our spectrum sensing framework, this chapter focuses on developing a domain-generalized network that can extract channel-invariant features. The primary objective is to enable the model to maintain its performance in new environments without the need for extensive retraining using labeled data from the target domain.

The electromagnetic wave propagation environment is highly diverse, leading to significant variations in the received signal distribution. While generating training data for a specific channel condition is straightforward, as demonstrated in Chapter 4 with the JDSS framework, a sensing model trained for one environment often performs poorly when deployed in different channel conditions. This creates

a practical challenge: the need for continuous model retraining to maintain performance across different deployment environments. Although one could potentially generate and collect training data for each new channel condition, this approach becomes impractical when considering the vast diversity of real-world wireless environments and the computational resources required for repeated retraining. Several studies have attempted to tackle this issue. For instance, in [48], cross-correlation property of the cyclic prefix of OFDM signals is exploited and a robust sensing network was proposed via a matching network. Besides, motivated by adversarial learning strategies, a deep transfer cooperative sensing (DTCS) approach was proposed [69]. However, these methods are often limited to specific signal types or require additional information about the training and target domains, limiting their applicability in real-world scenarios.

To overcome these obstacles, we propose an environment-robust spectrum sensing network (ER-SNet). The main contributions of this work are as follows:

- ER-SNet is a non-cooperative spectrum sensing scheme capable of detecting the presence of unknown PU signals in new propagation environments.
- The network operates on complex received signals, allowing it to extract underlying features more effectively than manual extraction. The core idea is to extract channel-insensitive features that are discriminative for PU state but robust to channel variations to some extent.
- Extensive simulations demonstrate the effectiveness of ER-SNet, particularly in significantly reducing the sensing performance gap between source channels and unknown target channels. The proposed approach consistently outperforms other baseline schemes, including ED [110], DTCS [69], DetectNet [42], JDSS [63], and transfer learning-based ResNet (TF-ResNet) [39].

By leveraging domain generalization techniques, ER-SNet aims to develop a spectrum sensing framework that can effectively shrink the performance gap between source channel conditions and unknown target channel conditions. This approach significantly reduces the reliance on labeled data and facilitates the deployment of spectrum sensing schemes in practical cognitive radio networks.

The remainder of this chapter is organized as follows. Section 5.2 presents the system model. Section 5.3 elucidates the architecture and methodology of the

proposed ER-SNet. Section 5.4 encompasses a detailed exposition of the dataset configuration and computational complexity analysis, followed by a comprehensive evaluation of the model's performance and a hyper-parameter analysis. Finally, Section 5.5 summarizes the key contributions of this work.

5.2 System Model

In this work, we consider a non-cooperative spectrum sensing scenario where only one antenna is deployed to receive the signal samples. A general spectrum sensing system can be formulated as a binary hypothesis test problem: H_0 and H_1 represent the absence or presence of PU signals in the sensing frequency band, respectively. Let $\mathbf{r}_n \in \mathbb{C}^{1\times L}$ denote the n^{th} complex observed vector by SU in one sensing period, where L denotes the sampling length. The binary hypothesis system can be formulated as:

$$\begin{cases} H_0 : \mathbf{r}_n = \mathbf{w}_n, \\ H_1 : \mathbf{r}_n = \mathbf{x}_n + \mathbf{w}_n, \end{cases} \tag{5.1}$$

where $\mathbf{w}_n \in \mathbb{C}^{1\times L}$ represents the i.i.d. additive noise following CSCG distribution with zero mean and variance σ_w^2, namely, $\mathbf{w}_n \sim CN\left(0, \sigma_w^2\right)$, $\mathbf{x}_n \in \mathbb{C}^{1\times L}$ is the n^{th} modulated signal transmitted by the PU through the fading channel. The solution of the above problem can be expressed as:

$$f_{\boldsymbol{\theta}|H_1}(\mathbf{r}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{5.2}$$

where $\mathbf{r} \in \mathbb{C}^{1\times L}$ is the received I/Q vectors and $f_{\boldsymbol{\theta}|H_1}(\,\cdot\,)$ denotes the output probability of hypothesis H_1 with trained parameters $\boldsymbol{\theta}$ after the transformation formula $f(\,\cdot\,)$ of the proposed ER-SNet. We can determine the state of PU signals by comparing $f_{\boldsymbol{\theta}|H_1}(\mathbf{r})$ with the detection threshold η. Typically, two performance metrics are used to evaluate the algorithm, namely, the probability of detection P_d, and the probability of false alarm P_{fa}. For these two probabilities, we have $P_d = \mathrm{P_r}\left\{f_{\boldsymbol{\theta}|H_1}(\mathbf{r}) > \eta \mid H_1\right\}$, or true positive, and $P_{fa} = \mathrm{P_r}\left\{f_{\boldsymbol{\theta}|H_1}(\mathbf{r}) > \eta \mid H_0\right\}$, or false positive, where $\mathrm{P_r}\{\,\cdot\,\}$ denotes the probability of the event.

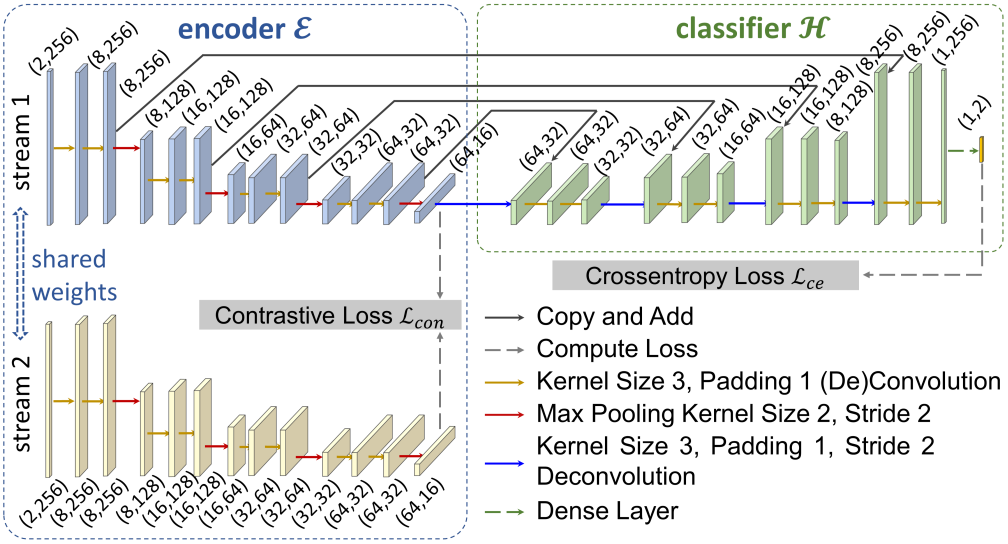


FIGURE 5.1: The proposed ER-SNet network architecture.

5.3 Environment-Robust Spectrum Sensing Scheme

In this section, we will first present the structure of the proposed ER-SNet. Then, we will introduce the training strategy and procedures of the network. Lastly, online deployment will be presented for the detection of PU signals.

5.3.1 Network Architecture of ER-SNet

As depicted in Figure 5.1, the proposed network contains two parts, namely, the encoder $\mathcal{E}$ and the classifier $\mathcal{H}$, and each module's output is labeled as (*Channel*, *Length*). Both streams of the neural network (NN) $\mathcal{E}$ encode the different input raw signals into high-dimensional features $\mathbf{z} = \mathcal{E}_{\boldsymbol{\theta}_e}(\mathbf{r})$ separately, where $\boldsymbol{\theta}_e$ denotes the training parameters of all the layers of NN $\mathcal{E}$. Since the high-dimensional features obtained by most of the existing DL-based spectrum sensing networks are highly sensitive to channel variations, it is difficult to infer the spectrum status precisely when the testing channel condition space Ω_2 is different from the training space Ω_1. While these spaces may share certain general features or statistical patterns, the diversity in channel conditions often leads to domain shifts that pose challenges for model generalization. For the purpose of this study, we consider $\Omega_2 \cap \Omega_1 = \emptyset$ to represent distinct domains, which allows for a rigorous evaluation of the proposed method's ability to generalize across unseen channel conditions. Adopting the contrastive loss function, a domain augmentation is conducted, which reduces the

channel-dependency of the encoding feature $\mathbf{z}$. In addition, the dependency of $\mathbf{z}$ is partly related to the size of training space, i.e., $|\Omega_1|$. With the increased number of source domains, the detection performance could be enhanced accordingly.

5.3.2 Training and Prediction of ER-SNet

Generally, the predicting process f of the proposed ER-SNet can be modelled by the composition of two functions, i.e., $f = \mathcal{E} \circ \mathcal{H}$, where $\mathcal{E} : \mathbf{r} \to \mathbf{z}$ aims to map the input I/Q vectors $\mathbf{r}$ into a high-dimensional embedding feature $\mathbf{z}$, and $\mathcal{H} : \mathbf{z} \to \mathbf{p}$ is to produce the classification probabilities of each hypothesis of the encoding features, i.e., $[p^0, p^1]^T$. Given the constraints of the available source datasets, the purpose of the network is to find out the best approximation of functions $\mathcal{E}$ and $\mathcal{H}$ with the designed loss.

5.3.2.1 Training Dataset Generation

To achieve channel invariant spectrum sensing, we collect the labelled samples from different channel conditions. The dataset $\mathcal{D}_s$ can be expressed as follows:

$$\{R, R', \mathcal{D}, \mathcal{B}\} = \{(\mathbf{r}_1, \mathbf{r}'_1, d_1, b_1), \ldots, (\mathbf{r}_K, \mathbf{r}'_K, d_K, b_K)\}, \tag{5.3}$$

where $R = \{\mathbf{r}_k\}_{k=1}^K$ denotes the K received samples with random class labels (0 or 1). Additionally, $R' = \{\mathbf{r}'_k\}_{k=1}^K$ shuffles and randomizes the orders of samples in R, where $\mathbf{r}_k$ and $\mathbf{r}'_k$ are sampled under $|\Omega_1|$ distinct channel conditions. $\mathcal{D} = \{d_k\}_{k=1}^K$ denotes whether the class labels of $\mathbf{r}_k$ and $\mathbf{r}'_k$ are the same ($d_k = 0$) or not ($d_k = 1$). $\mathcal{B} = \{b_k\}_{k=1}^K$ represents the true class labels of $\mathbf{r}_k$, where $b_k = 0$ indicates the absence of a primary user signal and $b_k = 1$ indicates its presence. During the process of training, one-hot encoding will be conducted for label d_k and b_k.

5.3.2.2 Training Process

The dataset consists of paired signal samples $\mathbf{r}_k$ and $\mathbf{r}'_k$, collected under different channel conditions. These paired inputs are fed into the two parallel streams of the encoder $\mathcal{E}$, as depicted in Figure 5.1. Both streams share identical network

weights, enabling the encoder to process these inputs symmetrically while ensuring consistency across the two streams.

The training of ER-SNet involves optimizing two key components: the encoder $\mathcal{E}$ and the classifier $\mathcal{H}$. The encoder $\mathcal{E}$ is designed to maximize the inter-class differences while minimizing the intra-class Euclidean distances, thereby mapping received signals from diverse channel environments to nearby area in the embedding feature space. Concurrently, the classifier $\mathcal{H}$ aims to accurately determine the spectrum status by minimizing the cross-entropy loss function.

The encoder $\mathcal{E}$ is structured as a Siamese network, comprising two parallel streams with identical CNN architectures and shared weights. The contrastive loss of the final encoding CNN layer is computed as follows:

$$\begin{aligned}\mathcal{L}_{\text{con}}(\boldsymbol{\theta_e}) =& \frac{1}{B}\sum_{k=1}^{B}\Big[(1-d_k)\left\|\mathcal{E}_{\boldsymbol{\theta_e}}\left(\mathbf{r}_k\right)-\mathcal{E}_{\boldsymbol{\theta_e}}\left(\mathbf{r'}_k\right)\right\|_F^2+ \\ & d_k(\max\left(0, m-\left\|\mathcal{E}_{\boldsymbol{\theta_e}}\left(\mathbf{r}_k\right)-\mathcal{E}_{\boldsymbol{\theta_e}}\left(\mathbf{r'}_k\right)\right\|_F\right)^2)\Big],\end{aligned} \tag{5.4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathcal{E}_{\boldsymbol{\theta_e}}(\cdot)$ is the function that maps the k^{th} input $\mathbf{r}_k$ into less channel-dependent feature space. In addition, B stands for the batch size, and m denotes the margin value that specifies the separability of the embedding features.

The determination of the margin value m is crucial for effective training. We employ a statistical method to establish an appropriate margin. This approach involves calculating the average distances of both positive and negative sample pairs within a batch, then setting the margin as:

$$m = \mu + J\sigma, \tag{5.5}$$

where μ is the mean distance, σ is the standard deviation of the distances, and J is a scaling factor. The value of J determines the proportion of distance distribution covered by the margin. Through extensive simulations, we empirically determined that $J = 2$ provides optimal performance. The chosen margin value ensures that the encoder learns to map samples from the same class closer together in the embedding space while maintaining sufficient separation between different

classes, thereby enhancing the network's ability to generalize across various channel conditions.

The classifier $\mathcal{H}$ is constituted with transposed CNN layers and a dense layer. Represented by $\mathcal{H}_{\theta_\mathbf{h}}$, where θ_h denotes the parameters of layers in the classifier $\mathcal{H}$, the output of the network can be expressed as

$$\begin{aligned} \mathcal{H}_{\boldsymbol{\theta_h}}\left(\mathcal{E}_{\boldsymbol{\theta_e}}\left(\mathbf{r}_k\right)\right) &= \left[\begin{array}{c} \mathcal{H}_{\boldsymbol{\theta_h}}\left(\mathcal{E}_{\boldsymbol{\theta_e}} \mid H_0\left(\mathbf{r}_k\right)\right) \\ \mathcal{H}_{\boldsymbol{\theta_h}}\left(\mathcal{E}_{\boldsymbol{\theta_e}} \mid H_1\left(\mathbf{r}_k\right)\right) \end{array}\right] \\ &= \left[\begin{array}{c} f_{\boldsymbol{\theta}|H_0}(\mathbf{r}_k) \\ f_{\boldsymbol{\theta}|H_1}(\mathbf{r}_k) \end{array}\right], \end{aligned} \tag{5.6}$$

with

$$f_{\boldsymbol{\theta}|H_0}(\mathbf{r}_k) + f_{\boldsymbol{\theta}|H_1}(\mathbf{r}_k) = 1, \tag{5.7}$$

where $f_{\boldsymbol{\theta}|H_i}(\mathbf{r}_k)(i = 0, 1)$ is the output probability of each hypothesis with hyper-parameters $\boldsymbol{\theta}$ of the whole network. We adopt the cross-entropy loss function for classification according to the maximum likelihood criterion, as given by

$$\begin{aligned} \mathcal{L}_{\text{cross}}(\boldsymbol{\theta_e}, \boldsymbol{\theta_h}) = & \frac{1}{B} \sum_{k=1}^{B} - \left[b_k \cdot \log\left(f_{\boldsymbol{\theta}|H_1}\left(\mathbf{r}_k\right)\right) + \right. \\ & \left. (1 - b_k) \cdot \log\left(1 - f_{\boldsymbol{\theta}|H_1}\left(\mathbf{r}_k\right)\right)\right], \end{aligned} \tag{5.8}$$

where b_k represents the class label of the k^{th} sample $\mathbf{r}_k$. To balance the portion of the contrastive loss and classification loss, equations (5.4) and (5.8) are normalized and weighted by λ_1 and λ_2, respectively. Hence, the total loss of the ER-SNet can be expressed as:

$$\mathcal{L}\left(\boldsymbol{\theta_e}, \boldsymbol{\theta_h}\right) = \lambda_1 \mathcal{L}_{\text{con}}(\boldsymbol{\theta_e}) + \lambda_2 \mathcal{L}_{\text{cross}}(\boldsymbol{\theta_e}, \boldsymbol{\theta_h}). \tag{5.9}$$

The optimal weights $(\boldsymbol{\theta_e}^*, \boldsymbol{\theta_h}^*)$ of ER-SNet can be obtained by

$$\left(\boldsymbol{\theta_e}^*, \boldsymbol{\theta_h}^*\right) = \underset{\boldsymbol{\theta_e}, \boldsymbol{\theta_h}}{\arg\min} \mathcal{L}\left(\boldsymbol{\theta_e}, \boldsymbol{\theta_h}\right). \tag{5.10}$$

5.3.2.3 Online prediction

With the well-trained ER-SNet, the prediction dataset $\mathcal{D}_t$ will be established from the testing channel space Ω_2. To better compare the detection performance, we need to maintain a constant probability of false alarm P_{fa}. As shown in equation (5.2), we shall derive a decision threshold η to identify the presence or absence of PU signals. V random samples $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_V\}$ with hypothesis H_0 are picked and rearranged in an ascending order of $f_{\boldsymbol{\theta}|H_1}(\mathbf{r}_v)$, as given by $\{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \ldots, \hat{\mathbf{r}}_V\}$, which satisfy

$$\forall 1 < i < j < V, f_{\boldsymbol{\theta}^*|H_1}(\hat{\mathbf{r}}_i) < f_{\boldsymbol{\theta}^*|H_1}(\hat{\mathbf{r}}_j). \tag{5.11}$$

Thus, the reordered sequence with V corresponding classification probabilities of hypothesis H_1 is given by:

$$\left\{ f_{\boldsymbol{\theta}^*|H_1}(\hat{\mathbf{r}}_1), f_{\boldsymbol{\theta}^*|H_1}(\hat{\mathbf{r}}_2), \ldots, f_{\boldsymbol{\theta}^*|H_1}(\hat{\mathbf{r}}_V) \right\}. \tag{5.12}$$

Lastly, the decision threshold can be obtained by

$$\eta = f_{\boldsymbol{\theta}^*|H_1}\left(\hat{\mathbf{r}}_{\text{floor}\left(V*\left(1-P_{fa}\right)\right)}\right), \tag{5.13}$$

where the function floor $(\cdot)$ rounds down to the nearest integer. Thus, we can regulate P_{fa} to be our desired value, and the PU state will be determined by equations (5.2) and (5.13).

Hence, the proposed scheme with training and testing procedures is summarized in **Algorithm 3**.

5.4 Simulation Results and Analysis

In this section, we will present the parameters of the dataset in detail, and analyze the complexity of our proposed scheme. Following that, simulation results and hyper-parameter analysis will be presented.

Algorithm 3 Environment-robust spectrum sensing scheme (ER-SNet)

Pre-processing: Collect training samples and constitute the dataset $\mathcal{D}_s = \{R, R', \mathcal{D}, \mathcal{B}\}$
Initialize: Set $\boldsymbol{\theta_e}$, $\boldsymbol{\theta_h}$ with random initial weights
1: **Repeat**
2: **Compute** the total loss $\mathcal{L}(\boldsymbol{\theta_e}, \boldsymbol{\theta_h})$ with Equation (5.9);
3: **Compute** the backward gradients;
4: **Update** the network's weights $\boldsymbol{\theta_e}$, $\boldsymbol{\theta_h}$ by employing gradient descent method *Adam*;
5: **Until** *accuracy* with no improvement;
6: **Return** the probability of each hypothesis, i.e., $f_{\boldsymbol{\theta}|H_0}(\mathbf{r})$, $f_{\boldsymbol{\theta}|H_1}(\mathbf{r})$, and the well-trained optimal parameters $\boldsymbol{\theta_e}^*$, $\boldsymbol{\theta_h}^*$.
Prediction: Evaluation of samples under untrained channel conditions utilizing the decision threshold η with the Equation (5.13).

5.4.1 Dataset Setting

In this section, we establish the training dataset by generating QPSK signals over diverse multipath fading channels. The signals are configured with a roll-off factor of 0.35, aligning with the widely recognized baseline dataset RadioML2016.10a [104], which is frequently employed in modulation classification research.

To facilitate the training process, we represent the complex-valued samples as concatenations of their real and imaginary components. The sample length and sampling frequency of signals are set to 256 and 200 kHz, respectively. Furthermore, in our training dataset, the SNR range varies between -18dB and 0dB with a 2dB increment and there are 5,000 samples with random labels under one SNR level across one type of communication channel.

While Chapter 4 employed the TDL-E channel model to validate joint optimization of denoising and sensing, this chapter requires a more diverse set of channel models to evaluate channel-invariant feature learning. To this end, we have chosen three distinct TDL channel models to serve as our source domain Ω_1, where $|\Omega_1| = 3$: the Extended Pedestrian A model (EPA), the Extended Vehicular A model (EVA), and Extended Typical Urban model (ETU) [118, 119]. While these models share the fundamental TDL structure with TDL-E from Chapter 4, they represent different wireless scenarios with distinct delay spreads and Doppler characteristics. These models are denoted as {Ds_EPA, Ds_EVA, Ds_ETU}. It is worth noting that $|\Omega_1|$ can be increased to encompass more typical wireless scenarios, which could

TABLE 5.1: Channel specifications

Name	$f_{Doppler}$	**Channel Model**
Ds_EPA	5 Hz	Delays: [0, 30, 70, 90, 110, 190, 410] ns Gains: [0, -1, -2, -3, -8, -17.2, -20.8] dB
Ds_EVA	70 Hz	Delays: [0, 30, 150, 310, 370, 710, 1090, 1730, 2510] ns Gains: [0, -1.5, -1.4, -3.6, -0.6, -9.1, -7, -12, -16.9] dB
Ds_ETU	70 Hz	Delays: [0, 50, 120, 200, 230, 500, 1600, 2300, 5000] ns Gains: [-1, -1, -1, 0, 0, 0, -3, -5, -7] dB
Dt_Rayleigh	8 Hz	Delays: [0, 0.2, 3, 9] µs Gains: [0, -7, -2, -1] dB
Dt_Rician	4 Hz	K factor: 4 Delays: [0, 0.25, 3, 8] µs Gains: [0, -2, -10, -3] dB

potentially enhance the robustness of the proposed method. This selection provides a relative comprehensive representation of various wireless communication scenarios. Specifically, the EPA model simulates urban pedestrian environments, characterized by relatively low mobility and moderate multipath conditions. The EVA model represents vehicular communication scenarios, featuring higher mobility and more complex multipath propagation. Lastly, the ETU model represents highly scattered urban environments with significant delay spread, emulating the most challenging urban canyon effects and dense multipath propagation. By incorporating these three models, source domains encompass a wide range of mobility conditions, from stationary to high-speed scenarios, as well as varying degrees of multipath complexity. This diverse set of channel models, encompassing varying mobility conditions and multipath complexities, is specifically chosen to challenge and enhance the training process of ER-SNet. By exposing the network to significantly different channel distributions during training, we aim to force ER-SNet to extract channel-invariant features. This approach is designed to improve the network's ability to generalize across a wide spectrum of real-world wireless communication environments, even those not explicitly represented in the training set. The specific parameters for these channel models, including maximum Doppler shift, multi-path delays, and gains, are delineated in Table 5.1.

To assess the generalization capabilities of our proposed ER-SNet across different unknown channel environments, we introduce a separate prediction space Ω_2. This space consists of two synthesized simplified channel models: {Dt_Rayleigh, Dt_Rician} [120]. Importantly, these models are distinct from those in the training space, ensuring $\Omega_2 \cap \Omega_1 = \emptyset$.

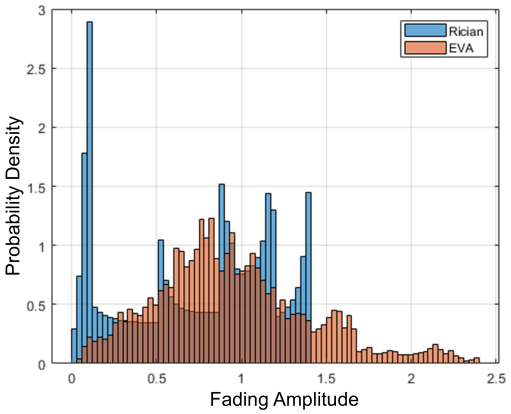
To quantify the diversity and distinctiveness of the source and target fading channels, we computed the Kolmogorov-Smirnov (KS) distances between their cumulative distribution functions (CDFs), which is a statistical metric that measures the maximum difference between the CDFs of two distributions. A higher KS distance indicates greater divergence between the two distributions, while a lower KS distance implies similarity. Table 5.2 summarizes the KS distances for various channel pairs, providing a quantitative measure of the variability among the channel models:

TABLE 5.2: Kolmogorov-Smirnov (KS) distances between cumulative distribution functions (CDFs) of the fading channel models in the source (Ω_1) and target (Ω_2) spaces.

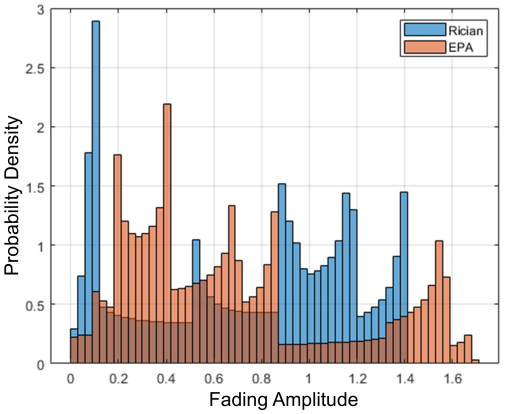
Channel Pair	**KS Distance**
Dt_Rician vs. Ds_EVA	0.18671
Dt_Rician vs. Ds_EPA	0.24631
Dt_Rician vs. Ds_ETU	0.20293
Dt_Rayleigh vs. Ds_EVA	0.31789
Dt_Rayleigh vs. Ds_EPA	0.13535
Dt_Rayleigh vs. Ds_ETU	0.23664

The KS distance analysis confirms the variability between the source channels target channels. Additionally, Figure 5.2 provides a visual comparison of the probability density function (PDF) histograms for Ω_1 and Ω_2. These quantitative and visual analyses ensure that the training and prediction spaces are sufficiently distinct, allowing for a rigorous evaluation of domain generalization capabilities.

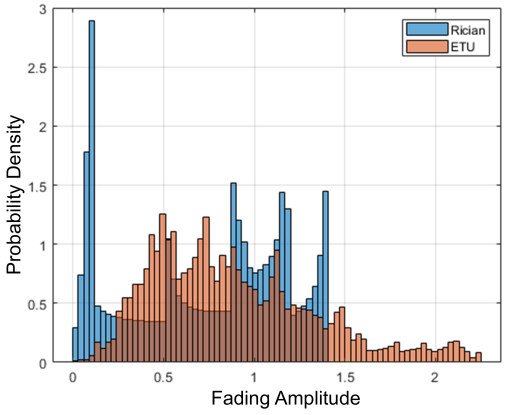
This configuration enables a thorough assessment of the proposed ER-SNet's ability to generalize to previously unseen channel conditions, validating its potential for real-world applications in diverse and dynamic wireless environments.



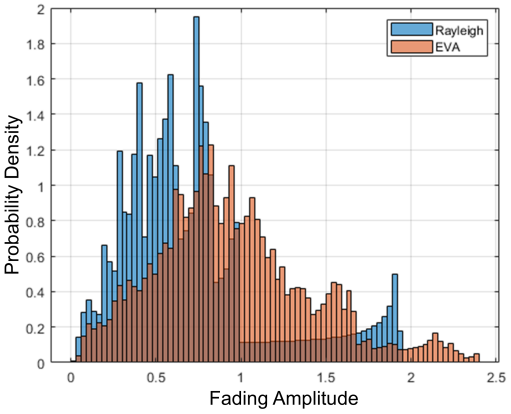
(a) Dt_Rician vs. Ds_EVA.



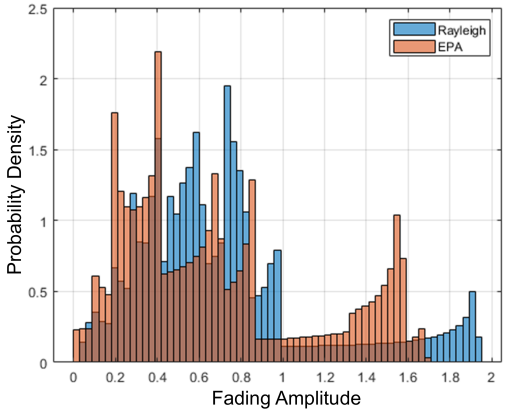
(b) Dt_Rician vs. Ds_EPA.



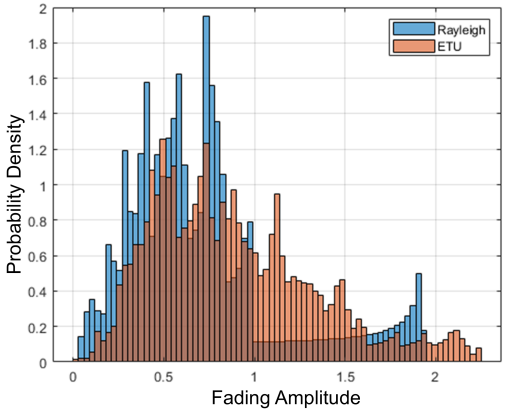
(c) Dt_Rician vs. Ds_ETU.



(d) Dt_Rayleigh vs. Ds_EVA.



(e) Dt_Rayleigh vs. Ds_EPA.



(f) Dt_Rayleigh vs. Ds_ETU.

FIGURE 5.2: PDF comparisons between source and target fading channels to assess variability in fading characteristics.

5.4.2 Complexity Analysis

A comprehensive analysis of the computational complexity for each approach is presented, focusing on FLOPs and the number of parameters. Table 5.3 provides a detailed comparison of these metrics across the different models. Specifically,

TABLE 5.3: Computational complexity in terms of FLOPs and number of parameters of the proposed approach and baselines

Model	**FLOPs (million)**	**No. of Parameters (million)**
ER-SNet	4.03	0.066
DTCS	2.51	0.23
TF-ResNet	12.99	0.434
JDSS	8.99	0.295
DetectNet	5.87	3.02

the ER-SNet mainly consists of multiple convolutional, deconvolutional, and dense layers. The complexity of 1D-CNN layer is $\mathcal{O}\left(C_{\text{in}} \cdot C_{\text{out}} \cdot k_s \cdot L_{\text{out}}\right)$, where C_{in} and C_{out} represent the number of input and output channels, respectively. k_s stands for the kernel size, and L_{out} denotes the output length. Besides, deconvolutional layers contribute a similar complexity expression, namely, $\mathcal{O}\left(C_{\text{in}} \cdot C_{\text{out}} \cdot k_s \cdot L_{\text{in}}\right)$, where L_{in} is the input length of the layer. Lastly, fully connected layers incur the computational complexity of $\mathcal{O}\left(M_{in} \cdot M_d\right)$, where M_{in} denotes the input dimension of the dense layer, and M_d represents the number of neurons. The analysis reveals that the proposed ER-SNet requires approximately 4.03 million FLOPs for processing a single sample, which is 1.52 million more than DTCS. This increase in computational complexity is attributed to the additional decoder section in ER-SNet, whereas DTCS employs several dense layers for the domain critic and classifier modules. Notably, while ER-SNet has a higher FLOPs count than DTCS, it maintains a lower parameter count, suggesting a more efficient use of model capacity.

5.4.3 Performance Evaluation

In the simulations, we compare the performance of our proposed ER-SNet to that of deep learning-based or other conventional spectrum sensing approaches, including DetectNet, JDSS, TF-ResNet, DTCS, and ED. To ensure a fair performance evaluation of each scheme, the training dataset is maintained consistent. It should be noted that the optimal hyper-parameters of each network have been obtained through extensive numerical simulations.

Figure 5.3 illustrates the comparative performance of various spectrum sensing approaches under different channel conditions at SNR=-12dB, with a false alarm rate $P_{fa} = 10\%$. The ED scheme will be excluded for comparisons in Figure 5.3 since the conventional approach does not require offline training phase. Note that the DetectNet and ResNet are typical DL-based methods, and it is difficult to derive a general function that can extract the common feature of training samples with different channel models. The proposed ER-SNet demonstrates superior performance across all testing scenarios. In the in-domain evaluation, where the testing data are drawn from the same distribution as the training data, ER-SNet achieves a probability of detection of approximately 77%, which is around 7% to 14% higher than those of other methods. The encoder $\mathcal{E}$ in ER-SNet can be viewed as a domain augmentation step, which has the ability to deal with the channel variations and help reducing the channel-dependency of the extracted features. Hence, the robustness of ER-SNet against channel variations is evident in the out-domain tests, where the channel conditions of the testing data differ from those of the training data. For both Rician and Rayleigh testing channels, it will have a relatively small accuracy drop from the in-domain performance. JDSS, proposed in Chapter 4, shows competitive performance in the in-domain evaluation, achieving a probability of detection close to that of ER-SNet. However, JDSS struggles to maintain its performance in out-domain scenarios, with a drop of approximately 17% under testing channels. This drop in performance highlights the limitations of JDSS in handling unseen channel conditions effectively. Although DTCS could perform relatively better than TF-ResNet and DetectNet, it requires re-training the model to set the specified target domain in order to obtain good adaptation performance to unknown channel conditions. Similarly, utilizing the strategy of transfer learning, TF-ResNet needs a small portion of labeled samples under the new channel condition for re-training to obtain similar level of probability of detection P_d as training stage. By contrast, the channel variations will cause around 29% drop in detection performance of DetectNet over testing unknown channel model. In general, the proposed ER-SNet can perform better in untrained environments without re-training the model.

Figure 5.4 illustrates the probability of detection (P_d) for various spectrum sensing approaches against different SNR levels at $P_{fa} = 0.1$. The performance is evaluated for both in-domain and out-domain scenarios, with {Dt_Rician} serving as the target domain for comparison while maintaining consistent training sets across

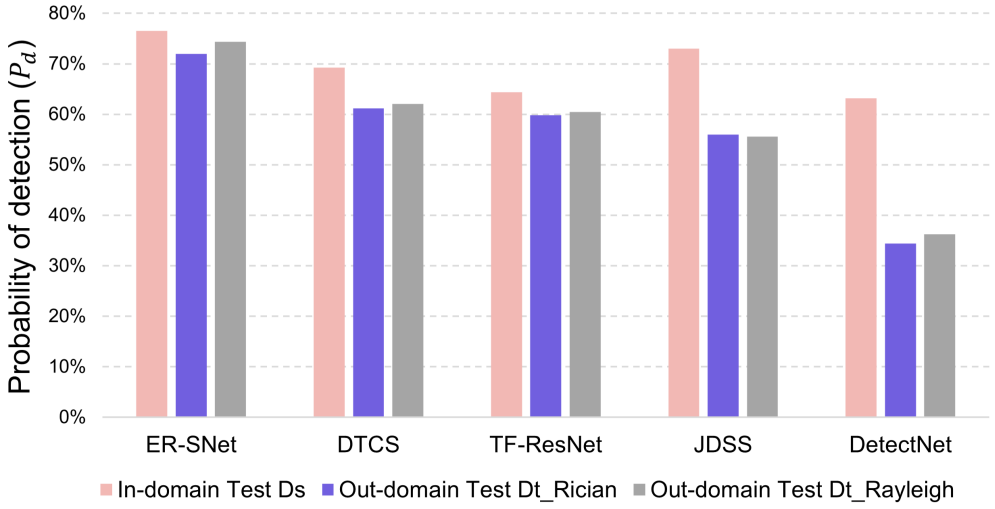
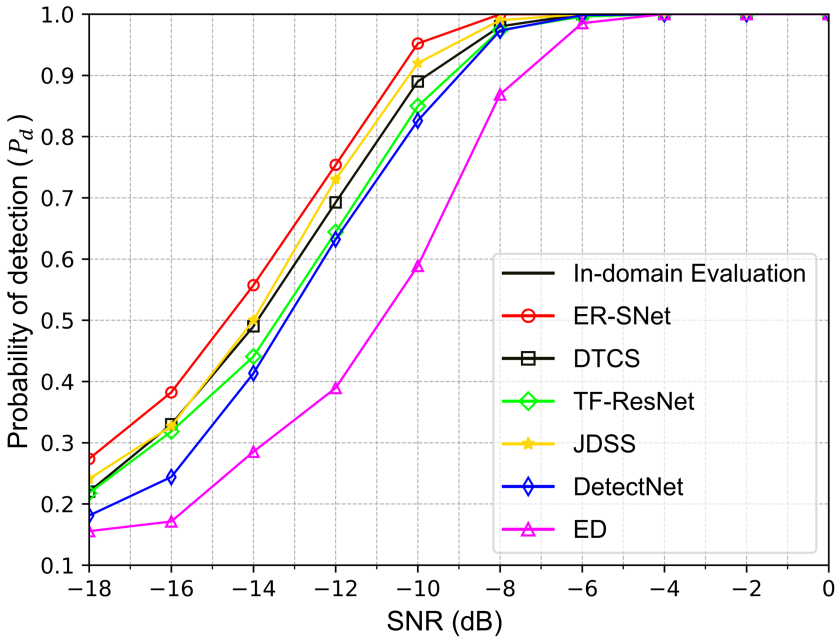


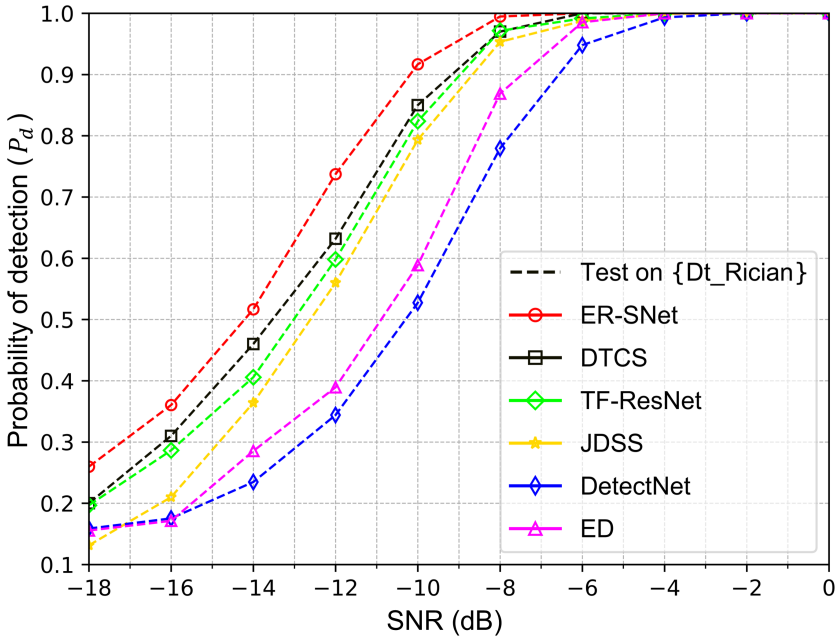
FIGURE 5.3: Sensitivity to different channel models at SNR=-12dB.

all methodologies. It is shown that the proposed ER-SNet demonstrates superior performance across the entire SNR range, particularly in the challenging low SNR regime. At lower SNR levels, P_d of ER-SNet is around 5% to 12% higher than that of other DL techniques. This performance advantage is wider in out-domain evaluations when compared to traditional DL-based methodologies, underscoring the robustness of ER-SNet to channel variations. Besides, DTCS approach exhibits relatively less performance gap from source domain to the target domain, attributed to its pre-determined target domain during the training phase. However, its performance in unknown domains is expected to degrade significantly without retraining. If a small portion of labeled samples from the target domain is available for fine-tuning, TF-ResNet could maintain comparable performance to its in-domain evaluation. JDSS achieves performance comparable to ER-SNet in in-domain scenarios, but when tested on out-domain channels, its performance drops by approximately 17% at an SNR of -12dB. DetectNet, employing a standard cross-entropy loss function for classification, demonstrates a noticeable performance gap compared to ER-SNet and DTCS. This gap widens in the out-domain scenario, highlighting the limitations of conventional deep learning approaches in extracting channel-invariant features. The ED method exhibits the poorest overall performance, although it is consistent across different channel conditions due to its non-learning nature. Notably, ED outperforms DetectNet in the out-domain scenario, emphasizing the challenges faced by non-adaptive deep learning models in unknown channel conditions. These results collectively underscore the efficacy

of ER-SNet in addressing the challenges of spectrum sensing across diverse channel conditions, particularly its ability to maintain relatively robust performance without the need for domain-specific retraining or adaptation.



(a) In-domain evaluation performance.



(b) Target domain prediction performance.

FIGURE 5.4: Comparisons of detection performance at $P_{fa} = 0.1$ between (1) in-domain evaluation and (2) target domain prediction.

Figure 5.5 shows the ROC curve for various spectrum sensing approaches under unknown Rayleigh channel conditions at an SNR of -12dB, which illustrates the

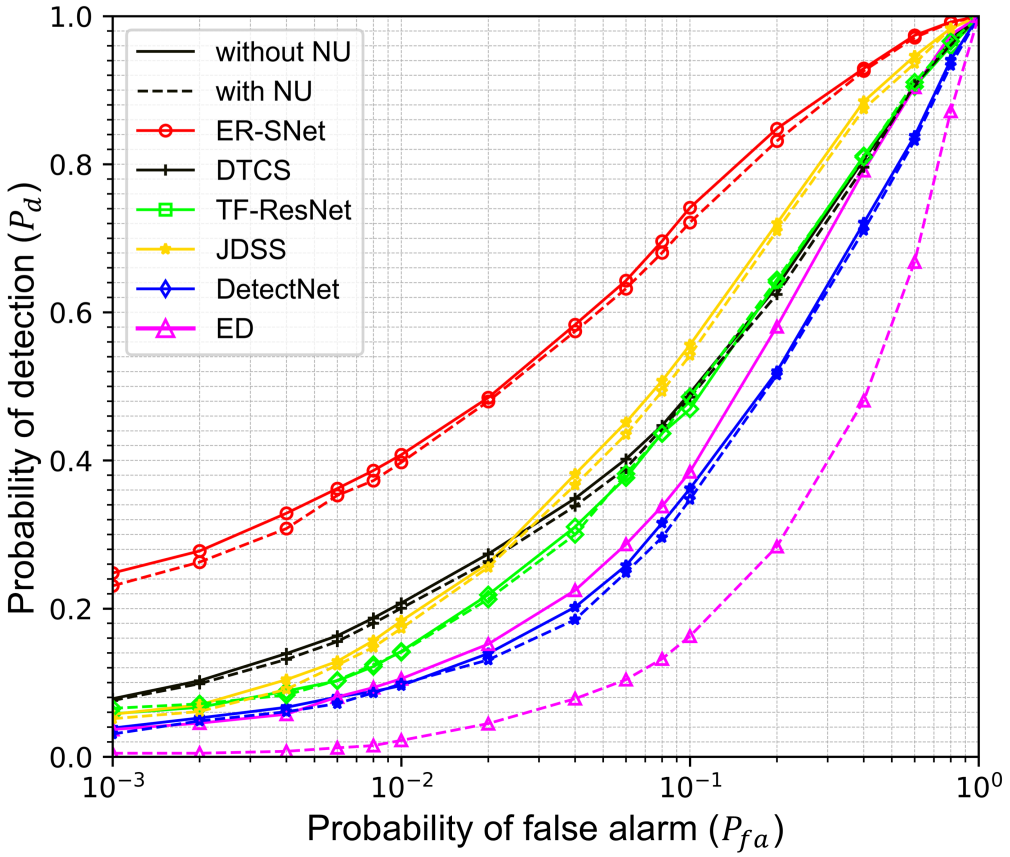


FIGURE 5.5: ROC curves at SNR=-12dB for prediction over unknown Rayleigh channel: {Dt_Rayleigh}.

trade-off between the probability of detection (P_d) and the probability of false alarm (P_{fa}). The proposed ER-SNet demonstrates superior performance across the entire range of P_{fa} values. At a standard operating point of $P_{fa} = 0.1$, ER-SNet achieves a P_d of approximately 75%, significantly outperforming other methods which range from 36% to 56%. It is important to note that in this comparison, all methods are evaluated without their respective domain adaptation capabilities to ensure a fair baseline comparison. Specifically, TF-ResNet is tested without its transfer learning mechanism, essentially functioning as a traditional DL-based method. Similarly, DTCS is evaluated on an unseen channel without pre-determination of its target domain. To evaluate robustness under more realistic conditions, the study incorporates a NU factor of 2dB ($\varepsilon = 1.58$). In this scenario, the estimated noise power $\hat{\sigma}_n^2$ is modeled as uniformly distributed over $\left[\frac{1}{\varepsilon}\sigma_n^2, \varepsilon\sigma_n^2\right]$, where ε denotes the NU factor and σ_n^2 represents the actual noise power. The performance of all methods under NU conditions is denoted by dashed lines in the figure. The ED method exhibits the most significant performance degradation under NU conditions, highlighting

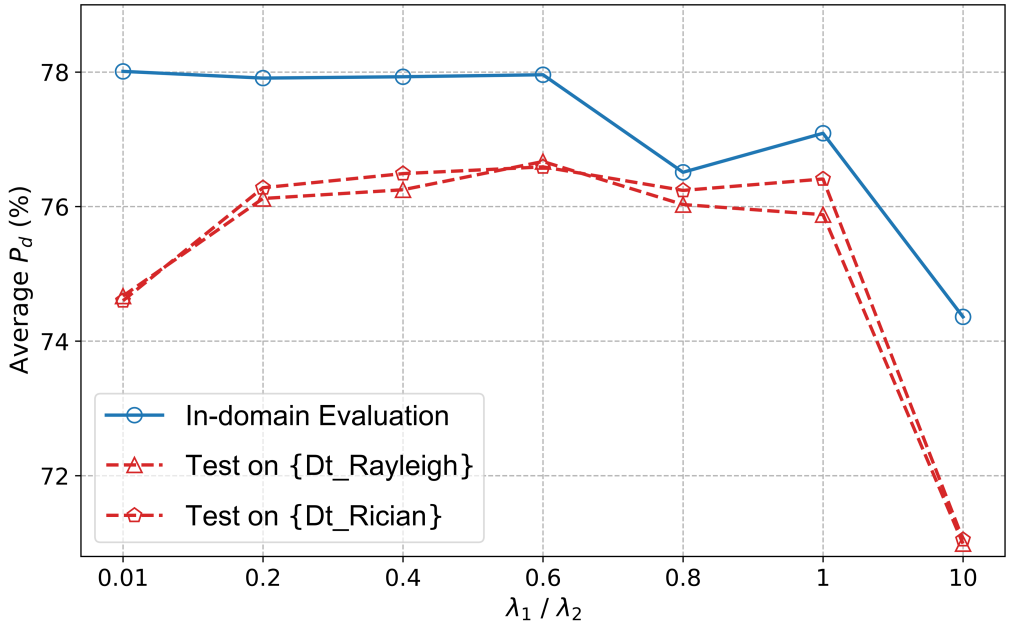


FIGURE 5.6: Impact of weighting factor ratio λ_1/λ_2 on average probability of detection (P_d) for in-domain evaluation and testing channels: {Dt_Rayleigh, Dt_Rician}.

its sensitivity to noise power estimation errors. In contrast, the deep learning-based approaches, including ER-SNet, DTCS, TF-ResNet, JDSS, and DetectNet, demonstrate greater resilience to NU, as they do not rely explicitly on noise power estimation for detection. Notably, while the DTCS method shows competitive performance in pre-determined target domains, its effectiveness diminishes in the unknown Rayleigh channel scenario. This observation underscores the limitations of DTCS approach under different testing channel conditions, which needs retraining for each specific target domain. The traditional DL-based approach DetectNet shows performance inferior to ED in the absence of NU. However, it maintains more stable performance under NU conditions, outperforming ED in this more challenging scenario. These results collectively validate the detection reliability and efficiency of the proposed ER-SNet, demonstrating its ability to maintain high performance across diverse and challenging spectrum sensing scenarios, including those with noise uncertainty.

5.4.4 Hyper-Parameter Analysis

Figure 5.6 illustrates the sensitivity of the ER-SNet model to variations in the weighting factor ratio λ_1/λ_2, where λ_2 is fixed at 1 and λ_1 ranges from 0.01 to 10.

TABLE 5.4: Average probability of detection (P_d) (%) on testing channels as a function of the number of source domains $|\Omega_1|$.

Number of source domains	**1**	**2**	**3**	**4**	**5**
{Dt_Rayleigh}	73.48%	75.83%	76.67%	76.93%	77.01%
{Dt_Rician}	72.94%	75.33%	76.59%	76.79%	76.94%

This analysis provides insights into the balance between the contrastive loss and the cross-entropy loss in the model's overall performance.

For $\lambda_1/\lambda_2 < 0.6$, the in-domain evaluation performance remains relatively stable, while the performance on testing channels shows a consistent improvement. This trend indicates that as λ_1 increases, the contrastive loss begins to exert a more significant influence, enhancing the model's generalization capabilities. At the extreme case of $\lambda_1/\lambda_2 = 0.01$, where CE loss dominates, the ER-SNet essentially degrades to a conventional deep learning architecture without domain augmentation, resulting in reduced generalizability. Conversely, when $\lambda_1/\lambda_2 > 0.6$, a notable decrease in detection performance is observed across both source and testing channels. This decline can be attributed to the diminished role of the CE loss, which impairs the training efficacy of classifier $\mathcal{H}$. The analysis reveals an optimal performance point at $\lambda_1/\lambda_2 \approx 0.6$, suggesting a balanced contribution from both loss components.

Table 5.4 presents the impact of increasing the number of source domains ($|\Omega_1|$) on the average detection performance for testing channels. The analysis extends to a maximum of five source channel conditions, maintaining approximately equal numbers of training samples across conditions to ensure fair comparison.

The results demonstrate a consistent improvement in average P_d as the number of source domains increases. This trend is observed for both Rayleigh and Rician testing channels, with the Rayleigh channel showing slightly higher performance across all domain configurations. Notably, when only one source domain is used, the ER-SNet achieves an average P_d of 73.48% for the Rayleigh channel and 72.94% for the Rician channel. The lower performance can be attributed to the variance between the training and testing environment distributions. However, ER-SNet performs reasonably well with only one training domain due to its contrastive learning mechanism, which extracts channel-invariant features by minimizing intra-class variance

and maximizing inter-class separability. Besides, inherent variability within a single training domain, such as differences in signal-to-noise ratio or fading conditions, provides the model with some level of robustness to unseen scenarios. For configurations with two to five source domain, the improvement, while modest, suggests that incorporating a diverse set of source domains enhances the model's ability to generalize to unseen target domains. Specifically, for the Rayleigh testing channel, the average P_d improves from 75.83% with two source domains to 77.01% with five source domains, representing a 1.18% increase. Similarly, for the Rician testing channel, performance improves from 75.33% to 76.94%, with a 1.61% increase.

These findings underscore the importance of training data diversity in improving the ER-SNet's representation learning capabilities. The incremental gains observed with each additional source domain suggest that the model becomes increasingly adept at extracting channel-invariant features, thereby enhancing its robustness to varied channel conditions in spectrum sensing tasks.

5.5 Conclusion

In this chapter, a novel network architecture and the training strategy have been designed to improve the detection performance and predict the state of PU over different channel conditions instead of re-training the model with numerous labeled samples. By combining the contrastive loss with classification loss, the optimized structure can effectively extract the channel invariant features for further accurate prediction of spectrum status. Numerical results have shown that the proposed channel-robust spectrum sensing scheme can achieve a greater probability of detection at the cost of a slightly higher computational complexity and is adaptable to other untrained channel environments.

The extensive experimentation with ER-SNet has provided significant insights into the development of channel-invariant spectrum sensing systems. Our analysis reveals that the diversity of source domains plays a crucial role in model generalization. Specifically, increasing the number of source domains from two to five leads to consistent improvement in detection performance across unknown channels, with gains of 1.18% and 1.61% for Rayleigh and Rician testing channels, respectively.

Furthermore, the weighting between contrastive loss and classification loss significantly impacts the model's performance. When the contrastive loss weight substantially exceeds the classification loss weight, we observe a notable decrease in performance, indicating that overly emphasizing channel-invariant feature learning could potentially compromise the model's ability to make accurate classification decisions. Although the evaluation of ER-SNet has demonstrated its robustness across specific channel environments, further exploration of more dynamic and complex scenarios such as highly time-varying channels could be conducted in the future.

While ER-SNet represents a significant advancement in spectrum sensing technology, it still relies on a substantial amount of labeled data for training. This dependency on labeled data presents challenges in real-world scenarios where acquiring high-quality labeled samples can be difficult, time-consuming, and costly.

To address these limitations and further enhance the practicality of spectrum sensing in cognitive radio networks, the next chapter will introduce a novel self-supervised learning approach. This method aims to leverage the abundance of unlabeled data typically available in radio environments while minimizing the need for extensive manual labeling. By exploring innovative pretext tasks and representation learning techniques, we seek to develop a self-supervised spectrum sensing framework that can achieve performance comparable to fully supervised methods, even with an extremely limited percentage of labeled data.

Chapter 6

Self-Supervised Learning for Spectrum Sensing with Limited Labeled Data

6.1 Introduction

Building upon the advancements presented in the previous chapters, which are based on supervised learning techniques, this chapter explores methods to achieve comparable results with significantly fewer labeled samples. Motivated by the need to reduce data dependence, we investigate the potential of self-supervised learning. Using the same synthetic data generation framework as previous chapters, we observe that our proposed DC4S approach can maintain high detection accuracy while requiring only a small fraction of the labeled data needed by traditional supervised approaches.

To achieve this goal, we propose a novel self-supervised learning framework for spectrum sensing, named dual-contrast self-supervised spectrum sensing (DC4S), which aims to achieve performance comparable to fully supervised learning methods while requiring only a limited number of labeled samples. The framework tailors data augmentation techniques specifically for communication signals, generating two distinct yet correlated views of the input data. After extracting spatial and temporal features, a cross-view prediction module is employed, which leverages past latent features from one view's representation to forecast the future state of another

view. This challenging predictive task forces the model to develop robust representations that are resilient to the variances stemming from different augmentations and numbers of historical timesteps. Additionally, a signal-context contrasting module is incorporated to further refine these representations by enhancing intra-sample context similarities while reducing inter-sample context similarities. With the representations extracted through a well-trained CNN encoder, a linear classifier is refined through supervised fine-tuning with a limited number of labeled samples. Finally, comprehensive simulations and experiments are carried out for performance evaluation and robustness validation. To sum up, the contributions of this work can be summarized as follows:

- The proposed DC4S framework consists of a pre-training phase using unlabeled samples, where an encoder is trained to acquire the capability of extracting general features from the input data. Subsequently, in the fine-tuning stage, both pre-trained encoder and the linear classifier are further refined using labeled samples to adapt to the specific downstream task.
- A comparative analysis of various data augmentation methods tailored for communication signals is presented. Based on experimental results, scaling and reverse augmentations are incorporated into the DC4S framework.
- An integration of a cross-view prediction module and a signal-context contrasting module is employed in the proposed framework, which facilitates the learning of discriminative representations from communication signals through a combined loss function.
- Extensive simulations validate the effectiveness of DC4S using limited labeled datasets, showing superior performance compared to traditional supervised learning, SSDNN [82], and CNN-CSL [121], across various metrics such as SNR levels and the probability of false alarms.
- The DC4S framework is thoroughly evaluated under diverse scenarios, including different signal modulation types, over-sampling rates, subcarrier numbers and channel conditions, showcasing its superior representation ability and robustness in various environments.

The proposed framework builds upon recent advancements in self-supervised learning for wireless communications, extending beyond traditional contrastive learning approaches. By incorporating temporal dynamics and contextual information, DC4S seeks to capture the intrinsic characteristics of communication signals more effectively than existing methods.

The remainder of this chapter is structured as follows. Section 6.2 delineates the system model and elucidates the problem formulation. Section 6.3 expounds on the data augmentation techniques, pre-training process, and fine-tuning methodology, encompassing a detailed description of the DC4S network architecture. Section 6.4 presents a comprehensive performance evaluation, including robustness validation, hyper-parameter analysis, and an examination of augmentation selection criteria. Finally, Section 6.5 summarizes the main contributions and findings of this research.

6.2 System Model and Problem Illustration

This section briefly describes the OFDM signal model under non-cooperative spectrum sensing scenario and the problem illustration of self-supervised spectrum sensing.

6.2.1 System Model

In this work, we assume that there exists only one antenna to receive OFDM signals, which is a non-cooperative spectrum sensing scenario. OFDM is a highly efficient modulation technique that distributes data across multiple subcarriers, denoted as N_{sc} in number. The modulation process employs QPSK to convert a random bit sequence A_m into a symbol sequence $C_{m,k}$, where k represents the sequence index within the m^{th} OFDM block. The baseband time-domain representation of the m^{th} OFDM block, denoted as $\mathbf{x}_m \in \mathbb{C}^{1\times Nsc\zeta}$, can be mathematically expressed as follows:

$$\mathbf{x}_m = \sum_{k=1}^{N_{sc}\zeta} \frac{1}{\sqrt{N_{sc}\zeta}} e^{j\frac{2\pi kn}{N_{sc}\zeta}} C_{m,k}, \quad n = 1, \cdots, N_{sc}\zeta, \tag{6.1}$$

where $\zeta = 4$ denotes the oversampling factor, and N_{sc} is set as 256 in our experiments, establishing the OFDM block length L as $N_{sc}\zeta = 1024$. Our simulations adopt the TDL-E channel model, which aligns with the typical 5G communication environments, and we extend our robustness analysis across different channel conditions, specifically from TDL-A to TDL-D channels. These channel models include a range of scenarios featuring LoS and NLoS elements, with the channel coefficients for each tap modeled according to a standard fading model equation in compliance with the TR38.901 specifications. The received m^{th} complexed signal $\mathbf{r}_m^{recv}$ is initially normalized and then divided into its in-phase and quadrature components. Subsequently, the real and imaginary components of the OFDM block are concatenated to form a two-dimensional real-valued vector, which serves as the input to the network. This vector can be represented as:

$$\mathbf{r} = \begin{bmatrix} \mathfrak{Re}\{\mathbf{r}^{recv}\} \\ \mathfrak{Im}\{\mathbf{r}^{recv}\} \end{bmatrix} = \begin{bmatrix} r_I[0] & r_I[1] & \cdots & r_I[L-1] \\ r_Q[0] & r_Q[1] & \cdots & r_Q[L-1] \end{bmatrix}, \tag{6.2}$$

where $\mathfrak{Re}\{\mathbf{r}^{recv}\} \in \mathbb{R}^L$, $\mathfrak{Im}\{\mathbf{r}^{recv}\} \in \mathbb{R}^L$, and $\mathbf{r} \in \mathbb{R}^{2\times L}$.

6.2.2 Problem Illustration

The spectrum sensing problem can be viewed as a binary hypothesis testing problem. After transmission through a multipath fading channel, the received signal at the SU, denoted as $\mathbf{r}_m$, can be described by the following hypotheses:

$$\begin{cases} H_0 : \mathbf{r}_m^{recv} = \mathbf{w}_m, \\ H_1 : \mathbf{r}_m^{recv} = \mathbf{h}_m * \mathbf{x}_m + \mathbf{w}_m, \end{cases} \tag{6.3}$$

where hypothesis H_0 denotes that the PU is assumed to be absent, while hypothesis H_1 indicates the presence of the PU. The term $\mathbf{w}_m \sim CN(0, \sigma_w^2)$ represents complex AWGN with zero mean and variance σ_w^2. The symbol $*$ denotes the convolution operation, and $\mathbf{h}_m$ represents the multi-path channel response, which is assumed to be constant within a single OFDM block. Following the pre-processing of the complex received signals $\mathbf{r}^{recv}$, the decision rule for spectrum sensing, utilizing the two-dimensional real-valued vector $\mathbf{r}_m$, can be expressed as:

$$f_{H_1|\boldsymbol{\phi}}(\mathbf{r}_m) \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{6.4}$$

where $f_{H_1|\boldsymbol{\phi}}(\cdot)$ represents the likelihood of hypothesis H_1, conditioned on the well-trained model weights $\boldsymbol{\phi}$, which is obtained through the supervised learning-based fine-tune network $f(\cdot)$ of the proposed DC4S framework. The state of the m^{th} PU signal is determined by comparing $f_{H_1|\boldsymbol{\phi}}(\mathbf{r}_m)$ with a predefined detection threshold η. As in previous chapters, the decision threshold is calculated using the Monte-Carlo method to ensure the fixed P_{fa} at 10%.

To effectively evaluate the performance of spectrum sensing, we employ two critical metrics: the probability of detection P_d, which quantifies the system's ability to correctly identify the presence of the PU signal, and the probability of false alarm, assessing the rate at which the system erroneously indicates the presence of the PU signal in its absence. These metrics are pivotal for optimizing the trade-off between utilizing the spectrum efficiently and minimizing interference with the PU. Hence, P_d and P_{fa} can be formulated as:

$$\begin{cases} P_d = \mathrm{P_r}\left\{ f_{H_1|\phi}(\mathbf{r}) > \eta \mid H_1 \right\}, \\ P_{fa} = \mathrm{P_r}\left\{ f_{H_1|\phi}(\mathbf{r}) > \eta \mid H_0 \right\}, \end{cases} \tag{6.5}$$

where $\mathrm{P_r}\{\cdot\}$ denotes the probability of a specified event.

Assume that the received signal dataset comprises an unlabeled dataset $\mathcal{D}^U = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$ and a labeled dataset $\mathcal{D}^L = \{(\mathbf{r}_{N+1}, y_{N+1}), \ldots, (\mathbf{r}_{N+M}, y_{N+M})\}$, where typically, $N \gg M$, and $y_i \in [0, 1]$ denotes the hypotheses H_0 or H_1. The objective of the proposed DC4S framework is to initially pre-train the neural network, using the unlabeled dataset $\mathcal{D}^U$ to achieve an optimal feature extraction module. Subsequently, in the fine-tuning stage, the labeled dataset $\mathcal{D}^L$ is employed to refine the pre-trained CNN encoder alongside a linear classifier to detect the presence or absence of the PU.

6.3 The Proposed Methodology

6.3.1 Communication Signal Augmentation

Signal augmentation is an important component in our contrastive self-supervised learning framework, as the choice of data augmentation operations significantly influences the quality of learned representations. Prior research has explored various

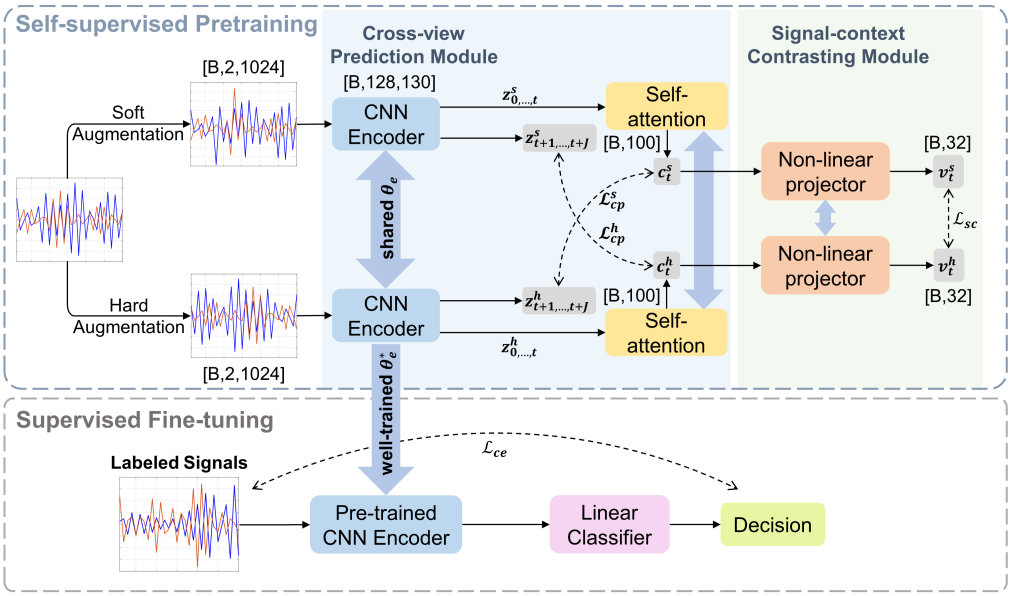


FIGURE 6.1: The proposed DC4S framework.

time-domain data augmentation techniques within DL-based algorithms, as illustrated in the works [122–124]. In this study, we classify augmentation strategies into two categories: soft augmentation and hard augmentation. Each category encompasses techniques specifically tailored for communication signals, particularly in the context of spectrum sensing.

6.3.1.1 Soft Augmentation

Soft augmentation methods apply subtle and minimally intrusive transformations that preserve the overall structure and content of the communication data. These techniques are devised to emulate minor variations that could manifest due to slight environmental fluctuations or discrepancies in equipment, thus enabling the model to learn more robust and invariant features.

- Jittering: Adds random Gaussian noise to the signals, simulating natural variations and background interference.
- Scaling: Adjusts the amplitude of the signals by a random factor within a specified range, simulating variations in signal strength caused by factors such as different transmission powers.

6.3.1.2 Hard Augmentation

Conversely, hard augmentation methods employ substantial transformations that significantly modify the signal's original architecture. These interventions are intended to encourage the model to learn high-level features that are invariant to drastic alterations in the data.

- Time Shifting: Shifts the signal in the time domain, simulating delays or variations in the transmission environment.
- Permutation: Splits and shuffles the time-series data, simulating scenarios where packets or data chunks might arrive out of order due to network routing or buffering processes.
- Time Warping: Applies a non-linear distortion to the time axis, simulating stretching or compression of the signal caused by multipath propagation or Doppler effects.
- Fast Fourier Transform (FFT): Transforms the signal into the frequency domain, providing a time-frequency perspective for analysis.
- Reverse: Inverts the temporal sequence of the signal, ensuring the model's performance is not constrained by the directionality of the sequence.

TABLE 6.1: Augmentation parameter descriptions applied for communication signals

Augmentation	**Parameters**
Jittering	$\sigma = 0.05$
Scaling	Mean=2, $\sigma = 0.1, 0.2$
Time shift	[-150,150]
Permutation	Max segments=9
Timewarping	No. speed change=7, max/min speed=4

Table. 6.1 presents the parameter descriptions of these signal augmentations, where σ is the standard deviation. The empirical analysis reveals that the set of soft and hard augmentations, specifically scaling as a soft augmentation combined with

scaling and reversing as hard augmentations, constitutes the most efficacious augmentation strategy (refer to Section 6.4.5 for details). For the signal $\mathbf{r} \in \mathbb{R}^{2\times L}$ in the unlabeled training set $\mathcal{D}^U$, the signal undergoes small-scale scaling to yield the augmented signal $\hat{\mathbf{r}}_i^s \in \mathbb{R}^{2\times L}$, and simultaneously, reversing followed by large-scale scaling to produce $\hat{\mathbf{r}}_i^h \in \mathbb{R}^{2\times L}$. These augmented signals are then processed through a CNN-based encoder $\mathcal{E}(\cdot)$ with the shared network parameters, for further facilitating cross-view prediction and signal-context contrasting tasks.

6.3.2 The Pre-Training of DC4S Framework

In this section, we will introduce the pre-training stage of the DC4S framework, as illustrated in Figure 6.1. Firstly, the framework generates two correlated yet distinct views of the input data, $\hat{\mathbf{r}}_i^s$ and $\hat{\mathbf{r}}_i^h$, through a combination of soft and hard augmentations. These augmented views are then encoded by $\mathcal{E}_{\boldsymbol{\theta}_e}(\cdot)$ to obtain high-dimensional latent representations. Then, the cross-view prediction module $\mathcal{S}_{\boldsymbol{\theta}_s}(\cdot)$, which comprises self-attention blocks, performs a cross-view prediction task where historical data of one augmented view is used to predict the future of the other. It allows the model to extract the temporal features of the communication signals. Subsequently, the outputs are projected to a lower-dimensional space via a projection head $\mathcal{P}_{\boldsymbol{\theta}_p}(\cdot)$, which is composed of non-linear fully connected layers. Finally, in the signal-context contrasting module, the agreement between the contexts of lower-dimensional features is maximized. In the following, we will delve into the details of each module.

6.3.2.1 CNN Encoder

The feature encoder extracts latent representations from the soft and hard augmented views of each input sample $\mathbf{r}$, denoted as $\hat{\mathbf{r}}^s$ and $\hat{\mathbf{r}}^h$, respectively. As illustrated in Fig. 6.2, the encoder's structure comprises three sequential CNN blocks. Each block includes a 1D convolutional layer, succeeded by a batch normalization layer, a rectified linear unit (ReLU) activation function, a max pooling layer, and a dropout layer to prevent overfitting. The encoder maps the input $\mathbf{r}$ to a high-dimensional latent space, producing $\mathbf{z} = \mathcal{E}_{\boldsymbol{\theta}e}(\mathbf{r})$, where $\mathbf{z} \in \mathbb{R}^{B\times 128\times 130}$. Here, B

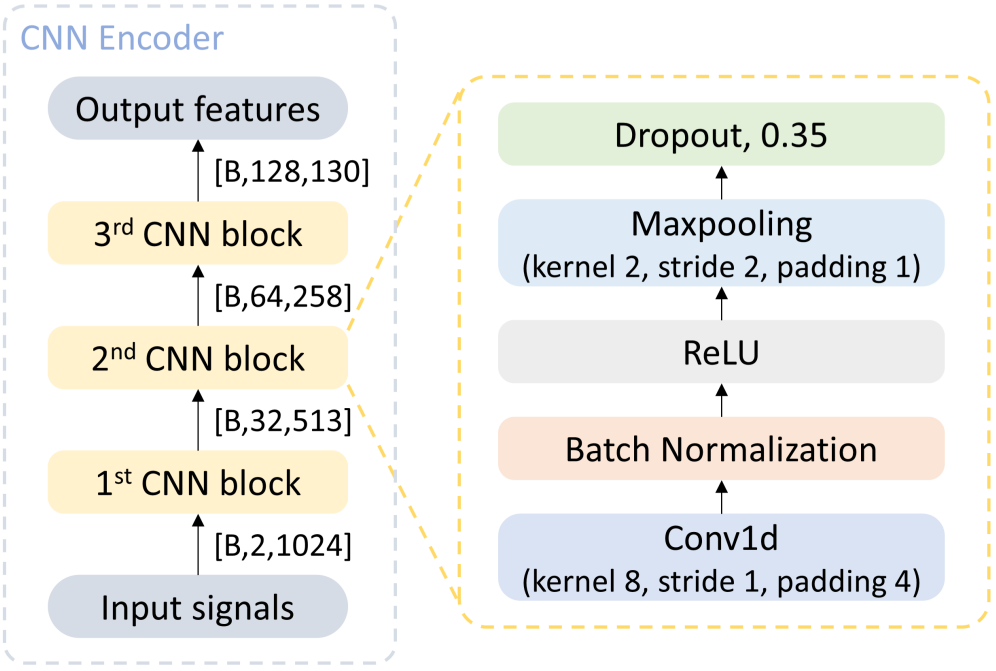


FIGURE 6.2: The structure of CNN feature encoder.

represents the batch size, 128 denotes the feature dimension, and 130 is the sequence length. This process yields $\mathbf{z}^s$ and $\mathbf{z}^h$ for the soft and hard augmentations, respectively, which are subsequently fed into the cross-view prediction module.

6.3.2.2 Cross-view Prediction Module

The cross-view prediction module leverages multiple self-attention blocks to capture temporal characteristics within the latent space through a softmax-based contrastive loss, which enhances model discrimination by encouraging similarity between corresponding predictions and ground truths while reducing similarity among non-corresponding ones. This approach aligns well with the temporal structure of communication signals, as the cross-view prediction task encourages the model to learn stable features that remain robust under various signal transformations. Unlike traditional sequence-to-sequence approaches that focus on temporal dependencies within a single context, this cross-view prediction approach captures both temporal coherence and representation invariance, enhancing the model's resilience to real-world signal distortions and noise. In our framework, the CNN encoder processes the input to yield latent representations $\mathbf{z}$, from which subsets are selectively encoded and decoded across distinct views, guided by the timestep parameter, to

facilitate cross-view prediction task. Specifically, the model $\mathcal{S}_{\boldsymbol{\theta}_s}$ aggregates the sequence $\mathbf{z}_{\leq t}$ into a context vector $\mathbf{c}_t = \mathcal{S}_{\boldsymbol{\theta}_s}(\mathbf{z}_{\leq t})$, where $\mathbf{c}_t \in \mathbb{R}^{B\times h}$. Here, t represents the random sampling of total timesteps, and h denotes the hidden dimension. This context vector $\mathbf{c}_t$ serves as a basis to predict subsequent timesteps $\mathbf{z}_{t+1}$ to $\mathbf{z}_{t+J}$ of the other correlated view. A log-bilinear model is employed to maintain the mutual information between the input $\mathbf{z}_{t+j}$ and context vector $\mathbf{c}_t$, defined as $f_k(\mathbf{z}_{t+j}, \mathbf{c}_t) = \exp(\mathbf{z}_{t+j}(\mathcal{W}_j(\mathbf{c}_t))^T)$. Here, $\mathcal{W}_j$ maps $\mathbf{c}_t$ to the same dimension as $\mathbf{z}$. Our approach introduces a cross-view prediction task, leveraging representations derived from hard and soft augmentations to predict future timesteps across augmented views, thus enhancing mutual information fidelity. The devised contrastive loss function aims to maximize the alignment between the predicted and true representations of the same sample, while concurrently reducing alignment with other samples in the mini-batch, denoted as $\mathcal{N}_{t,j}$, thereby enabling the model to extract intricate temporal relationships within the data effectively [125].

Hence, we compute the two losses $\mathcal{L}_{cp}^h$ and $\mathcal{L}_{cp}^s$ regarding with prediction using one view's past to another view's future, as follows:

$$\mathcal{L}_{cp}^h = -\frac{1}{J}\sum_{j=1}^{J}\log\frac{\exp(\mathbf{z}_{t+j}^s(\mathcal{W}_j(\mathbf{c}_t^h))^T)}{\sum_{n\in\mathcal{N}_{t,j}}\exp(\mathbf{z}_n^s(\mathcal{W}_j(\mathbf{c}_t^h))^T)}; \tag{6.6}$$

$$\mathcal{L}_{cp}^s = -\frac{1}{J}\sum_{j=1}^{J}\log\frac{\exp(\mathbf{z}_{t+j}^h(\mathcal{W}_j(\mathbf{c}_t^s))^T)}{\sum_{n\in\mathcal{N}_{t,j}}\exp(\mathbf{z}_n^h(\mathcal{W}_j(\mathbf{c}_t^s))^T)}. \tag{6.7}$$

Self-attention blocks are deployed for temporal feature extraction owing to the efficiency and rapid processing capabilities, which is applied in Transformer [103]. Within the cross-view prediction module, these blocks capture global contextual information across the sequence. By introducing a context token c, the self-attention mechanism enables this token to aggregate information from all positions through iterative interactions. Through multiple layers of self-attention, this token evolves to encapsulate a comprehensive global representation of the entire sequence. This global representation facilitates consistency learning across different augmented views of the input signal, allowing the model to align features not only at a local level but also holistically.

As shown in Fig. 6.3, four self-attention blocks are deployed, each of which mainly

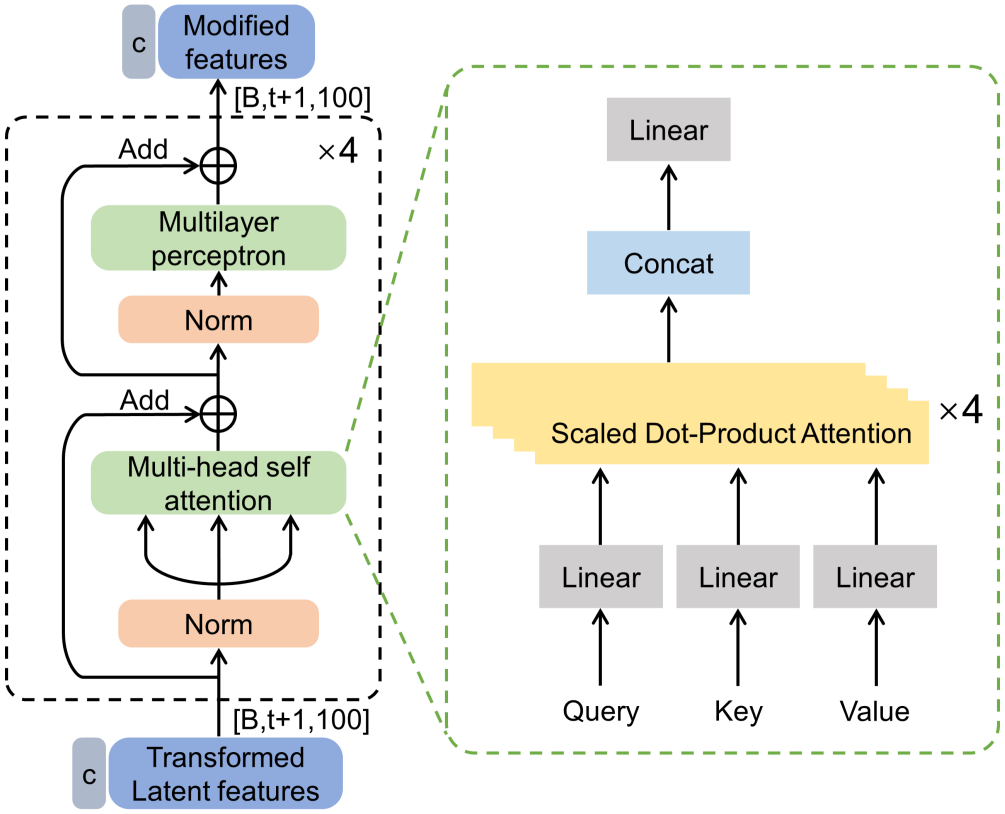


FIGURE 6.3: The structure of self-attention blocks.

comprises a multi-head self-attention (MSA) block succeeded by a multilayer perceptron (MLP) block. The MLP block consists of two dense layers with hidden unit of 64, between which are a ReLU layer and a dropout layer, to enhance model robustness and prevent overfitting. Besides, to facilitate more stable gradient flow, pre-normalization residual connections are employed [126].

Prior to input into the self-attention blocks, we transpose $\mathbf{z}$ to align the sequence length with the expected input format. We implement dynamic timestep selection by randomly sampling $t \in [0, 130 - J]$, where J denotes the number of timesteps to be predicted. Consequently, the input to the self-attention block is given by $\mathbf{z}_{\leq t} \in \mathbb{R}^{B \times t \times 128}$.

These features undergo an initial transformation through a linear layer, projecting them to an embedding space $\tilde{\mathbf{z}} \in \mathbb{R}^{B \times t \times h}$, where $h = 100$ is the hidden dimension size. Inspired by the popular NLP network model BERT [89], we introduce a learnable context token $\mathbf{c} \in \mathbb{R}^{B \times h}$, designed to aggregate information across the entire sequence. These features are then concatenated with the token $\mathbf{c}$, forming

$\Psi_0 = [\mathbf{c}; \tilde{\mathbf{z}}] \in \mathbb{R}^{B\times(t+1)\times h}$, where Ψ_0 represents the augmented input to the first self-attention block. This composite input undergoes a series of self-attention blocks, which can be expressed as:

$$\tilde{\Psi}_i = \mathrm{MSA}(\mathrm{Norm}(\Psi_{i-1})) + \Psi_{i-1}, \qquad 1 \le i \le 4; \tag{6.8}$$

$$\Psi_i = \mathrm{MLP}(\mathrm{Norm}(\tilde{\Psi}_i)) + \tilde{\Psi}_i, \qquad 1 \le i \le 4. \tag{6.9}$$

Delving into structure and procedure details of the MSA blocks, it utilizes four heads to extract intricate dependencies and relationships embedded in the input features. The initial step involves replicating the input feature to generate discrete query, key, and value vectors, denoted as Q_i, K_i, and V_i, respectively, for each head index $i = 1, \ldots, 4$. These vectors undergo individual processing through a dense layer, followed by a scaled dot-product attention mechanism. The output for each head is mathematically formulated as:

$$\mathrm{head}_i = \mathrm{Attention}_i\left(Q_i\mathcal{W}_i^Q, K_i\mathcal{W}_i^K, V_i\mathcal{W}_i^V\right), \tag{6.10}$$

where $\mathcal{W}_i^Q, \mathcal{W}_i^K, \mathcal{W}_i^V \in \mathbb{R}^{25\times25}$ represent the transformation matrices of the fully-connected layer preceding the scaled dot-product attention layer. Let $\widetilde{Q_i}$, $\widetilde{K_i}$ and $\widetilde{V_i}$ denote the results of $Q_iW_i^Q$, $K_iW_i^K$ and $V_iW_i^V$, respectively. Hence, the operation $\mathrm{Attention}_i$ is defined as:

$$\mathrm{Attention_i}\left(\widetilde{Q_i}, \widetilde{K_i}, \widetilde{V_i}\right) = \mathrm{softmax}\left(\frac{\widetilde{Q_i}\widetilde{K_i}^T}{\sqrt{\dim_k}}\right)\widetilde{V_i}, \tag{6.11}$$

where $\dim_k = 25$ denotes the channel size of K_i. Consequently, the output of a specific MSA block can be formulated as:

$$\mathrm{MSA}(Q, K, V) = \mathrm{Concat}\left(\mathrm{head}_i\right)\mathcal{W}^O, \tag{6.12}$$

where $\mathcal{W}^O \in \mathbb{R}^{100\times100}$ represents the weight matrix for the fully connected layer subsequent to the Concat layer.

Finally, the final output $\Psi_4 = [\mathbf{c}_4; \tilde{\mathbf{z}}_4] \in \mathbb{R}^{B\times(t+1)\times h}$ enables the extraction of the aggregated output context vector $\mathbf{c}_4 \in \mathbb{R}^{B\times h}$, which is employed to calculate the prediction loss and also serves as the input of the subsequent signal-context contrasting module.

6.3.2.3 Signal-context Contrasting Module

The signal-context contrasting module further enhances the model's capability to extract discriminative representations, building upon principles of contrastive learning [86]. It employs a non-linear projection head to map the output representations of cross-view prediction module into the space where the contrasting loss is calculated. The normalized temperature-scaled cross-entropy loss (NT-Xent) is utilized as the loss function [127], which is designed to maximize the similarity between positive pairs while minimizing similarity with negative samples within a mini-batch. For a batch of B input samples, with each sample generating a pair of augmented samples, the total count of data points becomes $2B$. $\mathbf{c}_t^{i^+}$ is denoted as the positive counterpart of a specific context $\mathbf{c}_t^i$, which is obtained from the other augmented view of the same input, thereby forming a positive pair $(\mathbf{c}_t^i, \mathbf{c}_t^{i^+})$. Conversely, the remaining $2(B-1)$ samples, originating from different inputs in the same mini-batch, are defined as negative samples for $\mathbf{c}_t^i$, enabling the formation of $2(B-1)$ negative pairs involving $\mathbf{c}_t^i$ and its negative samples. The measure of similarity between two augmented samples is quantified as follows:

$$\operatorname{sim}\left(\mathbf{c}_t^i, \mathbf{c}_t^{i^+}\right) = \frac{\mathbf{c}_t^i \mathbf{c}_t^{i^+}}{\|\mathbf{c}_t^i\| \left\|\mathbf{c}_t^{i^+}\right\|}, \tag{6.13}$$

where $\|\mathbf{c}_t^i\|$ refers to the ℓ_2 norm of $\mathbf{c}_t^i$. The contrasting loss function of a positive pair $\ell(i, i^+)$, which is normalized across the batch to maximize the positive sample's similarity over the cumulative similarity with all $2B-1$ samples, can be expressed as:

$$\ell(i, i^+) = -\log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{c}_t^i, \boldsymbol{c}_t^{i^+}\right)/\tau\right)}{\sum_{m=1}^{2N} \mathbb{1}_{[m \neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{c}_t^i, \boldsymbol{c}_t^m\right)/\tau\right)}, \tag{6.14}$$

where $\mathbb{1}_{[m \neq i]} \in \{0, 1\}$ is an indicator function, and τ represents a temperature parameter. Thus, the loss for all positive pairs can be accumulated and averaged to form the final signal-context contrasting loss functions $\mathcal{L}_{sc}$, as given by

$$\mathcal{L}_{sc} = \frac{1}{2B} \sum_{i=1}^{B} \left[\ell(i, i^+) + \ell(i^+, i)\right]. \tag{6.15}$$

Hence, the total loss function consists of the two cross-view prediction losses and the signal-context contrasting loss, which can be expressed as:

$$\mathcal{L}_{total} = \lambda_1 \cdot (\mathcal{L}_{cp}^{h} + \mathcal{L}_{cp}^{s}) + \lambda_2 \cdot \mathcal{L}_{sc}, \tag{6.16}$$

where λ_1 and λ_2 are weighting factors that optimize the trade-off between the two modules.

6.3.3 Fine-Tuning of DC4S Framework

During the supervised fine-tuning phase, both the pre-trained CNN encoder and a subsequent linear classifier, denoted as $f_{\boldsymbol{\phi}}(\cdot)$, are updated using a limited percentage of labeled samples $(\mathbf{r}_i, y_i) \in \mathcal{D}^L$. The architecture of the classifier is identical to that of the non-linear projection head in the signal-context contrasting module, with the only difference being the output dimension of the last layer, which equals the number of signal classes. The pre-trained CNN encoder can extract general features, while the classifier is solely dependent on the labeled samples for updating its parameters. Given a batch size of B_t, the cross-entropy loss function, which is utilized for the supervised fine-tuning stage of the DC4S framework, can be mathematically expressed as follows:

$$\begin{aligned} \mathcal{L}_{ce}(\boldsymbol{\phi}) = & \frac{1}{B_t} \sum_{i=1}^{B_t} - \left[y_i \cdot \log \left(f_{H_1|\boldsymbol{\phi}} \left(\mathbf{r}_i \right) \right) + \right. \\ & \left. (1 - y_i) \cdot \log \left(1 - f_{H_1|\boldsymbol{\phi}} \left(\mathbf{r}_i \right) \right) \right], \end{aligned} \tag{6.17}$$

where $\boldsymbol{\phi}$ contains CNN encoder weights $\boldsymbol{\theta}_e$ and linear classifier weights. Besides, y_i denotes the class label (0 or 1) of the i^{th} sample $\mathbf{r}_m$.

In conclusion, the proposed self-supervised spectrum sensing methodology that encompasses both pre-training and fine-tuning procedures is concisely summarized in **Algorithm 4**.

Algorithm 4 DC4S Framework Algorithm Flow

Initialization: The labeled dataset, $\mathcal{D}^L$, the unlabeled dataset $\mathcal{D}^U$, the deep learning network hyperparameters, and model weights $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_p$, $\boldsymbol{\phi}$.

Phase: Pre-training

Input: Received unlabeled OFDM signals.
while iteration falls within the designated epochs, and the stop criterion is not met **do**
    Shuffle the training dataset, initialize parameters $\boldsymbol{\theta}_e$,
    $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_p$, and partition into mini-batches of size B.
    for each mini-batch i **do**
        Apply soft and hard augmentations to obtain $\hat{\mathbf{r}}_i^s$
        and $\hat{\mathbf{r}}_i^h$ for each sample.
        Compute the total loss $\mathcal{L}_{total}$ as per Equation (6.16).
        Update the network modules $\mathcal{E}_{\boldsymbol{\theta}_e}$, $\mathcal{S}_{\boldsymbol{\theta}_s}$ and $\mathcal{P}_{\boldsymbol{\theta}_p}$.
    end for
end while
Save the feature encoder weights $\boldsymbol{\theta}_e$ as the pre-trained model.

Phase: Fine-tuning

Input The labeled dataset $\mathcal{D}^L$ with samples $\mathbf{r}_m$ and their corresponding ground truths y_m.
Shuffle the labeled training dataset, and divide into mini-batches of size B_t.
Import the pre-trained encoder weights from the Pre-training phase.
Integrate a linear classifier following the encoder.
Fine-tune the encoder and linear classifier.
for each mini-batch i **do**
    Calculate the cross-entropy loss $\mathcal{L}_{ce}$, as per Equation (6.17).
    Update the networks $f_{\boldsymbol{\phi}}$.
end for
return Optimal weights $\boldsymbol{\phi}$ of the fine-tuned model.
Evaluate the fine-tuned network by generating P_d versus SNR curves to assess performance and test robustness.

6.3.4 Complexity Analysis

The computational complexity of the DC4S framework is primarily determined by three main components: the CNN encoder, self-attention module, and signal-context contrasting module.

Let c_i, k_i, and l_i represent the number of channels, kernel size, and output length of the i^{th} layer, respectively. The CNN encoder has a computational complexity of $\sum_{i=1}^{L_1} c_{i-1} c_i k_i l_i$, with 26.32M FLOPs and 0.083M parameters. The cross-view prediction module's complexity is variable due to the random selection of

past feature length for prediction. Using expected values, the multi-head self-attention module, including initial linear transformations, exhibits a complexity of $O(\sum_{i=1}^{L_2}(l_i^2 d + d^2 l_i) + \sum_{i=1}^{L_3} n_{i-1} n_i)$, where d represents the hidden dimension, l_i denotes the feature length for the i^{th} layer of the multi-head self-attention block, and n_i represents the number of neurons in the i^{th} dense layer. This module contributes 11.4M FLOPs and 0.731M parameters. The signal-context contrasting module, containing non-linear projection layers, adds a marginal 8,704 FLOPs and 8,672 parameters. The two-branch structure in DC4S framework doubles the FLOPs to 75.44M while maintaining the same parameter count 0.8225M due to weight sharing. In the fine-tuning stage, the CNN encoder combined with the linear classifier results in approximately 26.35M FLOPs and 0.116M parameters.

6.4 Experimental Results and Discussions

In this section, we will present the simulation results and discussion on detection accuracy of the proposed DC4S using a limited number of labeled samples alongside benchmark results. An ablation study is then conducted to evaluate the contributions of individual component within the framework. Subsequently, the transferability of the learned features of the CNN encoder is examined by designing several robustness validation experiments. Besides, hyper-parameter analysis is performed to investigate three parameters, namely, predicted future timesteps J within the cross-view prediction module, weighting factors λ_1 and λ_2 in Equation (6.16). Finally, we explore the selection of appropriate augmentations to optimize our proposed DC4S framework to spectrum sensing problem.

6.4.1 Performance Evaluation

In this study, we assess the detection performance of DC4S framework as well as existing self-supervised spectrum sensing approach CNN-CSL, semi-supervised spectrum sensing approach SSDNN, and supervised learning using a specific percentage of labeled samples. To ensure a fair comparison of each spectrum sensing method's performance, the training datasets and the feature encoder are maintained consistent across all methodologies.

To systematically evaluate the effectiveness of our proposed self-supervised learning approach, we use synthetic data generated under controlled conditions. This allows us to quantitatively assess the model's ability to learn from limited labeled samples while maintaining experimental reproducibility. The dataset employed in this study consists of modulated signals with varying SNRs ranging from -18 dB to 0 dB, with an increment of 2 dB. The modulation scheme used is OFDM-QPSK, which is commonly utilized in modern communication systems. The dataset is divided into three subsets: pre-training, fine-tuning, and evaluation, as detailed in Table 6.2. For each SNR value, 10,000 unlabeled samples are allocated for pre-training, while the number of labeled samples for fine-tuning ranges from 1% to 100% of the predetermined baseline. The evaluation subset comprises 3,000 samples per SNR value, which is used to assess the performance of the well-trained models.

TABLE 6.2: Dataset parameters

Dataset Settings	**Value**
SNR range	-18∼0dB
Modulation	OFDM-QPSK
Pre-training unlabeled samples / SNR	10000
Fine-tuning labeled samples / SNR	500 (1%∼100%)
Evaluation samples / SNR	3000

Initially, we investigate the performance of supervised learning using varying numbers of labeled samples to determine the minimum saturation point, where adding more labeled samples does not significantly improve accuracy. Table 6.3 illustrates the mean detection accuracy and 95% confidence intervals for SNR values ranging from -18 to 0 dB. As the number of labeled samples increases, the mean detection accuracy improves, and the confidence intervals narrow until a saturation point is reached at approximately 500 labeled samples. Beyond this point, no significant improvement in performance is observed. Consequently, we set the number of labeled samples M for the supervised learning benchmark to 500. In the subsequent experiments, all percentages of labeled samples are based on this 500-sample baseline.

Figure 6.4 illustrates the detection accuracy achieved by the proposed DC4S framework, along with several baseline methods, across a wide range of SNR levels from

TABLE 6.3: Supervised learning performance with varying labeled sample sizes

Labeled sample size	**Mean accuracy ± 95% CI*(%)**
2	43.36 ± 8.28
5	61.35 ± 3.20
25	74.71 ± 1.62
100	80.38 ± 0.70
250	81.41 ± 0.42
500	81.99 ± 0.57
1000	82.05 ± 0.53

* **CI** denotes the confidence interval.

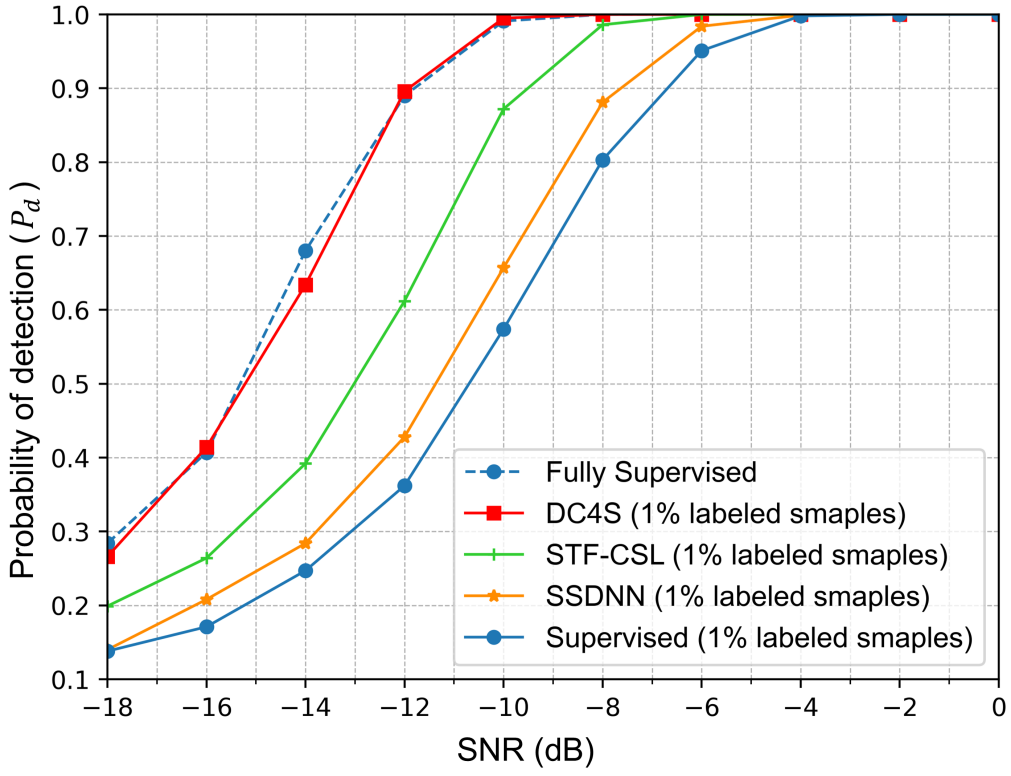


FIGURE 6.4: Detection performance of the proposed DC4S framework with comparisons to other approaches at $P_{fa} = 0.1$.

-18 dB to 0 dB. The DC4S framework, fine-tuned with only 1% labeled samples, demonstrates superior performance compared to the semi-supervised learning approach SSDNN and the self-supervised learning method CNN-CSL, both of which also utilize 1% labeled samples. Notably, the performance of DC4S closely approaches that of the fully supervised learning method using 100% labeled samples,

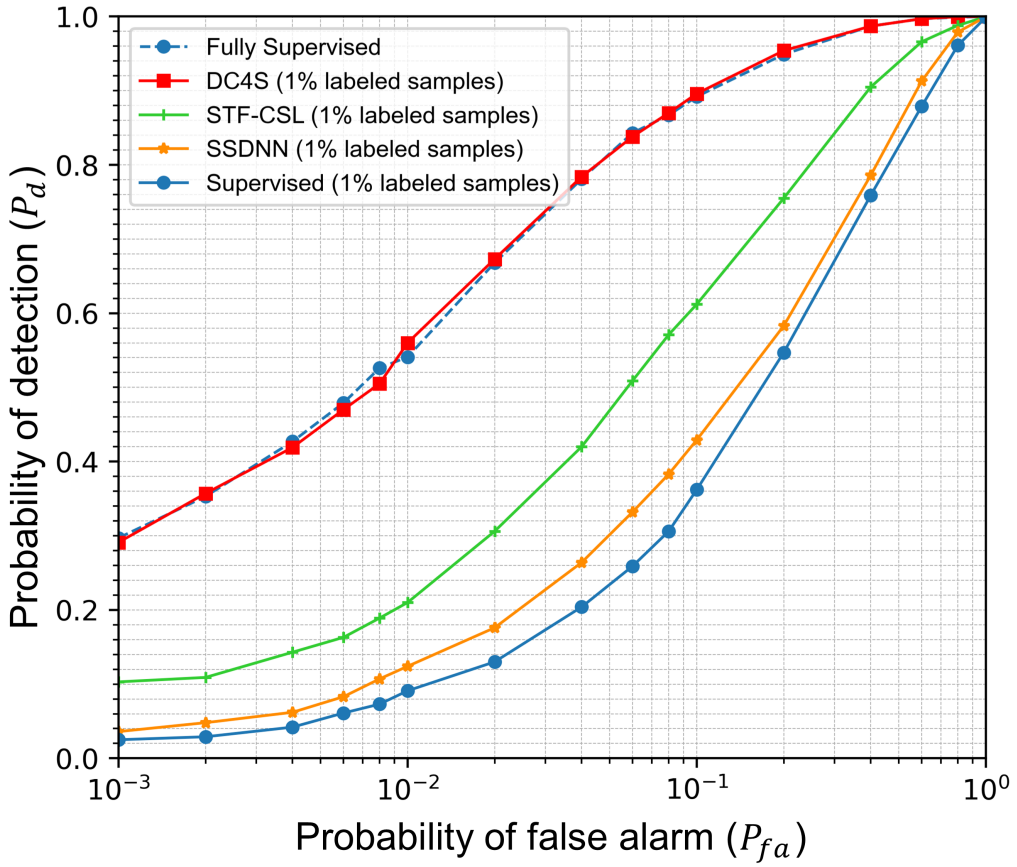


FIGURE 6.5: ROC curves for different approaches at SNR=-12dB.

which demonstrates the effectiveness of the proposed pre-training stage of DC4S framework in learning robust and discriminative features, enabling the model to achieve competitive performance even with limited labeled data. The results also reveal that CNN-CSL outperforms SSDNN, which can be attributed to two factors. Firstly, SSDNN requires multiple training iterations to identify high-confidence pseudo labels. However, due to the simple feature encoder employed in SSDNN, accurately identifying pseudo labels using only the initial 1% labeled samples is challenging. Secondly, the slightly better performance of CNN-CSL may be attributed to its pre-training stage, which employs a novel loss function that is more likely to extract discriminative representations. In scenarios characterized by extremely low SNRs, the proposed DC4S framework with 1% labeled samples achieves nearly 30% in P_d, which provides a significant enhancement, ranging from 10% to 17%, in comparison to CNN-CSL, SSDNN and supervised learning using the same number of labeled samples.

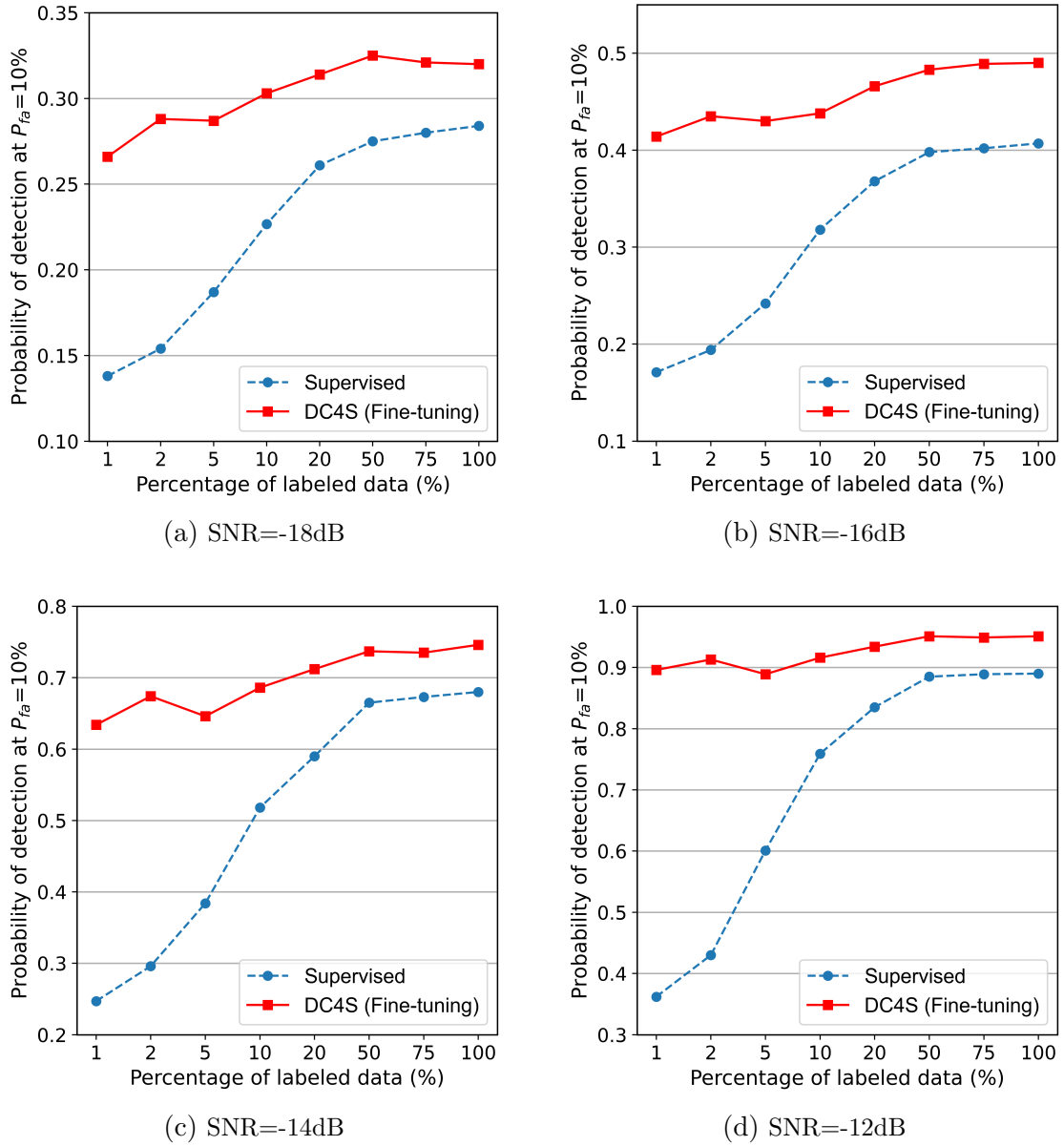
Figure 6.5 presents the ROC curves for several methodologies at an SNR level

of -12dB. It is observed that the DC4S framework consistently achieves a higher P_d for a given P_{fa} compared to the other methods utilizing 1% labeled samples. Specifically, when $P_{fa} = 0.001$, the proposed approach can obtain a P_d around 30%. Conversely, utilizing the same number of labeled samples, CNN-CSL, SSDNN and supervised learning achieve P_d values ranging from 2% to 20%. The ROC curves of DC4S lie closest to the top-left corner of the plot, indicating its superior trade-off between P_d and P_{fa}. Furthermore, the ROC curves of DC4S and supervised learning almost overlap at all SNR levels, indicating that DC4S can achieve comparable detection performance to fully supervised learning while requiring only a limited fraction of the labeled data.

Besides, as illustrated in Figure 6.6, a more detailed comparison is provided between the DC4S framework fine-tuned with different fractions of labeled samples and the fully supervised learning approach using 100% labeled samples at four specific SNR levels. It can be observed that supervised training performs poorly when only a limited amount of labeled data are available, while the DC4S framework, even when fine-tuned with merely 1% labeled data, demonstrates superior performance, achieving 26.6%, 41.4%, 63.4%, and 89.6% across SNR levels from -18 to -12 dB, respectively. This is significantly higher than the accuracy of supervised training, which is 13.8%, 17.1%, 24.7%, and 36.2% under the same conditions. Moreover, the DC4S framework fine-tuned with just 1% to 10% of labeled data can achieve better performance than the supervised model with a complete set of labeled data at different levels of SNR, which highlights its potential for practical deployment in wireless communication systems, where labeled data acquisition can be costly and time-consuming.

6.4.2 Ablation Study

To evaluate the individual contributions of the key components in our DC4S framework, we have conducted a comprehensive ablation study. This analysis focuses on the impact of the cross-view prediction module and the signal-context contrasting module in the pre-training stage. Table 6.4 presents the mean accuracy achieved with 1% labeled samples for different configurations of our model.



(a) SNR=-18dB

(b) SNR=-16dB

(c) SNR=-14dB

(d) SNR=-12dB

FIGURE 6.6: Comparative analysis of supervised training and DC4S fine-tuning under four SNR levels with different proportions of labeled data in terms of P_d.

The DC4S framework, incorporating both the cross-view prediction module and the signal-context contrasting Module, achieves the highest accuracy of 82.04%. This performance serves as our baseline for comparison.

When the signal-context contrasting module is removed, leaving only the CNN encoder and the cross-view prediction module, the accuracy decreases slightly to 80.64%. This relatively small drop in performance suggests that the cross-view prediction module contributes significantly to the model's ability to learn discriminative features. However, when we remove the cross-view prediction module and

TABLE 6.4: Ablation Study on DC4S: Mean Accuracy across SNRs with 1% Labeled Samples

Configuration	Accuracy
DC4S	82.04%
w/o signal-context contrasting module	80.64%
w/o cross-view prediction module	70.77%

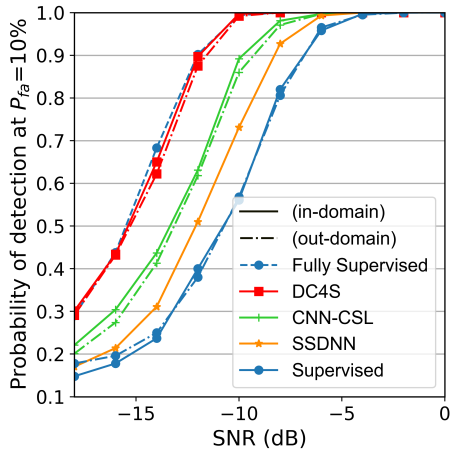
retain only the CNN encoder and the signal-context contrasting module, we observe a more substantial decrease in accuracy to 70.77%.

These results demonstrate the synergistic effect of combining both modules in our DC4S framework. While each module contributes positively to the model's performance, their integration leads to a more robust and effective spectrum sensing approach.

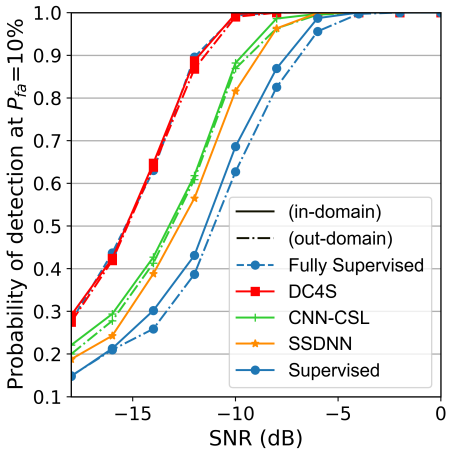
6.4.3 Robustness Validation Experiments

In this section, we investigate the robustness and transferability of the proposed DC4S framework, under various channel conditions, oversampling rates, subcarrier numbers, and modulation types. The performance of DC4S is compared with supervised learning approaches using 100% or 1% labeled samples, self-supervised learning frameworks CNN-CSL, and the semi-supervised learning approach SS-DNN. The solid lines represent in-domain scenarios, where the dataset for pre-training, fine-tuning, and testing is under the same condition, while the dashed lines denote out-domain scenarios, where fine-tuning and testing are performed on a dataset from a different domain.

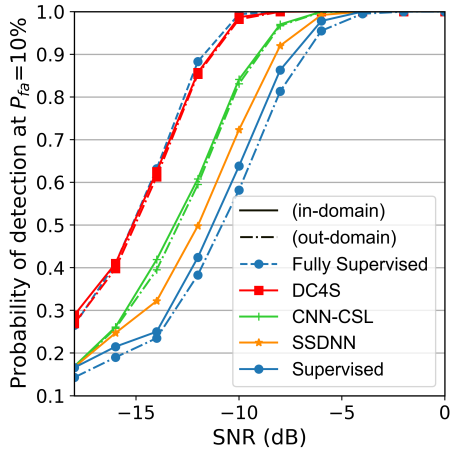
Figures 6.7(a)-(e) illustrate the robustness of the proposed DC4S framework against different channel conditions, namely AWGN channel as well as TDL-A to TDL-D, which differ in their path numbers and magnitudes [128]. The fully supervised learning approach, utilizing 100% labeled samples for training, serves as the baseline. Notably, DC4S exhibits robust performance across various channel conditions when fine-tuned using only 1% of the target condition labeled dataset. The performance of DC4S is nearly equivalent to that achieved by re-training the pre-trained



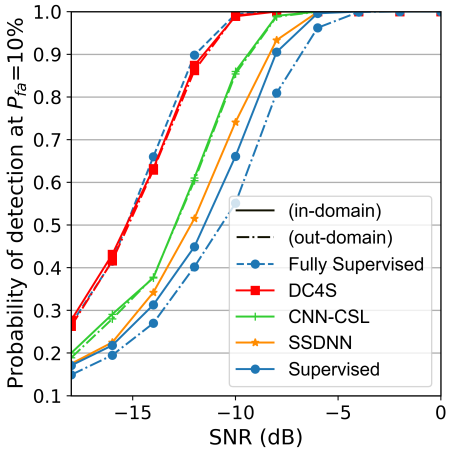
(a) TDL-A channel



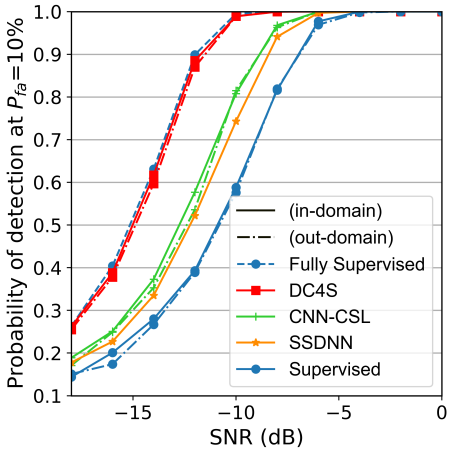
(b) TDL-B channel



(c) TDL-C channel



(d) TDL-D channel



(e) AWGN channel

FIGURE 6.7: Cross-domain robustness validation experiments performed on different channel conditions.

model using a large amount of target condition unlabeled data, highlighting the

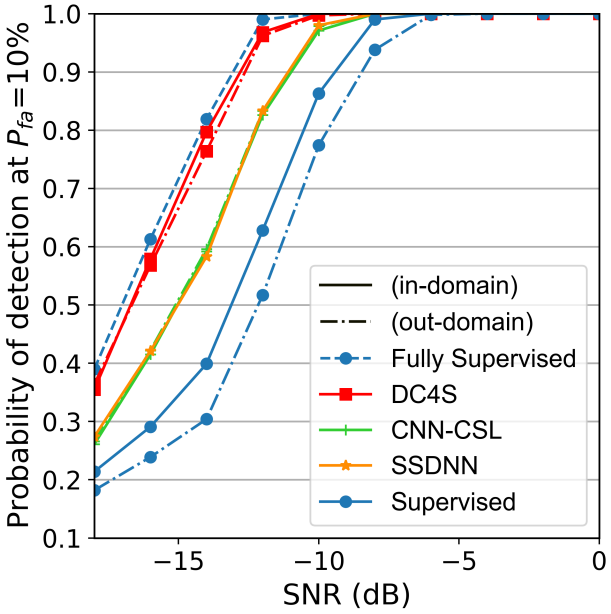


FIGURE 6.8: Cross-domain robustness validation experiments performed on oversampling rate $\zeta = 8$.

discriminability of the extracted features and the efficiency of the proposed framework in terms of time and resource utilization. However, the slight performance gap between DC4S and the fully supervised learning approach suggests that in some cases, more than 1% labeled samples may be required for fine-tuning to achieve optimal performance. Across all channel conditions, DC4S consistently achieves a P_d above 60% at SNR level of -14 dB when fine-tuned with only 1% of the target condition labeled dataset. In comparisons, CNN-CSL and SSDNN, using the same amount of labeled data, achieve a P_d of approximately 40% and 30%∼40%, respectively. Additionally, the supervised learning approach with only 1% labeled samples exhibits the lowest robustness against channel variations among all the schemes. This can be attributed to its inability to extract discriminative features using such a limited amount of labeled data, emphasizing the importance of the pre-training stage in the proposed DC4S framework.

The transferability of DC4S is further validated under different oversampling rates (Figure 6.8), number of subcarriers (Figure 6.9), and modulation types (Figure 6.10). The proposed framework maintains its superior performance across these diverse scenarios, demonstrating its adaptability and generalization capability. The results underscore the effectiveness of the self-supervised pre-training stage in extracting discriminative features, enabling efficient fine-tuning with limited labeled data from

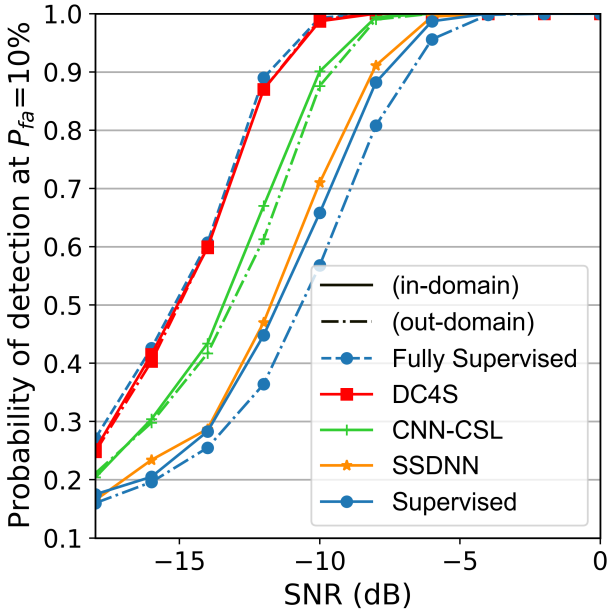


FIGURE 6.9: Cross-domain robustness validation experiments performed on number of subcarriers $N_{sc} = 64$.

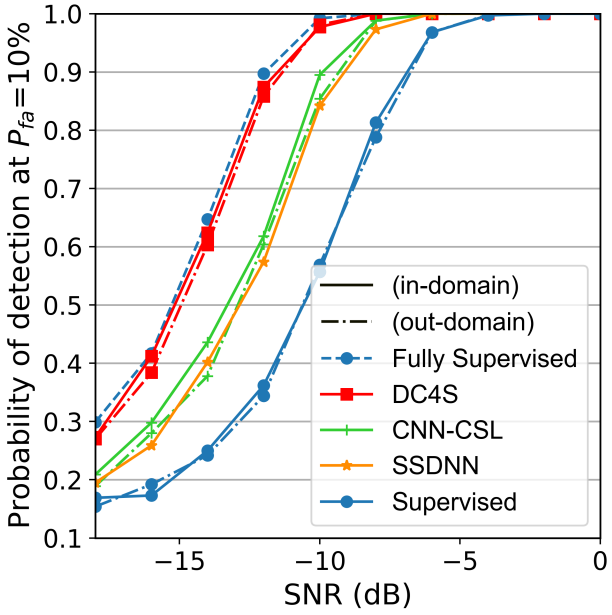


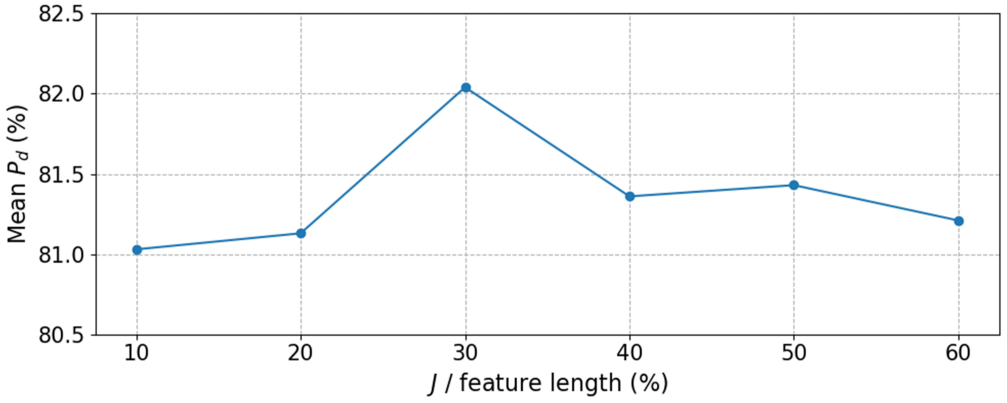
FIGURE 6.10: Cross-domain robustness validation experiments performed on OFDM-16QAM modulation type.

the target domain. It is worth noting that the supervised learning approach with only 1% labeled samples consistently performs the worst among all the compared methods across different channel conditions and system parameters. For instance, in Figure 6.10, supervised learning with 1% labeled samples achieves a P_d of merely 60% at an SNR of -10 dB, which is approximately 40%, 30%, 25% lower than that

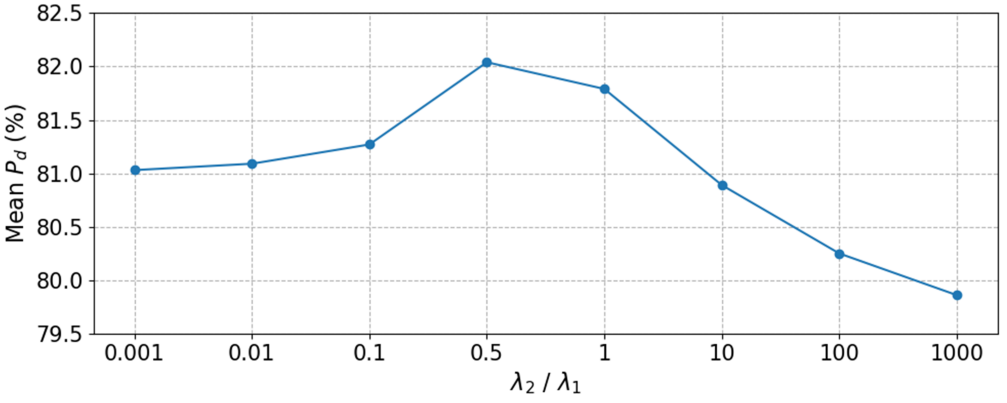
of DC4S, CNN-CSL and SSDNN, respectively. This performance gap remains significant across all the evaluated scenarios, highlighting the limitations of supervised learning when labeled data are scarce.

6.4.4 Hyper-Parameter Analysis

To illustrate the influence of key parameters on the performance of the DC4S framework, a comprehensive hyper-parameter analysis was undertaken, including three critical parameters: predicted future timesteps J within the cross-view prediction module, and the scalar hyperparameters λ_1 and λ_2 in the total loss function (Equation (6.16)). The evaluation experiments are conducted to appraise the performance under both limited and full label scenarios.



(a) Correlation between P_d and predictive timestep J.



(b) Correlation between P_d and the ratio of weighting factors λ_2/λ_1.

FIGURE 6.11: Experiments on hyper-parameter analysis. (a) The influence of altering the percentage of predicted future timesteps J; (b) The impact of adjusting the ratio of two weighting factors λ_1 and λ_2 on mean probability of detection (P_d).

Figure 6.11(a) illustrates the influence of the predictive timestep J on the P_d of DC4S framework, where the x-axis represents the ratio J/feature length, which is produced by the CNN encoder. The results clearly indicate that an increase in the proportion of predicted future timesteps leads to enhanced performance. Nevertheless, it is important to note that excessively large percentages may inversely affect performance due to the significant decrease in historical data available for the cross-view prediction task. Based on our observations, predicting 30% of the feature length yields the optimal detection performance.

Figure 6.11(b) depicts the results of varying λ_2 in the DC4S framework, assuming λ_1 is fixed at 1. The x-axis represents the ratio λ_2/λ_1, and the range of λ_2 values explored spans from 0.001 to 1000. The analysis reveals that the proposed network can perform the best when $\lambda_2 \approx 0.5$. Interestingly, we observe that when $\lambda_2 < 0.1$, the model exhibits less sensitivity to its value compared to the scenario where $\lambda_2 > 1$. This finding suggests that λ_1 plays a more significant role in the spectrum sensing task, and if λ_2 is substantially larger than λ_1, the performance undergoes a noticeable decrease.

6.4.5 Augmentation Selection

The appropriate selection of augmentations is pivotal for the efficacy of contrastive learning approaches due to their susceptibility to the type of augmentations employed [86]. In this section, we will investigate the optimal augmentation techniques for our self-supervised spectrum sensing problem.

To determine the optimal combination of soft and hard augmentations, we conduct a series of experiments using the linear evaluation method. First, we evaluate the performance of each soft augmentation (jittering, scaling) and hard augmentation (timeshift, timewarp, permutation, reverse, FFT) individually. As presented in Table 6.5, the accuracy with only soft augmentation is only approximately 66% when 1% labeled samples are used to finetune. Besides, the accuracy with only hard augmentation is from 79.05% to 80.26%. The results show that **scaling** outperforms **jittering** among the soft augmentations, while **reverse** yields the best performance among the hard augmentations.

TABLE 6.5: Performance of the DC4S framework with 1% and 100% labeled samples using different combinations of soft and hard augmentations.

Soft Augmentation	Hard Augmentation	Acc.(1%)	Acc.(100%)
Jittering	No Augmentation	66.16	83.06
Scaling	No Augmentation	66.29	83.53
No Augmentation	Time shift	79.15	83.99
No Augmentation	Timewarp	80.23	84.07
No Augmentation	Permutation	80.13	84.29
No Augmentation	Reverse	80.26	84.57
No Augmentation	FFT	79.05	83.72
Scaling	Reverse	80.45	84.57
Jittering	Reverse	80.31	84.41
Scaling	Permutation	80.24	84.30
Scaling	Timewarp	80.25	84.24
Scaling	Scaling + Reverse	**82.04**	**85.07**
Scaling	Jittering + Reverse	80.71	84.65
Scaling	Permutation + Reverse	80.41	84.51

Next, we combine the best-performing soft augmentation *scaling* with the best-performing hard augmentation *reverse* and compare the model's performance using this combined strategy against using only **scaling** or only **reverse**. The experimental results demonstrate that the combination of **scaling** and **reverse** significantly outperforms the individual augmentations, achieving 80.45% and 84.57% for 1% and 100% labeled samples, respectively. Due to the similar performance of **scaling** and **jittering** when used alone, we also test the combination of **jittering** as a soft augmentation and **reverse** as a hard augmentation. The results show that the combination of **scaling** and **reverse** performs better than **jittering** and **reverse**.

To further validate our findings, we experiment with different combinations of soft and hard augmentations, such as **scaling** and **timeshift**, **scaling** and **permutation**, **scaling** and **timewarp**, etc. However, none of these combinations surpasses the performance of combination of **scaling** and **reverse**. Notably, **jittering** (i.e., adding random noise) and **scaling** (i.e., adjusting amplitude), can also be considered as hard augmentations depending on the jittering standard deviation and the scaling factor. This observation leads us to explore the possibility of incorporating

scaling or **jittering** as a hard augmentation as well. Hence, as presented in Table 6.5, we test different combinations in hard augmentation such as **scaling** and **reverse**, **jittering** and **reverse**, **permutation** and **reverse**. The results show that using **scaling** and **reverse** as the hard augmentation performs the best, not only elevated the accuracy from 80.45% to 82.04% with 1% labeled dataset but also from 84.57% to 85.07% with the full labeled dataset.

6.5 Conclusion

In this chapter, we have proposed a novel dual-contrast self-supervised spectrum sensing framework, called DC4S, which leverages the distinct yet correlated views generated through soft and hard data augmentations to extract robust representations of communication signals. The effectiveness of the DC4S framework is significantly enhanced by the incorporation of a cross-view prediction module and a signal-context contrasting module, which work together to enable the learning of discriminative features crucial for accurate spectrum sensing.

Our comprehensive investigation of the DC4S framework has yielded valuable insights regarding self-supervised learning in spectrum sensing applications. Through extensive augmentation experiments, we have observed that correctly choosing and combining soft and hard augmentations is crucial for learning effective representations. Specifically, dividing augmentations into soft and hard categories could help maintain a balance between preserving original signal characteristics and introducing sufficient variations. Besides, our analysis of the cross-view prediction module has revealed that the module's effectiveness depends on the prediction length, which requires a balance between historical information utilization and future state prediction. Furthermore, maintaining a balanced ratio between cross-view prediction and signal-context contrasting losses is also essential, where we find that the cross-view prediction loss plays a more significant role in achieving robust performance.

Experimental results have demonstrated the superior performance of DC4S compared to existing self-supervised, semi-supervised, and supervised learning approaches. By fine-tuning both the pre-trained CNN encoder and a linear classifier with only 1% labeled data, we achieve comparable performance to fully supervised

training using 100% labeled data. This highlights the potential of DC4S in reducing the reliance on labeled data, which is often costly and time-consuming to acquire in practical wireless communication systems. Besides, we have conducted comprehensive experiments to validate the transferability and robustness of the DC4S framework under different scenarios such as various channel conditions, oversampling rates, subcarrier numbers, and modulation types. We have also performed a detailed hyper-parameter analysis on three pivotal parameters within the DC4S framework and explored the optimal selection of soft and hard augmentations to further enhance its effectiveness.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

This thesis has explored and advanced the application of deep learning techniques for robust non-cooperative spectrum sensing in cognitive radio networks. The research presented herein addresses several critical challenges in the field, contributing novel methodologies that enhance the accuracy, reliability, and adaptability of spectrum sensing under various challenging conditions.

The work began by proposing a two-stage spectrum sensing method that effectively combines deep learning-based denoising with spectrum sensing. This approach, utilizing a SCAE structure for signal reconstruction, demonstrated significant improvements in detection probability at fixed false alarm rates, particularly in noisy environments. This contribution addresses a fundamental challenge in spectrum sensing: the degradation of performance due to noise, especially in low SNR scenarios.

Building upon this foundation, the thesis then introduced a joint optimization framework for denoising and spectrum sensing in OFDM systems through 5G TDL fading channel. This unified deep learning model represents a significant advancement over traditional two-stage approaches by simultaneously learning to denoise received signals and perform spectrum sensing. The joint optimization not only improved overall sensing performance but also reduced the dependency

on the quality of the denoising stage. Notably, this approach achieved a substantial reduction in computational complexity compared to the previous SCAE-HCSG method, enhancing both efficiency and effectiveness. This offers a more streamlined and powerful solution for cognitive radio networks operating in challenging channel conditions.

Addressing the critical issue of domain shift in deep learning-based spectrum sensing, this research developed an environment-robust deep learning model that extracts channel-invariant features. They enable the model to maintain performance across different channel conditions without extensive retraining. This contribution significantly enhances the practicality and scalability of deep learning-based spectrum sensing in real-world cognitive radio networks, where channel conditions can vary widely and unpredictably.

Finally, recognizing the challenges posed by limited availability of labeled data in many practical scenarios, this thesis proposed a self-supervised pre-training framework called DC4S. This innovative approach leverages unlabeled data to learn meaningful representations of OFDM signals, demonstrating competitive performance compared to fully supervised learning approaches when fine-tuned with limited labeled data. This contribution opens new avenues for spectrum sensing in scenarios where labeled data are scarce, potentially broadening the applicability of deep learning techniques in this domain.

The comprehensive body of work presented in this thesis represents a significant step forward in the field of spectrum sensing for cognitive radio networks. By addressing key challenges such as noise mitigation, domain shift, and limited labeled data, this research enhances the robustness, adaptability, and practical applicability of deep learning-based spectrum sensing techniques.

7.2 Future Works

While this thesis has made significant strides in advancing deep learning techniques for non-cooperative spectrum sensing in cognitive radio networks, several promising avenues for future research emerge. These directions not only build upon the foundations laid by this work but also aim to address emerging challenges and opportunities in the evolving landscape of wireless communications.

Firstly, we will extend our DL-based spectrum sensing techniques on real-world datasets. While this thesis primarily utilizes simulated data for proof-of-concept validation, the use of real-world datasets would enable a more comprehensive evaluation of the proposed methods under practical conditions. Incorporating real-world data would help bridge the gap between theoretical evaluation and practical deployment, ensuring that the models are robust and reliable in diverse and unpredictable scenarios. Future work could involve the collection of real-world datasets, leveraging hardware platforms to capture spectrum data across varying environments, and integrating these datasets into the training and validation pipelines.

Besides, one of the most promising directions for future research is the extension of the developed techniques to ultra-wideband signal spectrum sensing [129, 130]. While this thesis has made significant strides in spectrum sensing, the ever-increasing demand for high-bandwidth applications and the trend towards wider bandwidth allocations in modern wireless systems necessitate further advancements in this area. Ultra-wideband sensing presents unique challenges, such as the need to process an immense amount of data in real-time and managing substantial computational complexity when dealing with signals that span a very wide frequency range. Future research could explore the adaptation of the deep learning architectures and methodologies developed in this thesis to handle ultra-wideband signals effectively. This might involve developing novel neural network architectures capable of efficiently processing ultra-wideband data, investigating sub-Nyquist sampling techniques in conjunction with deep learning to reduce data acquisition requirements, and exploring distributed learning approaches to manage the increased computational load of ultra-wideband sensing. Moreover, the application of advanced signal processing techniques such as compressive sensing, in combination with the deep learning methods proposed in this thesis, could potentially lead to more efficient ultra-wideband spectrum sensing solutions [131]. Such research would not only advance the field of cognitive radio networks but also contribute to the broader area of dynamic spectrum access in next-generation wireless systems.

Another intriguing direction for future work is the integration of the developed spectrum sensing techniques with specific applications, particularly in the domain of unmanned aerial vehicles (UAVs) [132–134]. The use of UAVs in wireless communications has been growing rapidly, with applications ranging from temporary network deployment in emergency situations to providing coverage in hard-to-reach

areas. Integrating advanced spectrum sensing capabilities into UAV-based communication systems presents both exciting opportunities and unique challenges. In the context of UAV applications, future research could explore how the deep learning-based spectrum sensing techniques developed in this thesis can be adapted to the dynamic and mobile nature of UAV environments. This might involve investigating the impact of UAV mobility on sensing performance, developing lightweight implementations of the proposed algorithms suitable for the limited computational resources of UAVs, and exploring how the three-dimensional nature of UAV deployments affects spectrum sensing strategies. While the idea of integrating advanced spectrum sensing with UAV applications is promising, it is important to acknowledge the potential challenges. These may include power constraints of UAVs, the need for real-time processing and decision-making, and the complexities introduced by the mobility and altitude variations of UAVs. However, given the increasing importance of UAVs in wireless communications and the potential benefits of cognitive radio capabilities in these systems, this research direction appears both practical and valuable.

An additional crucial direction for future research is the development of robust defense methodologies against potential attacks on deep learning-based spectrum sensing networks [135–139]. As the reliance on deep learning techniques for spectrum sensing grows, so does the vulnerability to adversarial attacks that could compromise the integrity and reliability of sensing decisions [140]. Malicious actors could potentially exploit weaknesses in the deep learning models to manipulate sensing outcomes, leading to false detections or missed opportunities for spectrum access. This could significantly degrade the efficiency and reliability of cognitive radio networks. Future work should focus on identifying potential attack vectors specific to spectrum sensing networks, such as data poisoning attacks during model training, evasion attacks during inference, or even attacks exploiting the physical layer characteristics of wireless signals. Research efforts could explore the development of robust deep learning architectures that are inherently resistant to adversarial perturbations, investigate the application of defensive distillation techniques to spectrum sensing models, or design novel detection mechanisms for identifying and mitigating attacks in real-time. By addressing these security challenges, future research can ensure that the advances made in deep learning-based spectrum sensing can be safely and reliably deployed in real-world cognitive radio networks, maintaining their performance even in the face of potential adversarial threats.

In conclusion, this thesis has laid a solid foundation for the application of deep learning techniques in non-cooperative spectrum sensing for cognitive radio networks. The future research directions outlined represent promising avenues to further enhance the efficiency, adaptability, and security of spectrum sensing systems. As wireless communication technologies continue to evolve, these advanced spectrum sensing techniques will play a crucial role in enabling more efficient spectrum utilization and supporting the growing demands of next-generation wireless networks. By pursuing these research directions, we can continue to push the boundaries of what is possible in cognitive radio networks, ultimately contributing to the development of more intelligent, secure, and efficient wireless communication systems.

List of Author's Publications[1]

- **Zhengyang Su**, Kah Chan Teh, Sirajudeen Gulam Razul, and Alex Chichung Kot, "Deep non-cooperative spectrum sensing over Rayleigh fading channel," IEEE Transactions on Vehicular Technology, vol. 71, no. 4, pp. 4460-4464, Apr. 2022.

- **Zhengyang Su**, Kah Chan Teh, Sirajudeen Gulam Razul, and Alex Chichung Kot, "Robust channel invariant deep non-cooperative spectrum sensing," IEEE Wireless Communications Letters, vol. 12, no. 3, pp. 436-440, Mar. 2023.

- **Zhengyang Su**, Kah Chan Teh, Yihang Xie, Sirajudeen Gulam Razul, Alex Chichung Kot, "Signal enhancement aided end-to-end deep learning approach for joint denoising and spectrum sensing," IEEE Transactions on Vehicular Technology, vol. 73, no.3, pp. 4424 - 4428, Mar. 2024.

- **Zhengyang Su**, Kah Chan Teh, Yihang Xie, Sirajudeen Gulam Razul, Alex Chichung Kot, "DC4S: A dual-contrast self-supervised learning framework for robust and efficient spectrum sensing with limited labeled data," IEEE Transactions on Cognitive Communications and Networking (Early Access), DOI: 10.1109/TCCN.2025.3544400.

- Bing Ren*, **Zhengyang Su***, Kah Chan Teh, Hongyang An, Erry Gunawan, Alex Chichung Kot, "CPAA: self-supervised cross-view prediction with automatic augmentation for OFDM modulation classification," IEEE Transactions on Vehicular Technology (Early Access), DOI: 10.1109/TVT.2025.3543800.

- Satoshi Tsutsui, **Zhengyang Su**, and Bihan Wen, "Benchmarking white blood cell classification under domain shift," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5.

[1]The superscript * indicates joint first authors

- Yu Hai, **Zhengyang Su**, Junjie Wu, Kah Chan Teh, Zhaoyi Shao, Wei Pu, Ruoming Li, Yulin Huang, and Jianyu Yang, "Deep spectral sensing and reconstruction for high-resolution imaging of MWP-SAR in complex electromagnetic environments," IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp. 1-15, Jun. 2024, DOI: 10.1109/TGRS.2024.3409892.

- Yihang Xie, Xiaobei Liu, **Zhengyang Su**, Kah Chan Teh, Yong Liang Guan, and Chaosan Yang, "Ensemble learning aided large communication model for multi-scenario non-linear distortion," IEEE Transactions on Cognitive Communications and Networking (Early Access), DOI: 10.1109/TCCN.2024.3524186.

Bibliography

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.

[2] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, 2005.

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surv. Tutor.*, vol. 11, no. 1, pp. 116–130, 2009.

[4] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: Requirements, challenges and design trade-offs," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 32–39, 2008.

[5] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrum sensing for cognitive radio: Challenges and solutions," *EURASIP J. Adv. Signal Process.*, vol. 2010, pp. 1–15, 2010.

[6] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proc. IEEE*, vol. 55, no. 4, pp. 523–531, 1967.

[7] M. Yang, Y. Li, X. Liu, and W. Tang, "Cyclostationary feature detection based spectrum sensing algorithm under complicated electromagnetic environment in cognitive radio networks," *China Commun.*, vol. 12, no. 9, pp. 35–44, 2015.

[8] J. Lorincz, I. Ramljak, and D. Begušić, "A review of the noise uncertainty impact on energy detection with different OFDM system designs," *Comput. Commun.*, vol. 148, pp. 185–207, 2019.

[9] W.-L. Chin, "On the noise uncertainty for the energy detection of OFDM signals," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7593–7602, 2019.

[10] P. Aparna and M. Jayasheela, "Cyclostationary feature detection in cognitive radio using different modulation schemes," *Int. J. Comput. Appl.*, vol. 47, no. 21, 2012.

[11] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2209–2221, 2013.

[12] Z. Song, Y. Gao, and R. Tafazolli, "A survey on spectrum sensing and learning technologies for 6G," *IEICE Trans. Commun.*, vol. 104, no. 10, pp. 1207–1216, 2021.

[13] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, 2019.

[14] N. Balwani, D. K. Patel, B. Soni, and M. López-Benítez, "Long short-term memory based spectrum sensing scheme for cognitive radio," in *Proc. IEEE 30th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.* IEEE, 2019, pp. 1–6.

[15] Y. Wang, L. Tu, J. Guo, and Z. Wang, "Residual learning based RF signal denoising," in *2018 IEEE International Conference on Applied System Invention (ICASI).* IEEE, 2018, pp. 15–18.

[16] E. Almazrouei, G. Gianini, N. Almoosa, and E. Damiani, "A deep learning approach to radio signal denoising," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW).* IEEE, 2019, pp. 1–8.

[17] H. Abouzid, O. Chakkor, O. G. Reyes, and S. Ventura, "Signal speech reconstruction and noise removal using convolutional denoising audioencoders with neural deep learning," *Analog Integr. Circuits Process.*, vol. 100, pp. 501–512, 2019.

[18] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proc. IEEE*, vol. 97, no. 5, pp. 894–914, 2009.

[19] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.

[20] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Signal Process. Mag.*, vol. 29, no. 3, pp. 101–116, 2012.

[21] R. Tandra and A. Sahai, "Fundamental limits on detection in low SNR under noise uncertainty," in *Proc. Int. Conf. Wireless Netw. Commun. Mobile Comput.*, vol. 1, 2005, pp. 464–469.

[22] P. B. Gohain, S. Chaudhari, and V. Koivunen, "Cooperative energy detection with heterogeneous sensors under noise uncertainty: SNR wall and use of evidence theory," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 473–485, 2018.

[23] Y. Zeng, C. L. Koh, and Y.-C. Liang, "Maximum eigenvalue detection: Theory and application," *Proc. IEEE Int. Conf. Commun.*, pp. 4160–4164, 2008.

[24] Z. Tian and G. B. Giannakis, "A wavelet approach to wideband spectrum sensing for cognitive radios," in *Proc. Int. Conf. Cognit. Radio Oriented Wireless Netw. Commun. (CROWNCOM)*. IEEE, 2006, pp. 1–5.

[25] K. Hassan, R. Gautier, I. Dayoub, M. Berbineau, and E. Radoi, "Multiple-antenna-based blind spectrum sensing in the presence of impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2248–2257, 2013.

[26] K. Sherbin and V. Sindhu, "Cyclostationary feature detection for spectrum sensing in cognitive radio network," in *Proc. Int. Conf. Intell. Comput. Control Syst.* IEEE, 2019, pp. 1250–1254.

[27] F. Benedetto, G. Giunta, and L. Pallotta, "Cognitive satellite communications spectrum sensing based on higher order moments," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 574–578, 2020.

[28] F. Chiti, R. Fantacci, D. Marabissi, and A. Tani, "Correlation-based spectrum sensing with oversampling and optimal weights selection for OFDM-based networks coexistence in TVWS," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 511–523, 2020.

[29] Y. Wang, W. Xu, F. Wang, X. Qin, N. Ma, and M. Pan, "Robust spectrum sensing based on phase difference distribution," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 1, pp. 28–42, 2022.

[30] Y. Zeng, Y.-C. Liang, and R. Zhang, "Blindly combined energy detection for spectrum sensing in cognitive radio," *IEEE Signal Process. Lett.*, vol. 15, pp. 649–652, 2008.

[31] Y. Zeng and Y.-C. Liang, "Spectrum-sensing algorithms for cognitive radio based on statistical covariances," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1804–1815, 2008.

[32] F. Salahdine, H. El Ghazi, N. Kaabouch, and W. F. Fihri, "Matched filter detection with dynamic threshold for cognitive radio networks," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*. IEEE, 2015, pp. 1–6.

[33] M. R. Vyas, D. K. Patel, and M. Lopez-Benitez, "Artificial neural network based hybrid spectrum sensing scheme for cognitive radio," *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, pp. 1–7, 2017.

[34] Y. Wang, W. Xu, Z. Qin, Y. Zhang, H. Gao, M. Pan, and J. Lin, "Deep neural network-based robust spectrum sensing: Exploiting phase difference distribution," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2021, pp. 1–7.

[35] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, 2019.

[36] J. Xie, C. Liu, Y.-C. Liang, and J. Fang, "Activity pattern aware spectrum sensing: A CNN-based deep learning approach," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1025–1028, 2019.

[37] K. Tekbıyık, Ö. Akbunar, A. R. Ekti, A. Görçin, G. K. Kurt, and K. A. Qaraqe, "Spectrum sensing and signal identification with deep learning based on spectral correlation function," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 514–10 527, 2021.

[38] Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, "Deep STFT-CNN for spectrum sensing in cognitive radio," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 864–868, 2020.

[39] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Commun.*, vol. 17, no. 2, pp. 138–148, 2020.

[40] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2196–2200, 2020.

[41] S. Goyal, P. Bedi, J. Kumar, and V. Varadarajan, "Deep learning application for sensing available spectrum for cognitive radio: An ECRNN approach," *Peer Peer Netw. Appl.*, vol. 14, no. 5, pp. 3235–3249, 2021.

[42] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, 2019.

[43] S. Solanki, V. Dehalwar, and J. Choudhary, "Deep learning for spectrum sensing in cognitive radio," *Symmetry*, vol. 13, no. 1, p. 147, 2021.

[44] K. Yang, Z. Huang, X. Wang, and X. Li, "A blind spectrum sensing method based on deep learning," *Sensors*, vol. 19, no. 10, p. 2270, 2019.

[45] H. Liu, X. Zhu, and T. Fujii, "Ensemble deep learning based cooperative spectrum sensing with stacking fusion center," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*. IEEE, 2018, pp. 1841–1846.

[46] J. Tian, P. Cheng, Z. Chen, M. Li, H. Hu, Y. Li, and B. Vucetic, "A machine learning-enabled spectrum sensing method for OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 374–11 378, 2019.

[47] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Deep learning network based spectrum sensing methods for OFDM systems," *arXiv preprint arXiv:1807.09414*, 2018.

[48] Q. Cheng, Z. Shi, and J. Yuan, "Spectrum sensing in full-duplex OFDM systems using one-shot learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2021, pp. 1–6.

[49] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Sensing OFDM signal: A deep learning approach," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7785–7798, 2019.

[50] G. Pan, J. Li, and F. Lin, "A cognitive radio spectrum sensing method for an OFDM signal based on deep learning and cycle spectrum," *Int. J. Digit. Multimed. Broadcast.*, vol. 2020, pp. 1–10, 2020.

[51] R. Liu, Y. Ma, X. Zhang, and Y. Gao, "Deep learning-based spectrum sensing in space-air-ground integrated networks," *J. Commun. Inf. Netw.*, vol. 6, no. 1, pp. 82–90, 2021.

[52] X. Ding, T. Ni, Y. Zou, and G. Zhang, "Deep learning for satellites based spectrum sensing systems: A low computational complexity perspective," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1366–1371, 2022.

[53] W. Wang and J. Peng, "Cooperative spectrum sensing algorithm for UAV based on deep learning," in *Veh. Technol. Conf. (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[54] Y. Mo, J. Huang, and G. Qian, "Deep learning approach to UAV detection and classification by using compressively sensed RF signal," *Sensors*, vol. 22, no. 8, p. 3072, 2022.

[55] B. Shang, L. Liu, R. M. Rao, V. Marojevic, and J. H. Reed, "3D spectrum sharing for hybrid D2D and UAV networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5375–5389, 2020.

[56] B. Hamdaoui, B. Khalfi, and M. Guizani, "Compressed wideband spectrum sensing: Concept, challenges, and enablers," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 136–141, 2018.

[57] S. Chandhok, H. Joshi, A. V. Subramanyam, and S. J. Darak, "Sensenet: Deep learning based wideband spectrum sensing and modulation classification network," *arXiv preprint arXiv:1912.05255*, 2019.

[58] X. Meng, H. Inaltekin, and B. Krongold, "End-to-end deep learning-based compressive spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2020, pp. 1–6.

[59] B. Khalfi, A. Zaid, and B. Hamdaoui, "When machine learning meets compressive sampling for wideband spectrum sensing," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*. IEEE, 2017, pp. 1120–1125.

[60] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "Deepsense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *Proc. IEEE Conf. Comput. Commun.* IEEE, 2021, pp. 1–10.

[61] R. Mei and Z. Wang, "Deep learning-based wideband spectrum sensing: A low computational complexity approach," *IEEE Commun. Lett.*, 2023.

[62] Z. Su, K. C. Teh, S. G. Razul, and A. C. Kot, "Deep non-cooperative spectrum sensing over rayleigh fading channel," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4460–4464, 2021.

[63] Z. Su, K. C. Teh, Y. Xie, S. G. Razul, and A. C. Kot, "Signal enhancement aided end-to-end deep learning approach for joint denoising and spectrum sensing," *IEEE Trans. Veh. Technol.*, 2023.

[64] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[65] C. Liu, Z. Wei, D. W. K. Ng, J. Yuan, and Y.-C. Liang, "Deep transfer learning for signal detection in ambient backscatter communications," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 3, pp. 1624–1638, 2020.

[66] B. M. Pati, M. Kaneko, and A. Taparugssanagorn, "A deep convolutional neural network based transfer learning method for non-cooperative spectrum sensing," *IEEE Access*, vol. 8, pp. 164 529–164 545, 2020.

[67] J. Miao, Y. Li, X. Jing, F. Zhang, and J. Mu, "Spectrum sensing based on adversarial transfer learning," *IET Commun.*, vol. 16, no. 17, pp. 2059–2069, 2022.

[68] C. Wang, Y. Xu, Z. Chen, J. Tian, P. Cheng, and M. Li, "Adversarial learning-based spectrum sensing in cognitive radio," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 3, pp. 498–502, 2021.

[69] L. Li, H. Jiang, and H. He, "Deep transfer cooperative sensing in cognitive radio," *IEEE Wirel. Commun. Lett.*, vol. 10, no. 6, pp. 1354–1358, 2021.

[70] K. Bu, Y. He, X. Jing, and J. Han, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 880–884, 2020.

[71] A. Bassou, S. A. Chouakri, N. Mellah, M. Khelifi *et al.*, "Spectrum sensing and modulation recognition using a novel CNN deep learning model and learning transfer technique." *Prz. Elektrotechniczny*, vol. 99, no. 5, 2023.

[72] S. Solanki, V. Dehalwar, J. Choudhary, M. L. Kolhe, and K. Ogura, "Spectrum sensing in cognitive radio using CNN-RNN and transfer learning," *IEEE Access*, vol. 10, pp. 113 482–113 492, 2022.

[73] Z. Su, K. C. Teh, S. G. Razul, and A. C. Kot, "Robust channel invariant deep noncooperative spectrum sensing," *IEEE Wirel. Commun. Lett.*, vol. 12, no. 3, pp. 436–440, 2022.

[74] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2018, pp. 1–6.

[75] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5307–5319, 2020.

[76] N. A. Khalek and W. Hamouda, "Unsupervised two-stage learning framework for cooperative spectrum sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2021, pp. 1–6.

[77] G. C. Sobabe, Y. Song, X. Bai, and B. Guo, "A cooperative spectrum sensing algorithm based on unsupervised learning," in *Proc. 10th Int. Congr. Image Signal Proc. BioMed. Eng. Inf.* IEEE, 2017, pp. 1–6.

[78] R. Yelalwar and Y. Ravinder, "Unsupervised machine learning schemes for cooperative spectrum sensing in cognitive radio," *Int. J. Comput. Appl. Technol.*, vol. 73, no. 1, pp. 66–78, 2023.

[79] N. A. Khalek and W. Hamouda, "Deepsense: An unsupervised deep clustering approach for cooperative spectrum sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2023, pp. 1868–1873.

[80] N. A. Khalek and W. Hamouda, "Intelligent spectrum sensing: An unsupervised learning approach based on dimensionality reduction," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2022, pp. 171–176.

[81] N. A. Khalek and W. Hamouda, "Deap learning: A data-driven approach to unsupervised cooperative spectrum sensing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*. IEEE, 2023, pp. 6389–6394.

[82] Y. Zhang and Z. Zhao, "Limited data spectrum sensing based on semi-supervised deep neural network," *IEEE Access*, vol. 9, pp. 166 423–166 435, 2021.

[83] R. Hu, Z. Ye, B. Chen, O. van Kaick, and H. Huang, "Self-supervised color-concept association via image colorization," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 1, pp. 247–256, 2022.

[84] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 6707–6717.

[85] Z. Feng, C. Xu, and D. Tao, "Self-supervised representation learning by rotation feature decoupling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 10 364–10 374.

[86] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *Proc. Int. Conf. Mach. Learn.*, pp. 1597–1607, 2020.

[87] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Ieee, 2009, pp. 248–255.

[88] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[89] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[90] X. Li, Z. Zhao, Y. Zhang, S. Zheng, and S. Dai, "Spectrum sensing algorithm based on self-supervised contrast learning," *Electronics*, vol. 12, no. 6, p. 1317, 2023.

[91] D. Liu, T. Wang, S. Liu, R. Wang, S. Yao, and T. Abdelzaher, "Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective," in *Proc. IEEE Int. Conf. Comput. Commun. Netw. (ICCCN)*. IEEE, 2021, pp. 1–10.

[92] S. Yao, A. Piao, W. Jiang, Y. Zhao, H. Shao, S. Liu, D. Liu, J. Li, T. Wang, S. Hu *et al.*, "STFNets: Learning sensing signals from the time-frequency perspective with short-time Fourier neural networks," in *Proc. World Wide Web Conf.*, 2019, pp. 2192–2202.

[93] B. Ren, Z. Su, K. C. Teh, H. An, E. Gunawan, and A. C. Kot, "Cpaa: Self-supervised cross-view prediction with automatic augmentation for ofdm modulation classification," *IEEE Trans. Veh. Technol.*, pp. 1–15, 2025.

[94] Z. Su, K. C. Teh, Y. Xie, S. G. Razul, and A. C. Kot, "DC4S: A dual-contrast self-supervised learning framework for robust and efficient spectrum sensing with limited labeled data," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2025.

[95] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, 2018.

[96] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 760–10 772, 2018.

[97] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 1, pp. 114–117, 2017.

[98] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, 2019.

[99] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.

[100] X. Yi and C. Zhong, "Deep learning for joint channel estimation and signal detection in ofdm systems," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2780–2784, 2020.

[101] S. Atapattu, C. Tellambura, and H. Jiang, *Energy detection for spectrum sensing in cognitive radio.* New York, NY, USA: Springer, Feb. 2014.

[102] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.* Springer, 2011, pp. 52–59.

[103] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[104] T. J. O'shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. 6th GNU Radio Conf.*, vol. 1, no. 1, 2016.

[105] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 5353–5360.

[106] E. Nurvitadhi, J. Sim, D. Sheffield, A. Mishra, S. Krishnan, and D. Marr, "Accelerating recurrent neural networks in analytics servers: Comparison of FPGA, CPU, GPU, and ASIC," in *Proc. 26th Int. Conf. Field Programmable Logic Appl. (FPL).* IEEE, 2016, pp. 1–4.

[107] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[108] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.* IEEE, 2010, pp. 2366–2369.

[109] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study," *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, 2019.

[110] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 4, pp. 1326–1337, 2008.

[111] N. An, C. Zou, C. Zhang, C. Pan, F. Yang, and J. Song, "Spectrum sensing for DTMB system: A CNN approach," *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 271–278, 2021.

[112] S. Chaudhari, V. Koivunen, and H. V. Poor, "Autocorrelation-based decentralized sequential detection of OFDM signals in cognitive radios," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2690–2700, 2009.

[113] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP Specification Series, Tech. Rep. TR 38.901, 2018, [Online]. Available: http://www.3gpp.org/DynaReport/38901.htm.

[114] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.* Springer, 2015, pp. 234–241.

[115] R. Yamamoto, E. Song, and J.-M. Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *arXiv preprint arXiv:1904.04472*, 2019.

[116] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 94–98, 2018.

[117] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* IEEE, 2020, pp. 6199–6203.

[118] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception," 3GPP, Tech. Specif. TS 36.101, 2022. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2411

[119] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," 3GPP, Tech. Specif. TS 36.104, 2022. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2412

[120] E. Perenda, S. Rajendran, G. Bovet, S. Pollin, and M. Zheleva, "Learning the unknown: Improving modulation classification performance in unseen scenarios," in *Proc. IEEE Conf. Comput. Commun.* IEEE, 2021, pp. 1–10.

[121] C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo, "Exploring contrastive learning in human activity recognition for healthcare," *arXiv preprint arXiv:2011.11542*, 2020.

[122] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[123] J. Pöppelbaum, G. S. Chadha, and A. Schwung, "Contrastive learning based self-supervised time-series analysis," *Appl. Soft. Comput.*, vol. 117, p. 108397, 2022.

[124] B. Liu, Z. Zhang, and R. Cui, "Efficient time series augmentation methods," *Proc. Int. Congr. Image Signal Process. Biomed. Eng. Informat.*, pp. 1004–1009, 2020.

[125] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[126] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," *arXiv preprint arXiv:1906.01787*, 2019.

[127] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[128] Y. Xie, K. C. Teh, and A. C. Kot, "Comm-transformer: A robust deep learning based receiver for OFDM system under TDL channel," *IEEE Trans. Commun.*, vol. 72, no. 4, pp. 2014–2026, 2024.

[129] Z. Luo, Q. Huang, X. Chen, R. Wang, F. Wu, G. Chen, and Q. Zhang, "Spectrum sensing everywhere: Wide-band spectrum sensing with low-cost UWB nodes," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2112–2127, 2024.

[130] X. Li, Z. Hu, C. Shen, H. Wu, and Y. Zhao, "TFF_aDCNN: A pre-trained base model for intelligent wideband spectrum sensing," *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 12 912–12 926, 2023.

[131] R. Mei and Z. Wang, "Compressed spectrum sensing of sparse wideband signals based on deep learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 8434–8444, 2024.

[132] N. Gul, S. M. Kim, J. Ali, and J. Kim, "UAV aided virtual cooperative spectrum sensing for cognitive radio networks," *PLoS One*, vol. 18, no. 9, p. e0291077, 2023.

[133] J. Wu, P. Li, J. Bao, and H. Ge, "Quick multiband spectrum sensing for delay-constraint cognitive UAV networks," *IEEE Sens. J.*, vol. 22, no. 19, pp. 19 088–19 100, 2022.

[134] Z. Luo, J. Xiong, F. Luo, X. Wang, C. Shan, S. Jin, and D. Liu, "UAV spectrum sensing of normalized spectrum based on GRU network," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*. IEEE, 2021, pp. 328–333.

[135] S. Shrivastava, A. Rajesh, P. K. Bora, B. Chen, M. Dai, X. Lin, and H. Wang, "A survey on security issues in cognitive radio based cooperative sensing," *IET Commun.*, vol. 15, no. 7, pp. 875–905, 2021.

[136] R. Biswas, J. Wu, X. Du, and Y. Yang, "Mitigation of the spectrum sensing data falsifying attack in cognitive radio networks," *Cyber- Phys. Syst.*, vol. 7, no. 3, pp. 159–178, 2021.

[137] Z. Luo, S. Zhao, Z. Lu, J. Xu, and Y. E. Sagduyu, "When attackers meet AI: Learning-empowered attacks in cooperative spectrum sensing," *IEEE. Trans. Mob. Comput.*, vol. 21, no. 5, pp. 1892–1908, 2020.

[138] K. Yadav, S. D. Roy, and S. Kundu, "Defense against spectrum sensing data falsification attacker in cognitive radio networks," *Wirel. Pers. Commun.*, vol. 112, no. 2, pp. 849–862, 2020.

[139] N. Marriwala, H. Punj, S. Panda, I. Kaur, and D. Rathore, "An authentication based approach for prevention of spectrum sensing data falsification attacks in cognitive radio network," *Wirel. Pers. Commun.*, vol. 124, no. 1, pp. 119–145, 2022.

[140] S. Zheng, L. Ye, X. Wang, J. Chen, H. Zhou, C. Lou, Z. Zhao, and X. Yang, "Primary user adversarial attacks on deep learning-based spectrum sensing and the defense method," *China Commun.*, vol. 18, no. 12, pp. 94–107, 2021.