Robust Representation Learning Solutions for Wireless Sensing Applications

Borna Barahimi

A thesis submitted to the
Department of Electrical Engineering and Computer Science
in conformity with the requirements for
the degree of Master of Science

YORK UNIVERSITY
TORONTO, ONTARIO

August 2024

© Borna Barahimi, 2024

Abstract

WiFi sensing, a technique for utilizing wireless signals for monitoring human activities and environmental conditions, holds substantial potential in diverse applications including human activity recognition (HAR). It offers a powerful, continuous, and non-intrusive monitoring solution. This technology eliminates the need for wearable sensors, and even functions outside the line-of-sight. However, the large-scale deployment of WiFi sensing faces several challenges: (1) limited computational power in WiFi devices, (2) the cost and complexity of annotating channel state information (CSI) data, and (3) ensuring model generalization across different environments.

The first part of the thesis addresses the limited computation power of edge devices by developing a Real-time Sensing and Compression Network (RSCNet). RSCNet is a cloud-based architecture designed to alleviate computational constraints on edge devices. It achieves this through efficient CSI compression at the edge and subsequent sensing and reconstruction in the cloud. RSCNet employs window-based CSI compression and LSTM-based recurrent blocks, significantly reducing computational demands and communication overheads while maintaining high sensing accuracy.

The second part of the thesis addresses the issue of limited labeled data by developing self-supervised learning (SSL) method, namely Context-Aware Predictive Coding (CAPC) method. CAPC combines contrastive predictive coding with the

Barlow Twins method, enhancing the model's ability to learn robust representations from unlabeled CSI time-series data. This approach improves model generalization, particularly when labeled data is scarce. CAPC also introduces a novel augmentation technique, dual view, which isolates free space propagation information from hardware distortions, further enhancing representation quality for WiFi sensing applications.

Through extensive evaluations, this thesis demonstrates the effectiveness of both RSCNet and CAPC. RSCNet achieves results on par with the state-of-the-art performance in HAR tasks while drastically reducing computational burdens on edge devices. CAPC outperforms baseline SSL approaches and traditional supervised methods, showcasing its superior generalization capabilities in unseen environments. The dual view augmentation further enhances CAPC's performance by reducing electronic distortions. This thesis concludes that RSCNet and CAPC contribute significantly to the advancement of robust and practical wireless sensing technologies. These frameworks address critical challenges in the field, paving the way for wider adoption of WiFi sensing in real-world applications.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Hina Tabassum, whose guidance, encouragement, and insightful feedback have been instrumental in the completion of this thesis. Her expertise and unwavering support have been invaluable throughout this journey.

I would also like to extend my heartfelt thanks to my committee members, Dr. Ruth Urner, Dr. Omer Waqar, and Dr. Kohitij Kar, for their valuable input, constructive criticism, and continuous encouragement. Their diverse perspectives have significantly enriched my research.

A special acknowledgment is reserved for Dr. Mohammad Omer from Cognitive Systems Corp., whose invaluable insights and support have been crucial to the practical aspects and direction of this research. His expertise and commitment to this project have been exceptionally inspiring and beneficial.

I am deeply thankful to all the members of the NGWN Lab for creating a stimulating and supportive environment. In particular, I would like to recognize Mohammad Amin Saeidi, Amirhossein Mohammadi, Mehrazin Alizadeh, Dr. Saba Asaad, and Zijiang Yan for their camaraderie and intellectual contributions.

I am immensely grateful to Fereshteh Forghani, whose unwavering support and encouragement have been a constant source of strength and motivation throughout

this journey.

Most importantly, I owe my deepest gratitude to my parents, Farahnaz and Hamid. Their unwavering love, sacrifices, and encouragement have been the foundation of my achievements. This thesis would not have been possible without their constant support and belief in me.

Contents

List of Figures

Chapter 1

Introduction

In today's digital age, wireless devices like smartphones, laptops, and Internet-of-Things (IoT) devices are everywhere. These devices use radio-frequency (RF) signals to communicate, connecting us in our connected world. However, these invisible signals, bouncing off objects like walls, furniture, and even us, have potential beyond communication. When captured and analyzed, they can reveal personal and environmental information without our knowledge.

This is the foundation of WiFi sensing, a cutting-edge field within wireless sensing technologies. It uses the properties of electromagnetic waves to passively gather information about an environment, eliminating the need for extra sensors. As objects and people interact with WiFi and RF signals, these signals are reflected, diffracted, and scattered. These interactions create multi-path effects that contain rich information about the surroundings, which can be decoded using two key metrics: channel state information (CSI) and received signal strength indicator (RSSI).

CSI provides a detailed metric encompassing wireless channel properties such as amplitude and phase, while RSSI measures the power level of the received signal affected by distance and obstructions. Variations in CSI caused by movements within

WiFi coverage can reveal intricate details about human activities, locations, and even vital signs. This ability to passively monitor and analyze an environment opens up a plethora of applications in fields ranging from healthcare and security to smart home automation and human-computer interaction [3].

WiFi sensing stands alongside other significant sensing technologies, such as IoT sensing and radar-based sensing. Traditional IoT sensing employs mechanical, electrical, optical, and chemical sensors to measure and convert physical entities into usable data. For instance, mechanical sensors like liquid thermometers and ultrasonic sensors use the deformation of sensing elements to gather information, while electrical sensors exploit principles such as the piezoelectric and piezoresistive effects to convert inputs into electrical outputs. Optical sensors, including infrared sensors and optical fiber sensors, detect objects by examining their influence on light paths, and chemical sensors measure changes through chemical reactions [4].

Radar-based active sensing, on the other hand, uses radio waves to detect and analyze objects within the environment. By transmitting radio waves and analyzing the echoes returned from objects, radar systems can determine a target's physical characteristics [5]. This technology finds extensive application in air traffic control, geophysical monitoring, weather observation, and defence and security surveillance [6].

1.1 WiFi Sensing: Benefits and Applications

The widespread availability of the internet has led to a significant increase in WiFi-enabled devices and access points (APs) in various environments, such as commercial and residential areas. WiFi APs have evolved from simple routers to sensor devices

for human sensing applications, enabled by the analysis of wireless signal characteristics like RSSI or fine-grained CSI. CSI, a detailed metric, includes wireless channel properties such as amplitude and phase, while RSSI measures the power level of the received signal affected by distance and obstructions. CSI captures variations in RF signals as they move through a physical space, interacting with objects or human bodies, causing reflection, diffraction, and scattering. These multi-path effects convey valuable information about the environment, including human movements, locations, and the state of objects [3].

WiFi sensing has the potential to redefine the sensing paradigm. It prioritizes privacy and facilitates ubiquitous, non-invasive, and passive sensing as users do not need to carry sensors [7]. WiFi sensing is cost-efficient as it capitalizes on existing WiFi infrastructure. Unlike systems limited by line-of-sight (LOS) constraints, wireless signals provide rich data through reflection and diffraction, even in non-line-of-sight (NLOS) scenarios where obstructions exist between the target and WiFi device. Notably, these signals can operate in the dark, offering round-the-clock functionality that cameras cannot match. WiFi signals permit the extraction of specific details, such as human position and vital signs, without visual information, making them more favorable where privacy concerns exist, such as smart homes and elder care facilities.

Since its introduction, WiFi sensing has been utilized for a variety of novel sensing tasks. The most prevalent applications are localization and human activity recognition (HAR). These are followed by uses in gesture recognition, crowd counting, occupancy detection, and health monitoring, such as respiration rate detection.

Localization

A prominent research focus in WiFi sensing is localization, which involves tracking a target's location within a space using ambient WiFi signals. Traditional WiFi-based localization methods necessitate that the target being tracked is equipped with transmitting or receiving hardware, as demonstrated in [8] where a set of fixed WiFi devices are positioned in the area. More recent studies, however, track human targets in a device-free manner, without requiring the individual to carry a WiFi device. For instance, the work of Zhou et al. [9] utilizes CSI data to create a database of environmental signal fingerprints corresponding to various physical positions within an indoor setting. Nevertheless, environmental changes can decrease the sensing accuracy of a WiFi sensing system. To address this issue, techniques such as domain adaptation [10], [11], [12] and transfer learning [13] have been proposed.

Human Activity and Gesture Recognition

WiFi Sensing has increasingly been employed in Human Activity Recognition (HAR). As in localization, where human presence affects signal propagation between transmitters and receivers, monitoring CSI variations enables the identification of subtle details about a person's actions. Not only can we capture stationary activities like sitting or standing, but also by capturing the temporal features of the captured CSI we can detect mobile activities like walking and running [14]. Fall detection is another application of HAR which safeguards the well-being of elderly or ill individuals without invading their privacy, unlike camera-based systems [3], [15].

Numerous studies aim to identify more intricate human movements, focusing on

hand and finger gestures [16]. Recognizing these fine-grained gestures enables innovative gesture-based interactions in smart home environments [17] and in-vehicle controls [18]. Sign Language detection is another use case of gesture recognition using WiFi Sensing [19]. Moreover, user authentication can be achieved by analyzing specific gestures performed by individuals, as these gestures reveal unique characteristics that can be used to identify valid users [20].

Crowd Counting and Occupancy Detection

Understanding the movement and counting of people in indoor environments is valuable for various applications, including customer mobility analytics, monitoring secure locations, and detecting people during rescue missions [21, 22, 23]. Typically, crowd counting is conducted as individuals move through an environment [24]. Stationary crowd sizes can also be estimated using WiFi signals by detecting subtle, random fidgeting movements within the crowd [25]. This method is beneficial in both safety-critical and non-critical situations, such as emergency evacuations, where it helps track the number of people exiting and remaining in a building [26], and in service environments like retail stores, airports, hospitals, and theme parks to enhance efficiency and user satisfaction. Additionally, in transportation, WiFi sensing can optimize bus and train schedules and improve boarding and payment procedures based on passenger counts, thus streamlining services [27]. However, the same WiFi signals used for these purposes can be exploited by adversaries to track individuals in non-public environments, raising privacy concerns [28].

Health Monitoring

Continuous health monitoring in private residences is necessary for patients and the elderly. However, wearable sensors may be inconvenient for users and camera-based systems can be intrusive and raise privacy concerns. WiFi sensing has emerged as a popular solution for health monitoring tasks due to its device-free and non-invasive nature. Specifically, respiration tracking [29], [7] is a common application, which can be achieved by identifying peaks in signal variation over time which are caused by the subtle chest movements of the person. Tracking respiration with CSI can also help detect irregular breathing patterns, such as apnea or tachypnea, especially in sleep [30]. WiFi sensing can also be applied to sleep monitoring and detecting unhealthy sleep actions, such as rhythmic movement disorders [31] and nocturnal seizures [32]. This method is suitable for privacy-friendly sleep monitoring as it does not require continuous audio or video recording throughout the night. More detailed sensing using CSI has been utilized to track individuals' heart rates, which can help reveal heart rhythm variability [33], [34].

1.2 Why Machine Learning for WiFi Sensing?

Traditional WiFi sensing methods leverage CSI, which captures the amplitude and phase of signals across different subcarrier frequencies, to gain insights into the wireless environment. These methods employ a variety of modeling-based algorithms to analyze the signal behavior and interaction with objects. The Free Space Propagation Model, for instance, estimates signal attenuation in line-of-sight scenarios, while the Fresnel Zone Model focuses on signal propagation and interactions within the Fresnel zones [35]. Techniques such as Angle of Arrival (AoA) and Angle of Departure

(AoD) help determine the directions of incoming and outgoing signals, respectively, and Time of Flight (ToF) measurements gauge the distance based on the propagation time of signals from the transmitter to the receiver [36], [37]. Additionally, the Multiple Signal Classification (MUSIC) algorithm is useful for estimating AoA and AoD, and the Power Delay Profile (PDP), obtained via the Inverse Fast Fourier Transform (IFFT) of CSI, aids in ToF estimations. CSI similarity metrics, which compute the cross-correlation of CSI matrices, effectively distinguish between static and moving objects [38]. Collectively, these diverse methodologies form the core of traditional WiFi sensing techniques, providing comprehensive insights into the dynamics of wireless environments. However, these traditional methods face several limitations:

- **Difficulty capturing complex human activities:** They struggle to model the intricacies and variations in human movements, like gait, due to their reliance on simplified models [2].
- **High effort in model establishment and parameter tuning:** Developing accurate models and finding the right parameters is often a laborious and time-consuming process.
- **Reliance on precise measurements and extensive signal processing:** The need for accurate data and complex signal processing can limit their applicability in real-world scenarios with noise and variability.
- **Lack of adaptability:** Traditional methods are often not easily reusable or adaptable to new tasks, scenarios, or environments, requiring significant rework for different applications [37].

These limitations pave the way for the exploration of machine learning techniques, and more specifically deep learning methods, which provide a more promising alternative. One of the major benefits of these methods is that they require little to no signal processing. This decreases the overhead of data preparation and pre-processing, making the models more efficient and easier to implement. Furthermore, machine learning methods are innately adaptable. They can improve and adapt as more training data becomes available. This characteristic is particularly prevalent in deep learning models, which are known for their capacity to improve with data volume. Deep learning models also offer significant automation benefits. They remove the need for intensive feature engineering or tuning of learning parameters. This not only reduces the manual effort involved in model development but also makes the process more efficient and less prone to human error. Furthermore, the re-usability of deep learning models makes them highly valuable. The same model can be used across different datasets or problem domains without the need for retraining from scratch. This provides a significant advantage in terms of time and computational efficiency. Moreover, deep learning models are not just reusable but also versatile. High-accuracy pre-trained models can be repurposed for a variety of tasks, making them an efficient choice for a broad range of applications.

The aforementioned strengths of machine learning, in particular deep learning, make them well-suited for the intricate task of WiFi sensing. They are capable of overcoming the limitations of traditional methods, promising a robust and adaptable approach to understanding and interpreting the complexity of human activities.

1.3 Challenges in Deep Learning-enabled WiFi Sensing

Deep learning techniques have shown great potential in advancing WiFi sensing applications. However, several challenges need to be addressed to fully exploit the capabilities of deep learning models in the context of WiFi sensing. In this subsection, we discuss some of the key challenges and discuss potential solutions to overcome them. By understanding and addressing these challenges, we can pave the way for the development of more efficient, accurate, and adaptable WiFi sensing systems in diverse real-world scenarios.

1.3.1 Complexity of Deep Learning Models

In practice, deployment of deep learning models can be challenging where there are low-power and low-cost devices with limited computational resources. Moreover, numerous WiFi sensing applications, including respiration rate detection [7], necessitate real-time processing in practice. Consequently, it is essential that models for these tasks are designed to be sufficiently streamlined, enabling them to leverage the processing capabilities of next-generation routers or smart home systems. One solution addressing this challenge is to perform sensing on the cloud instead of edge devices. However, this would result in high communication overhead due to CSI transmission. Therefore, careful consideration of the limitations of edge devices is important when designing deep learning models for sensing tasks, to ensure practicality and effectiveness.

1.3.2 Dataset Availability and Standardization

The advancement of deep learning applications in WiFi sensing is significantly constrained by the availability of extensive, accurately labeled datasets. Unlike data types that are more understandable, such as images or audio, CSI is inherently abstract and complex, making manual labeling impractical. This necessitates that annotation be conducted concurrently with data extraction, rather than as a separate post-processing activity. Furthermore, it is not common practice to publicly publish CSI datasets. Consequently, a prominent issue within the WiFi sensing literature is the absence of standardized and publicly accessible datasets for various sensing tasks. This deficiency hampers the ability to effectively compare the efficacy of proposed methods, posing additional challenges to the field's progression.

1.3.3 Environment Robustness

The primary challenge associated with WiFi sensing and its scalability pertains to the environmental distribution bias that impacts deep learning models. WiFi signals are influenced by a multitude of factors, including network settings, environmental conditions, and mobility situations. As a result, while many studies in this field have shown remarkable performance, most of these experiments use the same environment for both training and testing. This practice may lead to models that do not maintain the same level of accuracy when tested in new environments. This problem poses hurdles to the practical deployment of these systems in real-world scenarios due to the need to retrain the model with data from the new environment.

1.3.4 Self-supervised Learning Challenges

To address the challenges related to labeled datasets and environment generalization, recent methods, including semi-supervised [39, 40], and self-supervised [41, 42, 43, 44, 45, 46] learning, have been exploring the use of unlabeled CSI to reduce dependence on labeled data and adapt deep learning methods to unseen environments. Self-supervised learning (SSL), in particular, has attracted significant attention. It involves pre-training a network with unlabeled data on a pretext task to generate adaptable representations for downstream tasks like sensing. However, SSL is not without its challenges. One major issue is identifying appropriate pretext tasks, which can vary greatly between applications. For example, augmentations like image rotation and color saturation, common in computer vision, are not applicable for CSI WiFi sensing, leaving the identification of suitable pretext tasks for this domain as an open question.

Furthermore, SSL methods can suffer from "collapse," where trivial solutions are learned instead of meaningful representations. While recent research has proposed solutions like contrastive learning, stop-gradients, and extra predictors [47, 48, 49] to prevent this collapse. Yet, the effectiveness of these methods in WiFi sensing has not been thoroughly investigated.

1.4 Scope of the Thesis

This thesis explores innovative approaches in robust representation learning tailored for wireless sensing applications, specifically focusing on WiFi-based sensing. The widespread deployment of WiFi infrastructure combined with the detailed data provided by CSI positions WiFi sensing as a promising avenue for HAR and similar

applications. Despite this potential, several challenges hinder the broader adoption of WiFi sensing technologies, including the limited computational resources of edge devices, the high dimensionality and rapid sampling rates of CSI, the scarcity of labeled data for supervised learning, and the strong environmental dependence of CSI-based methods.

The thesis addresses these challenges by focusing on two primary areas: efficient joint CSI compression and sensing, and SSL for robust representation learning.

1.4.1 Efficient Joint CSI Compression and Sensing:

The initial component of this thesis addresses the computational burdens associated with transmitting high-dimensional CSI data from edge devices to cloud servers. Traditional approaches often demand extensive computational power that edge devices may lack and typically perform sensing tasks entirely on the device. To counter this, a new network architecture called the **R**eal-time **S**ensing and **C**ompression **Net**work (**RSCNet**) is introduced. RSCNet facilitates efficient cloud-based sensing through joint CSI compression and sensing, enabling the compression of CSI data at the edge before its transmission to the cloud. Unlike most current methods, the cloud server then undertakes both the sensing task and the restoration of the original CSI for archival purposes. This method significantly lowers both the communication overhead and computational demands on edge devices while preserving sufficient sensing accuracy.

1.4.2 SSL for Robust Representation Learning:

The lack of labeled data for WiFi sensing tasks presents a substantial challenge. Labeling CSI data usually requires real-time annotation, which can be labor-intensive and costly due to the complex nature of CSI. Models trained on limited labeled data typically exhibit poor generalization, especially in novel environments. To address this issue, the second part of this thesis delves into the use of SSL techniques for WiFi sensing. SSL allows models to derive meaningful representations from unlabeled data, decreasing reliance on manual annotation and enhancing model generalization. This thesis introduces a specialized SSL method for time-series CSI data, the **C**ontext-**A**ware **P**redictive **C**oding (**CAPC**) framework, which accounts for the temporal dynamics and characteristics of wireless channels. CAPC integrates contrastive [50] and non-contrastive [51] SSL methods to improve the learning of robust, feature-rich representations. Utilizing unlabeled data, these methods foster the development of robust representations that adapt to new environments and new tasks with only a few labeled data, thereby boosting the performance and relevance of WiFi sensing systems.

Together, these studies form a cohesive thesis theme focused on advancing the robustness and practicality of wireless sensing technologies through innovative representation learning strategies. These strategies aim to overcome typical obstacles such as computational constraints at the edge and the scarcity of labeled data, prevalent in real-world scenarios.

1.5 Contributions

This thesis makes several key contributions to the field of wireless sensing, which are encapsulated in two main studies:

1. RSCNet: Real-time Sensing and Compression Network for Cloud-Based Wi-Fi Sensing:

    - Introduction of RSCNet, a novel architecture designed for real-time cloud-based Wi-Fi sensing, featuring an encoder for CSI compression at the edge and a multi-task network in the cloud for both WiFi sensing and CSI reconstruction.
    - Demonstration of the significance of window-based CSI compression, which enables efficient real-time HAR with reduced communication overheads compared to traditional fixed-duration sampling.
    - Incorporation of LSTM-based recurrent blocks to leverage the time-series representation of CSI windows, leading to improvements in both reconstruction and human activity classification accuracy.
    - Achievement of state-of-the-art HAR performance on the UT-HAR dataset [1], with computational demands on the edge device reduced to about 1/50 of those required by baseline models.

2. CAPC: Context-Aware Predictive Coding for Self-Supervised Representation Learning in WiFi Sensing:

    - Development of a novel SSL framework, CAPC, tailored for CSI WiFi sensing. CAPC combines contrastive predictive coding autoregressive design

and Barlow Twins dual network to capture temporal dynamics and create contextually consistent and robust representations.

- Introduction of a new augmentation technique that leverages both uplink and downlink CSI, enhancing model generalization by isolating free space propagation effects and minimizing electronic distortions and CSI estimation errors.
- Comprehensive analysis of various augmentations for time-series CSI data, identifying the optimal combination for WiFi sensing and different SSL methods.
- Demonstration of CAPC's enhanced ability for representation learning on the SignFi [19] and UT HAR [1] dataset outperforming both baseline SSL approaches and supervised methods in unseen environments and new tasks.

1.6 Research Outcomes

- B. Barahimi, H. Singh, H. Tabassum, O. Waqar, and M. Omer, "RSCNet: Dynamic CSI Compression for Cloud-based WiFi Sensing," in Proc. of IEEE International Conference on Communications (ICC), Denver, USA, 2024.
- B. Barahimi, H. Tabassum, O. Waqar, and M. Omer, "Context-Aware Predictive Coding: A Representation Learning Framework for WiFi Sensing," (submitted).

Chapter 2

Fundamentals and Literature Review

In this chapter, we provide the necessary preliminaries and review relevant work on WiFi sensing and deep learning methodologies that form the foundation of our approach. We will also discuss various applications of WiFi sensing to contextualize its significance and utility in different domains.

2.1 Preliminaries

2.1.1 Channel State Information

CSI characterizes the way wireless signals propagate from a transmitter to a receiver over specific carrier frequencies. Employing Multiple-Input Multiple-Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) techniques in the physical layer of the IEEE 802.11n/ac standards improve the transmission channels impacted by multi-path propagation. It offers a detailed description of the wireless environment, highlighting how it is influenced by multi-path effects. These effects emerge from variations and movements of the transmitter, receiver, and nearby objects. The CSI, represented as H, defines the transformation of the transmitted signal x to produce

the received signal y, accounting for noise η at a given timestamp t, i.e., $y = Hx + \eta$. CSI can be represented using the Channel Impulse Response (CIR), i.e., $h(\tau) = \sum_{n=1}^{N} a_n \delta(t - \tau_n)$, where the factors a_n and $\tau_n(t)$ describe the amplitude attenuation and propagation delay of the nth path at timestamp t, respectively. $\delta(t)$ denotes the Dirac-delta function. Subsequently, the channel frequency response (CFR) at carrier frequency f and time t can be modelled as follows:

$$H(f;t) = \sum_{n=1}^{N} a_n e^{-j2\pi f \tau_n(t)}, \tag{2.1}$$

To measure CSI, the WiFi transmitter emits known Long Training Symbols (LTFs). Using these symbols and the received signals, the receiver calculates the CSI matrix [37]. The CSI for each subcarrier i at timestamp t can be modeled as a complex number $H_{f_i,t} = |H_{f_i,t}|e^{j\angle H_{f_i,t}}$. Both the amplitude $|H_{f_i,t}|$ and phase $\angle H_{f_i,t}$ are impacted by the displacements and movements of the transmitter, receiver, and surrounding objects and humans. In contrast to the received signal strength, CSI offers superior resolution for sensing tasks. This is attributed to its precise characterization of phase shift and amplitude attenuation for each subcarrier. This comprehensive information allows CSI to serve as *WiFi image frames* of the environment, providing a nuanced understanding [2].

In WiFi sensing, a sequence of *CSI frames* compiles into a matrix $H = \{H_t \in \mathbb{C}^{N_a \times N_s}\}$. Here, N_a and N_s represent the number of antennas and subcarriers, respectively, while N_t, the length of H, stands for the time dimension.

Remark: By segmenting H over time dimension, it becomes possible to generate a series of distinct windows, with each window encompassing a given number of CSI frames N_f. In this thesis, these specific constructs will be consistently referred to

as *CSI window*. For WiFi sensing applications, typically the amplitude information $|H_t|$ is deemed sufficient. Thus, we prioritize utilizing the amplitude $|H|$ for sensing applications.

2.1.2 CSI Extraction Tools

While CSI is an essential part of WiFi chips for wireless communication, not all off-the-shelf WiFi cards report it. The most widely used tool for CSI measurements is the Linux 802.11n CSI Tool [52], which uses Intel 5300 WiFi cards to report compressed CSIs for the Intel WiFi Wireless Link 5300 802.11n MIMO radios recording 30 subcarriers for each pair of antennas. The Atheros CSI Tool [53] reports uncompressed CSIs using Qualcomm Atheros WiFi cards increasing the CSI data resolution by recording 56 CSI subcarriers in a 20MHz WiFi channel and 114 in 40MHz. The Nexmon CSI Tool [54], [55] enables CSI recording on smartphones and Raspberry Pi with a some of the Broadcom WiFi chips and can capture 256 subcarriers for 80MHz. However, previous research [56], [57] has shown that the CSI data generated by the Nexmon CSI Tool can be quite noisy. More recently, the ESP32 CSI Toolkit [58] enables the collection of CSI from the ESP32 WiFi-enabled microcontroller, enabling a simpler and more lightweight solution for CSI collection.

The availability of CSI collection tools and hardware has led to the development of commercial WiFi-based sensing solutions offering services like security, elder care, and home automation [59]. Despite these advancements, the scope of applications remains constrained because the IEEE 802 standard lacks specifications for sensing features, leading to proprietary solutions using outdated communication standards like 802.11n with limited interoperability. To address this, the IEEE 802.11bf Task

Group (TGbf) was established in September 2020 to work on an amendment to the IEEE 802.11 standard to better support WiFi sensing capabilities [60].

2.1.3 CSI WiFi Sensing Datasets

WiFi sensing systems increasingly leverage machine learning methods, making publicly available datasets crucial for training and validating deep learning models. Yet, the availability of CSI datasets for WiFi sensing is limited due to: (1) the aforementioned challenges in CSI collection from WiFi chips, (2) the high cost of CSI annotation, necessitated by the unreadability of CSI data to humans, and (3) the uncommon practice of publishing CSI datasets.

Yousefi et al. [1] published the first publicly available CSI dataset for Human Activity Recognition (HAR) in an office setting, followed by a segmented version known as UT-HAR [2]. Subsequently, several CSI datasets were released tailored for HAR across various environments [61, 62, 63, 64]. Baha et al. [65] developed a dataset accommodating both line-of-sight and non-line-of-sight conditions. Specific-purpose datasets have been introduced, such as FallDeFi [66] for fall detection, ARIL [67] for HAR and localization. Additionally, SignFi [19] was introduced for sign language recognition which offered synchronized uplink and downlink CSI for two environments. For benchmarking, Yang et al. [2] introduced the NTU-Fi dataset, aimed at evaluating sensing models in terms of performance, size, and complexity for HAR and human identification tasks.

There's a growing trend towards creating diverse datasets that encompass a wider range of environments, users, and hardware specifications. Datasets such as CPAR [68] and RF-NET [69] provide multi-user and multi-environmental data, respectively.

To capture more environmental variability, CSIDA [70] and Widar 3.0 [71] recorded CSI data across different locations, orientations, and user scenarios. OPERAnet [72], a multimodal dataset for HAR, encompasses WiFi CSI alongside data from Passive WiFi Radar (PWR), Ultra-Wideband (UWB), and Kinect skeleton sensors. MM-Fi [73] is another multimodal dataset, which includes WiFi CSI, video frames, depth frames, LiDAR point clouds, and mmWave radar point clouds. Additionally, Meneghello et al. [74] created the SHARPax dataset to explore the effects of varying wireless parameters, such as bandwidth and transceiver hardware, as well as environmental and user factors on HAR. The WiMANS [75] dataset was introduced for simultaneous multi-user sensing in a multi-environment setting which also includes video frames for multi-modal training and comparison with computer vision solutions for sensing.

2.1.4 Deep Learning Models

Recent advances in deep learning-enabled sensing have demonstrated remarkable potential in extracting and modelling complex patterns from CSI without intensive feature engineering [76, 37]. Unlike modeling-based methods that rely on handcrafted features created using human expertise, deep learning automatically extracts features by learning from large datasets and optimizing the model through backpropagation [2].

In WiFi sensing, we typically encounter classification tasks. For instance, in HAR, the model needs to classify a person's activity into one of several predefined categories based on CSI data. A standard deep learning classification model consists of two main components: a feature extractor (or encoder) and a classifier. The feature extractor's

role is crucial as it transforms the CSI data into a more compact, lower-dimensional representation that retains essential features relevant to the task. This encoding is designed to highlight patterns in the WiFi sensing data that are critical for accurate activity recognition. Following this, the classifier component, generally composed of multiple fully connected layers or traditional classifiers such as support vector machine (SVM) and Random Forest, takes over. Its primary function is to learn an effective decision boundary based on the representations provided by the encoder. This decision boundary is vital for distinguishing between the different activities.

The design of the encoder is a pivotal aspect in successfully extracting meaningful patterns from the CSI data for WiFi sensing tasks. This process demands a careful balance of dimensionality reduction and feature preservation to ensure the model performs well in recognizing and classifying different human activities based on WiFi signals.

Below, we explore several fundamental deep learning architectures and their application in WiFi sensing tasks.

Multilayer Perceptron

The Multilayer Perceptron (MLP) is a fundamental architecture in deep learning and commonly serves as a classifier in many deep neural networks. It typically comprises several fully connected layers, each followed by an activation function. The output of each node in a layer is determined by the equation:

$$y = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{2.2}$$

In this formula, x_i denotes the ith input to the layer, and w_i is the corresponding

weight. The variable n is the total number of inputs to the layer. The term b stands for the bias associated with the node. The function σ represents the activation function, which introduces non-linearity into the MLP model. The variable y is the output of the node.

The input CSI or its processed variant through an encoder is typically transformed into a flat vector $x \in \mathbb{R}^{N_a N_s N_t}$, where it combines dimensions related to the antenna, subcarrier, and time. This transformation ignores the inherent structure within the CSI data. Although MLPs can be effective, particularly with abundant labelled data or simpler classification tasks, they are computationally expensive due to the large size of the input vector that mixes various dimensions. Consequently, MLPs are rarely used for feature extraction. Instead, they are more commonly employed in the final classification stage, where they operate on a compact representation produced by an encoder.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are inspired by the biological processes observed in the visual cortex of the brain [77] and were originally designed for image recognition tasks [78]. Unlike MLPs, CNNs utilize shared weights and spatial pooling to efficiently process information. The fundamental operations in CNNs include convolutional kernels and pooling layers that extract and integrate hierarchical features from images.

In CNNs, convolutional layers employ 2D filters to interact with local regions of the input image, capturing basic visual features in the initial layers. As information progresses through deeper layers, these features are combined to form more complex

representations suitable for the downstream task. The convolution operation, defined without bias, is as follows:

$$C(\boldsymbol{I}, \boldsymbol{K})[i, j] = \sum_{m} \sum_{n} \boldsymbol{I}[i + m, j + n] \cdot \boldsymbol{K}[m, n], \tag{2.3}$$

Here, C represents the convolution operation, $\mathbf{I}$ is the input image, and $\mathbf{K}$ is the convolution kernel, with m and n indexing the spatial dimensions of the kernel.

To reduce the spatial dimensions of feature maps while preserving important features, CNNs often incorporate pooling layers that perform operations such as max pooling or average pooling. These layers calculate the maximum or average value within a local region, effectively downsampling the input.

Additionally, CNN architectures may include variations such as strided convolutions, which provide a memory-efficient downsampling method [79], and transposed convolutions, which are used primarily in autoencoders for upsampling [80]. Another notable variant is the dilated convolution [81], which introduces gaps into the convolution kernel using a *dilation rate*. This operation allows the network to expand its receptive field without increasing the computational burden:

$$D(\boldsymbol{I}, \boldsymbol{K})[i, j] = \sum_{m} \sum_{n} \boldsymbol{I}[i + dm, j + dn] \cdot \boldsymbol{K}[m, n], \tag{2.4}$$

In this equation, D stands for dilated convolution, and d is the dilation rate. The dilation modifies the kernel size to $k' = k+(k-1)(d-1)$, where k is the original kernel size, and k' is the effective size post-dilation, demonstrating the increased receptive field of the dilated layer. Notably, we utilized this type of convolution in our model designed to decrease computational demand and make them suitable for low-cost IoT

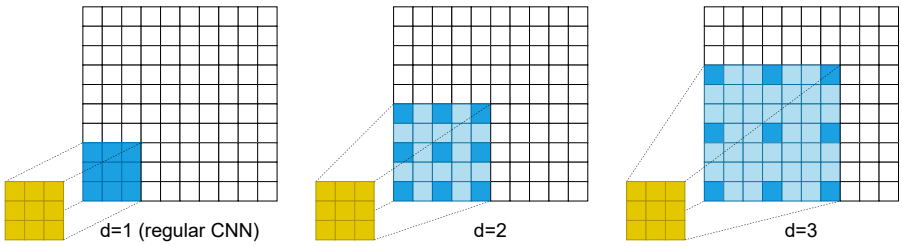


Figure 2.1: Illustrations of dilated convolution operations with varying dilation rates are shown, where solid areas indicate the active operations and shadowed areas represent the receptive field. These visuals demonstrate that increasing the dilation rate expands the receptive field without altering the kernel size or computational complexity.

devices. Figure 2.1 showcases the operational effect of the dilated convolutional layer vs the regular CNN.

The way CSI is fed into CNNs is crucial, especially considering that CNNs were initially designed for image data with dimensions of height, width, and color channels. In contrast, CSI involves dimensions of time, subcarriers, antennas, and potentially the link dimension when synchronized CSI from several wireless links is available. Typically, in WiFi sensing applications using CNNs, the antenna dimension is treated similarly to the color channels in images. Each antenna, capturing slightly different signal propagation, provides a unique perspective of the environment, akin to how color channels capture different visual information.

Recurrent Neural Networks

One of the most popular approaches for sequence modelling is the use of recurrent neural networks [82], initially developed to include a notion of persistence to hidden representations and handle variable sequence lengths. The key idea is to leverage

parameter sharing to apply the same model iteratively to each data point while maintaining a notion of recurrence. RNNs implement recurrence by having the output of the model at the current timestep be a function of the output or hidden state from the previous timestep. Its principle is to create internal memory to store historical patterns, which are trained via back-propagation through time [83].

Recurrent Neural Networks (RNNs) are a fundamental method for sequence modeling, originally designed to incorporate persistent hidden representations and adapt to sequences of varying lengths [82]. The core concept behind RNNs involves the repetitive application of the same model to each element of a sequence, using shared parameters and maintaining a recurring process. RNNs achieve this by feeding the output or hidden state from one timestep as an input into the next timestep. This process is essentially building an internal memory that captures historical data patterns, which is trained using back-propagation through time [83].

For an input sequence represented as $x = \{x^t \in \mathbb{R}^{N_s}\}_{t=0}^{\tau}$, the hidden state at any timestep t is determined using the model's parameters $\boldsymbol{W_x}$ and $\boldsymbol{W_h}$ in the following manner:

$$h^t = \sigma(\boldsymbol{W_x} x^t + \boldsymbol{W_h} h^{t-1}) \tag{2.5}$$

where activation function σ is usually Tanh or Sigmoid functions. This design of RNN gives the unique advantage of giving the option to utilize all of the hidden outputs of each timestep $h^{1\cdots\tau}$ or just utilizing the last one h^{τ}. However, RNNs suffer from vanishing gradient problems during backpropagation and thus cannot capture long-term dependencies of data. Therefore, some variations of RNN has been proposed such as LSTM [84] and GRU [85] units which handle the gradient

instabilities and have longer memories by adding memory cells for the cost of more computational power.

In this equation, σ represents an activation function, which is typically either the Tanh or Sigmoid function. This RNN configuration offers flexibility in that it allows the utilization of either all hidden states $h^{1\cdots\tau}$ or just the final state h^{τ}. However, standard RNNs often encounter issues with vanishing gradients during backpropagation, limiting their ability to model long-term data dependencies. To address this, variants such as LSTM [84] and GRU [85] have been developed. These adaptations introduce memory cells that stabilize gradients and extend memory capability, albeit at a higher computational cost.

Additionally, in applications such as CSI WiFi sensing, individual or sequences of frames may be initially segmented into windows and processed with a standard CNN model. This preprocessing step extracts features and compresses the data dimensionality before it is fed into an RNN to capture temporal dynamics [86], [71]. This technique effectively reduces the computational burden on the subsequent RNN.

Transformers

Transformers, specifically with the introduction of the attention mechanism [87], emerged as a powerful alternative to RNNs for NLP applications. They were developed to mitigate the high computational demands and limited memory span of RNNs. The fundamental structure of a transformer includes both an encoder and a decoder, each comprising positional and linear embeddings, multiple layers of multi-head attention, a multi-layer perceptron (MLP), and layer normalization.

The self-attention mechanism of transformers is defined as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{2.6}$$

where $\boldsymbol{Q}$ (query), $\boldsymbol{K}$ (key), and $\boldsymbol{V}$ (value) are matrices generated from the linear embeddings of CSI patches $x^i \in$, with i ranging from 0 to T. Here, d_k denotes the dimensionality of the key vectors. Attention is calculated across all patch pairs using cosine similarity, with each individual calculation representing an attention head within the potentially multi-head setup.

The transformer architecture was later extended for computer vision where one image consists of many patches [88] and the decoder part is usually replaced with a classifier and only the coder is utilized. In CSI WiFi sensing, a similar methodology is considered in works such as THAT [89]. However, the transformer has a great number of parameters that makes the training cost expensive, and enormous labelled CSI data is hard to collect, which makes transformers not really attractive for supervised learning [2].

The utility of transformers has been extended to computer vision [88], where an image is divided into numerous patches. Typically, in such applications, the decoder is often replaced by a classifier, primarily utilizing the encoder. This approach is also applied in CSI WiFi sensing scenarios as demonstrated in works like THAT [89]. Despite their effectiveness, transformers have a significant drawback due to their extensive number of parameters, leading to high training costs. Furthermore, the challenge of acquiring large amounts of labeled CSI data makes transformers less appealing for supervised learning tasks [2].

2.2 Related Works

2.2.1 Deep Learning Methods for WiFi Sensing

Various approaches, such as LSTM-based methods are used to extract temporal information from CSI data [90], [91], CNN commonly is used to extract spatial features [92] and compress CSI data [86]. Additionally, THAT [89] proposed a two-stream Transformer-based model to utilize both time-over-channel and channel-over-time features. Also, ResNet architectures with attention mechanisms for gesture recognition [93] were proposed as well. DeepSense [94] utilized a combination of autoencoder, CNN, and LSTM to reduce noise, extract features and capture temporal features for HAR application. SecureSense was proposed to prevent adversarial attacks to sensing prediction by mitigating the effect of the attacks on the distribution of the prediction.

Notably, some studies have also addressed the environmental dependencies of deep learning methods. Widar3.0 [71] used the hand-crafted environment-independent feature called BVP as the input of their deep learning model and AFEE-MatNet [95] applied a matching network to learn common features among environments.

2.2.2 Self-supervised and Semi-supervised Learning

Nevertheless, the aforementioned research works follow a supervised approach, thus necessitating large volumes of labeled CSI data. A fundamental challenge is to train the model without labels, while enabling generalization across different environment settings. Recently, a handful of research works considered semi-supervised learning to reduce the reliance on labeled datasets. For instance, WiADG[40], where a pretrained encoder, initially trained in a supervised manner, is fine-tuned for new environments using adversarial networks. This semi-supervised approach can adapt to

new settings without the additional labelled data in the target environment, though it is contingent on having labelled data for the source environment. Fidora [39] used a semi-supervised approach by employing Variational Auto Encoders (VAEs) to augment the labelled dataset and create synthetic data. Subsequently, a feature extractor is applied to generate a representation from labeled and unlabeled samples. These representations are then processed through a decoder and classifier for CSI reconstruction and classification, respectively. SiFall [96] also leveraged VAEs in conjunction with anomaly detection techniques for fall detection. BTS [97] proposed a semi-supervised teacher-student learning approach inspired by BYOL [49] to solve time-varying effects induced by environment changes.

SSL is emerging as a powerful technique that leverages unlabeled data to train deep learning models, generating effective representations without the need for explicit labels, but requires knowledge of what makes some samples semantically close to others [98]. In the initial phase, the model undergoes pre-training using unlabeled dataset, thereby generating useful representations from the data without relying on explicit labels. In the subsequent stage, trained encoder weights are typically frozen, and the model shifts to supervised learning using a limited labeled target dataset. During this phase, fine-tuning takes place, exploiting the previously learned representations to improve the performance.

Recent SSL-based WiFi sensing methods, such as RF-URL[99], Lau et al. [42], STF-CSL[43], and DualConFi[45], learn invariant representations by contrasting or aligning different views of the input samples, which are created through various augmentation techniques. These methods focus on creating representations that minimize

the distance between similar instances in an embedding space. RF-URL used doppler-frequency-spectrum, AoA, and ToF data as augmentations and InfoNCE contrastive loss function. STF-CSL integrated short-time-fourier-neural networks (STFNets) in their encoders, with a variety of frequency and time-domain augmentations. In [42], different views of CSI corresponding to a specific activity (or sample) captured by several receivers placed at various locations are considered positive samples, while views of other unrelated samples are treated as negative samples. Except for RF-URL, these methods adopt a contrastive learning strategy akin to SimCLR [47], using the NT-Xent loss function. More recently, AutoFi[41] has emerged with a non-contraststive geometric SSL method and few-shot learning for WiFi sensing.

However, the aforementioned research overlooks the temporal aspects of CSI, wireless propagation channels, and transceiver characteristics. Additionally, the considered augmentation techniques often do not align well with the inherent nature of CSI data. Furthermore, recent advancements in non-contrastive methods [51], [100], which eliminate the need for negative samples, have not been explored in the WiFi sensing domain.

2.2.3 CSI Compression

Other than the issue of labelled CSI datasets, the complexity of the DNN models is another problem. Although there have been masses of works on robust WiFi sensing systems using deep learning models, current methods demand substantial computation resources. In practice, IoT devices at the edge suffer from limited power and computation capabilities. To overcome this issue, transmission of CSI to cloud servers becomes necessary to enable cloud-based WiFi sensing. Yet, the high dimensionality

and sampling rate of CSI lead to a substantial transmission overhead [101]. As such, there is an undeniable need to compress the CSI at the edge before its transmission to the cloud. In addition, besides sensing, CSI reconstruction is becoming essential for data logging at cloud servers in systems like Healthcare IoT where low latency and real-time sensing are essential [102]. Thus, cloud servers not only need to host accurate sensing but also require CSI reconstruction to maintain data logs for legal reasons [103].

None of the aforementioned research works considered the problem of joint CSI compression and sensing. Recently, EfficientFi [103] tackled sensing and CSI compression jointly by compressing CSI into a quantized low-dimensional space at the edge using a trainable shared codebook between the AP and the cloud server. However, EfficientFi is still vulnerable to high communication overhead as its decoder utilizes max-unpool layers in the cloud which requires the indices from the max-pool layers of the encoder to up-sample the CSI to its original dimension.

While the exploration of CSI compression for WiFi sensing remains limited, it has been extensively studied in massive multiple-input multiple-output (MIMO) systems. In such systems, CSI is critical for both user equipment (UE) for signal detection and base stations (BS) for signal precoding. CSI is estimated at the UE and subsequently transmitted back to the BS via feedback links, which is essential for operational efficiency [104]. However, as the number of antennas increases, so does the size of the CSI matrices, significantly increasing the feedback overhead. Thus, effective CSI compression is essential to reduce both the overhead and latency of CSI feedback [101].

Traditional compressed sensing (CS) methods like LASSO [105] and BM3D-AMP

[106] have been utilized to compress the CSI matrix at UEs and reconstruct it at the BS. However, these methods face several limitations. They assume perfect CSI sparsity, overlook channel statistics in random projections, and induce considerable latency due to their iterative nature. To overcome these challenges, deep learning-based approaches have been introduced for CSI feedback compression. CSINet [107] and its LSTM-enhanced variant [108] have been shown to surpass traditional CS-based methods. Additionally, recent work by Mashhadi et al. [109] has explored CSI compression in a distributed setting, achieving efficient compression and reconstruction of CSI from multiple users. Furthermore, recently, DCRNet [101] employs dilated CNN layers and a flexible decoder design, reducing model complexity.

Chapter 3

RSCNet: Joint CSI Compression and Reconstruction

3.1 Introduction

While deep learning has shown great promise in WiFi sensing applications, the computational demands of these models pose significant challenges for resource-constrained IoT devices at the edge. Figure 3.1 illustrates the computational complexity of various deep learning models commonly employed in WiFi sensing [2], revealing that even the most streamlined architectures necessitate millions of FLOPs. This computational burden can severely impede the real-time sensing capabilities of these devices.

To address this issue, transmitting raw CSI to cloud servers for processing has become a common practice. However, the high dimensionality and sampling rate of CSI data result in substantial transmission overheads. As a result, compressing CSI at the edge before transmission is crucial for efficient cloud-based WiFi sensing. Furthermore, in applications like health care, CSI reconstruction is essential for data logging at cloud servers for archival purposes, necessitating a solution that not only enables accurate sensing but also facilitates CSI reconstruction. Our contributions

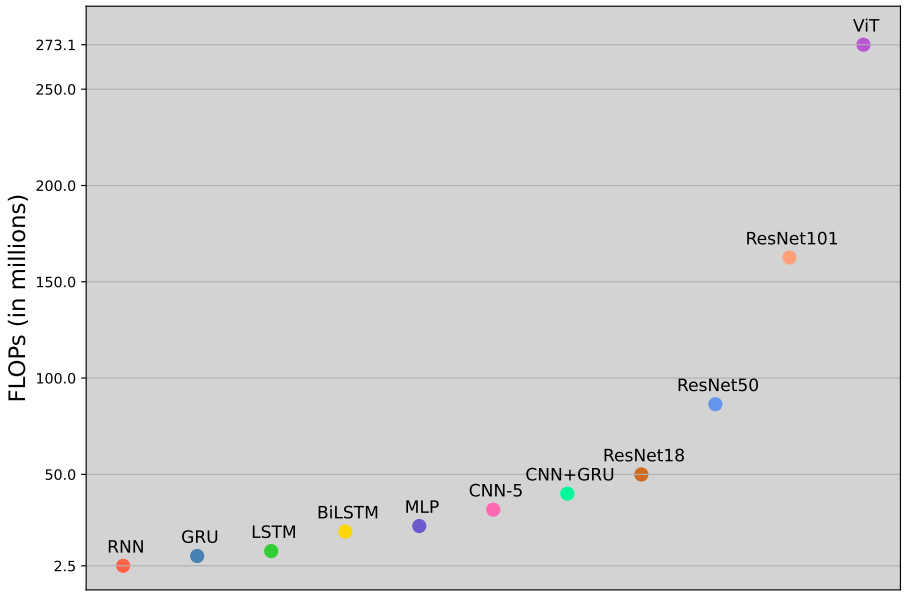


Figure 3.1: Comparative analysis of computational complexity across various deep learning models applied to the UT-HAR dataset [1], [2]

are summarized as follows:

1. We present RSCNet, a novel network architecture designed specifically for real-time cloud-based Wi-Fi sensing for HAR. The network encompasses an encoder for CSI compression at the edge and a dual-network in the cloud for both WiFi sensing and CSI reconstruction. The proposed architecture offers a lightweight encoder by using dilated convolutional layers and residual connections to be employable in low-resource edge devices and an expandable decoder design to optimize the trade-off between complexity and reconstruction error.

2. We demonstrate the significance of window-based CSI compression, resulting in efficient real-time HAR with reduced communication overheads compared to traditional fixed task duration based samplings.

3. We incorporate LSTM-based recurrent blocks to leverage the time-series representation of CSI windows, improving compression and human activity classification.

4. Through extensive experimentation on UT-HAR dataset [1], we observe that a window size of 50 frames strikes a balance, yielding a peak HAR accuracy of 97.2% and NMSE of -22.759 dB, under a compression ratio of $\eta = 1/90$. The framework maintains its robust performance across a range of compression ratios. We note that expanding the decoder with an expansion rate of 5 results in a reduction of reconstruction error by 0.84 dB compared to an expansion rate of 1, albeit at the cost of approximately 9 times higher Floating Point Operations Per Second (FLOP) counts.

5. We showcase that utilization of RSCNet in a edge-cloud setting can result in about 73% to 98% reduction of complexity on edge device when compared to deployment of SenseFi baselines [2] on edge devices.

6. We have made RSCNet's source code available on GitHub at `https://github.com/bornabr/RSCNet`.

3.2 Method

In this section, we discuss our proposed framework, RSCNet, and give details on the real-time compression of CSI windows and the multi-task learning process of CSI reconstruction and HAR as demonstrated in Fig. 3.2.

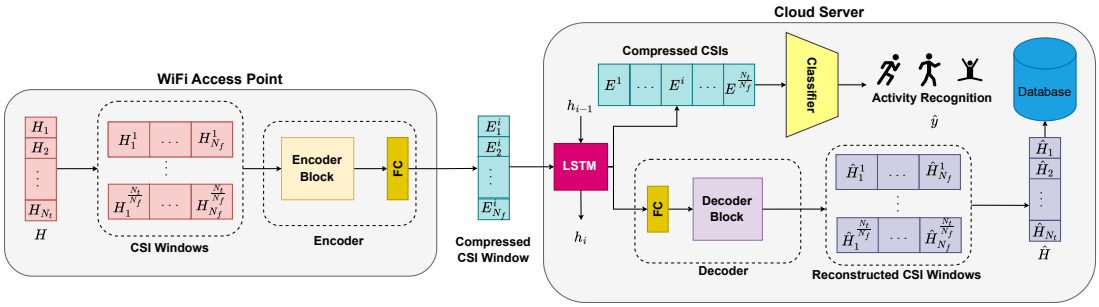


Figure 3.2: Design of the proposed RSCNet system

3.2.1 System Overview

Our model, RSCNet, is bifurcated into the edge and cloud models. The edge model is employed at the edge devices or access points (APs) where CSI is acquired. A CSI window, which contains several frames of CSI undergoes compression through an encoder. Compressed CSI embedding is then transmitted to the server where the cloud model, which incorporates recurrent blocks for temporal CSI enhancement, is used to do CSI reconstruction using a decoder. Additionally, a classifier accumulates the outputs of recurrent blocks for all of the compressed CSI windows of an activity to do sensing.

The encoder and decoder are inspired by DCRNet [110]. The encoder has an encoder block for feature extraction and a fully connected layer (FC) which receives the flattened version of the output of the encoder block to reduce the data size based on the compression rate (η). Similarly, the decoder has an FC layer to bring the size of data back to its original one and reshape it to make the CSI matrix then it goes to a decoder block which aims to restore the original CSI. This architecture utilizes dilated convolutional layers to enable large respective fields while keeping a minimal number of parameters, ensuring a lightweight model design suitable for edge

devices. Moreover, this decoder design provides a trade-off between the decoder's number of parameters i.e. computational cost and its reconstruction capabilities via an expansion rate hyperparameter, ρ.

Other than CSI reconstruction, the cloud model performs HAR using a multi-layer perceptron (MLP) as the classifier. RSCNet's objective is to achieve high HAR accuracy with minimal error in CSI reconstruction due to compression. Consequently, RSCNet is capable of compressing CSI to a low-dimension space, without diminishing the HAR accuracy.

3.2.2 CSI Windowing for Real-Time Compression and HAR

RSCNet utilizes segmentation of the continuous CSI data stream, represented as $N_a \times N_s \times N_t$, along the time dimension into distinct windows of size $N_a \times N_s \times N_f$, where $N_f \leq N_t$ is the number of CSI frames in each window. This segmentation enables the system to manage data more effectively, facilitating both real-time HAR and efficient compression. After feature extraction in the edge model, the output is compressed, with the compression ratio, $\eta \in [\frac{1}{N_a \cdot N_s \cdot N_f}, 1]$, determining the degree to which the CSI window size is reduced. For instance, $\eta = 1/90$ means that the compressed CSI window is 90 times smaller than the original CSI window.

In the cloud, each compressed CSI window undergoes the recurrent block to extract features related to the previous CSI windows. Then, it goes through the reconstruction process. Once a CSI window is restored, it gets merged with other windows to regenerate the complete original CSI activity sample. For classification/HAR, all compressed CSI windows are stacked to form an embedding that captures the essence of the original CSI, denoted as $(N_a \cdot N_s \cdot N_h \cdot \eta) \times (N_t/N_f)$, where N_h is the hidden

state size of the recurrent block. This data is subsequently flattened and given to the classifier for HAR. The RSCNet framework, by strategically handling CSI data in real-time, ensures computational efficiency, minimizes latency and preserves data's temporal fidelity.

3.2.3 Recurrent Block Integration

To further optimize the efficiency and accuracy of CSI reconstruction and HAR, we incorporate a recurrent block at the beginning of the cloud model. This block utilizes LSTM units, well-recognized for their powers in capturing temporal relationships in sequential data. Each LSTM receives its input from two sources: the current compressed CSI window and the hidden state from the preceding CSI window. This allows the LSTM to benefit from the temporal continuity inherent in sequential CSI windows, enhancing its understanding of the current window in the context of previous data.

After processing through the LSTM, its output serves a dual purpose. Firstly, it's fed into the decoder to refine the CSI reconstruction, ensuring it is both accurate and contextually relevant. Secondly, when combined with the outputs from other windows, it forms a comprehensive representation fed into the classifier. This ensemble of temporal information improves the classifier's accuracy, allowing for more precise WiFi-enable HAR or sensing.

3.2.4 Encoder and Decoder Designs

The encoder block has a 5×5 head convolution at the beginning that extracts features from the input CSI matrix and fuses the information from different antennas. The data goes through a residual network with three asymmetric dilated convolution layers, each with a 3×3 kernel size, namely DConv blocks. In contrast to standard convolutions, the DConv layers implement dilated convolutions to enhance the receptive field of the convolutional layers without necessitating an increase in the kernel size, ensuring computational efficiency is maintained. Specifically, DConv extracts features at a particular interval, denoted as d or the dilation rate which results in a bigger receptive field. Each DConv layer in our model utilizes a unique dilation rate, d, allowing for varying receptive fields and a richer feature extraction from the input CSI matrix. These features are concatenated with a separate standard 3×3 convolution. After concatenation, a 1×1 convolution is used to reshape the data back to its original shape form before adding the initial input as residuals to the final output.

The decoder block does initial feature extraction using a 5×5 head convolution. The decoder employs two sequential dilated residual decoder blocks to recover the compressed information. The residual decoder blocks follow the design principle of the encoder, using two parallel branches and an identity map. The first branch uses a 3×3 dilated convolution with $d = 2$ to increase the feature dimension based on ρ, a flexible expansion rate hyperparameter, which can be adjusted according to devices with varying computational capacities. 3×1 and 1×3 convolution layers with dilation of 3 and out channel size of $c = 3\rho$, where ρ is the expansion rate of the model, are used. A 3×3 convolution layer is used to reduce the channel dimension back to

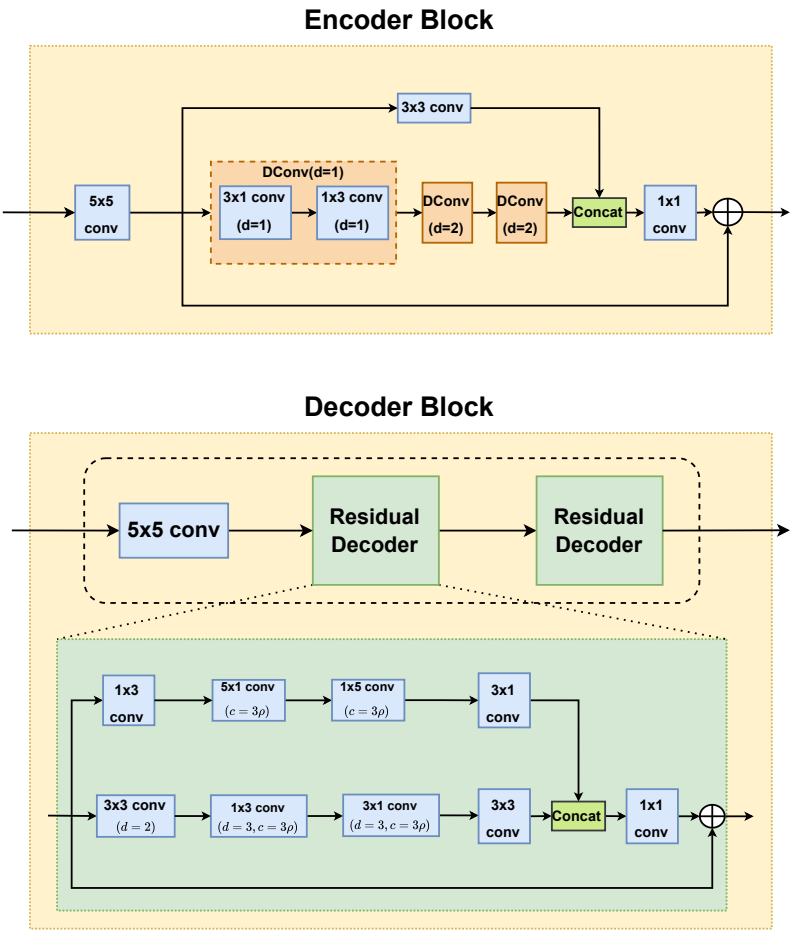


Figure 3.3: Detailed illustration of RSCNet encoder and decoder block designs

N_a. In the second branch, convolution layers filters of sizes 1×3 and 3×1 are used to increase and reduce feature dimensions at the beginning and end of the branch, respectively. 5×1 and 1×5 convolutions follow a similar out channel size as in the first branch, but without any dilation. Similar to the encoder, the output of these branches is concatenated and a 1×1 convolution is used to reshape the data back to its original dimensions.

3.2.5 Multi-task Learning for HAR and CSI Reconstruction

Within the architecture of RSCNet, the compressed CSI output from the encoder plays a dual role: it is used both for CSI reconstruction and enabling HAR. To ensure that this compressed representation is both a faithful representation of the original CSI and retains the necessary discriminative features for HAR, a unique loss function, $\mathcal{L}$, is employed:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_r \tag{3.1}$$

Here, $\mathcal{L}_r$ represents the Mean Square Error (MSE) which captures the error in CSI reconstruction. It's mathematically expressed in Equation 3.2:

$$\mathcal{L}_r = \mathbb{E}\left\{ \left\| \boldsymbol{H} - \hat{\boldsymbol{H}} \right\|_2^2 \right\} \tag{3.2}$$

In the above, $\boldsymbol{H}$ represents the original CSI, while $\hat{\boldsymbol{H}}$ denotes its reconstructed counterpart. The MSE seeks to minimize the differences between these two matrices, aiming for accurate CSI reconstruction with minimal error.

On the other hand, $\mathcal{L}_c$ signifies the Cross-Entropy loss for HAR, as detailed in Equation 3.3:

$$\mathcal{L}_c(x, y) = -\mathbb{E}_{(x,y)} \sum_{i=1}^{C} \left[\mathbb{I}[y = i] \log \left(\sigma \left(\hat{y}_i(x) \right) \right) \right] \tag{3.3}$$

In this formulation, $\mathbb{I}[y = i]$ is an indicator function that returns 1 if the label y is equal to class i and 0 otherwise. σ is the softmax function, and $\hat{y}_i(x)$ provides the predicted probability of class i for a given input x by the classifier. The Cross-Entropy loss calculates the difference between the predicted probabilities and actual class labels, with a focus on optimizing sensing accuracy.

The coefficient λ in the combined loss function serves as a weight factor, balancing the scale of two loss functions. It ensures that while the compressed CSI is a robust representative of the original data, it also retains the nuances necessary for precise HAR or sensing.

3.3 Experimental Settings

In this section, we explain the data set-up followed by the settings for model training and evaluation criteria. Then, we present NMSE, HAR accuracy, and FLOPs count of RSCNet as criteria for comparing our results with the benchmark SenseFi [2].

3.3.1 Data Setup

In order to validate the efficiency of the RSCNet framework, we utilized the UT-HAR dataset [1], collected by the University of Toronto. This dataset, gathered using Intel 5300 NIC [52], comprises 3 pairs of antennas and 30 subcarriers per pair. Given that the UT-HAR dataset originally consisted of continuous CSI data, we opt for a segmented version offered by [2], containing approximately 5000 samples, each with 250 frames of CSI matrices. This HAR dataset encompasses 7 distinct categories for human activity: lie down, fall, walk, run, sit down, stand up, and empty room. For the purpose of our experiments, we divide the dataset into training, validation, and testing sets, consisting of 3977, 496, and 500 samples, respectively.

3.3.2 Training Setting

We implement our model using the PyTorch framework. The model is optimized using the Stochastic Gradient Descent (SGD) with a learning rate of 0.01, momentum of

0.9, and weight decay of 1.5×10^6. For learning rate adjustment during training, we utilize the Cosine Annealing schedule for learning rate adjustments. The training process is conducted with a batch size of 512 across 300 epochs. Additionally, we note that the scales of $\mathcal{L}_c$ and $\mathcal{L}_r$ become similar and yield the best classification accuracy and reconstruction error when using $\lambda = 50$. The expansion rate, used for the decoder block is $\rho = 1$ unless specified otherwise. Finally, the classifier utilized for HAR has two hidden layers containing 512 and 128 nodes.

3.3.3 Evaluation Criterion

To assess the sensing capabilities of the framework, we evaluate the recognition accuracy on the test dataset. The compression performance is ascertained using the Normalized Mean Square Error (NMSE), which is reported in dB and defined as follows:

$$\text{NMSE} = \mathbb{E}\left\{ \frac{\left\| \boldsymbol{H} - \hat{\boldsymbol{H}} \right\|_2^2}{\|\boldsymbol{H}\|_2^2} \right\} \tag{3.4}$$

To further comprehend our model's efficiency, particularly in resource-limited scenarios, we assessed its complexity by computing FLOP counts, underscoring the model's suitability for resource-constrained IoT devices and cloud servers.

3.4 Results and Discussions

3.4.1 Choice of Number of CSI Frames N_f

RSCNet gives flexibility in choosing the number of CSI frames in a window, N_f, for compression and transmission to the cloud. The selection of N_f is an important decision as it can affect the complexity of the encoder and the decoder, the frequency of CSI transmission, and the HAR performance. In the results depicted in Table 3.1, we observe that by increasing the number of frames, N_f, from 5 to 50, there's an overall improvement in HAR accuracy, peaking at 97.2%. However, when extending the N_f further to a maximum of 250 which is the size of the time dimension, N_t, the HAR accuracy reduces and settles at 95%. It becomes evident that neither a minimal nor an exceedingly large N_f is optimal. While the former might lack capturing sufficient temporal features required for sensing recognition, the latter reduces the number of windows or sequence length which impacts the power of recurrent block to capture temporal information.

Table 3.1: Comparing NMSE and accuracy across different frame counts with $\eta = 1/90$ compression

N_f	NMSE (dB)	Accuracy (%)
5	<u>-23.212</u>	91.80
10	**-23.915**	95.80
25	-22.373	95.60
50	-22.759	**97.20**
125	-21.881	<u>96.20</u>
250	-20.160	95.00

3.4.2 FLOPs vs Number of CSI Frames N_f

The complexity of the network is also influenced by N_f, as illustrated in Fig. 3.4. Given that the data shape remains consistent across most network layers, an increase in the number of frames can amplify the encoder's complexity by up to 150 times and the decoder's by up to 50 times. However, the classifier complexity follows a reverse pattern as its number of parameters depends on the sequence length which decreases as the number of CSI frames increases. Thus, opting for a smaller window size can enhance the encoder's efficiency, but it will increase the classifier complexity in the cloud.

Furthermore, as N_f increases in a window, both the size of the CSI data and its compressed version expand, concurrently leading to a reduced frequency of data transmission but with higher overhead. This dynamic highlights the trade-off RSCNet faces between window size and transmission frequency. Therefore, an improper N_f selection can entail transmission overheads, either as frequent transmissions for smaller N_f or as data-heavy, infrequent bursts for larger ones. Due to the balance achieved at N_f =50 frames, in terms of HAR accuracy, NMSE, and complexity, we chose this frame count for subsequent tests. This selection underscores the need to optimize N_f to ensure efficient data compression, optimal transmission, and sustained HAR accuracy.

The primary objective of RSCNet is to minimize the computational demands of WiFi devices without compromising sensing performance. Therefore, we compare the complexity of the RSCNet encoder, which is deployed on these devices, to two leading SenseFi models [2], CNN-5 and ResNet18, which serve as our baselines and are only deployable on-device. Our findings show that our encoder achieves a reduction in

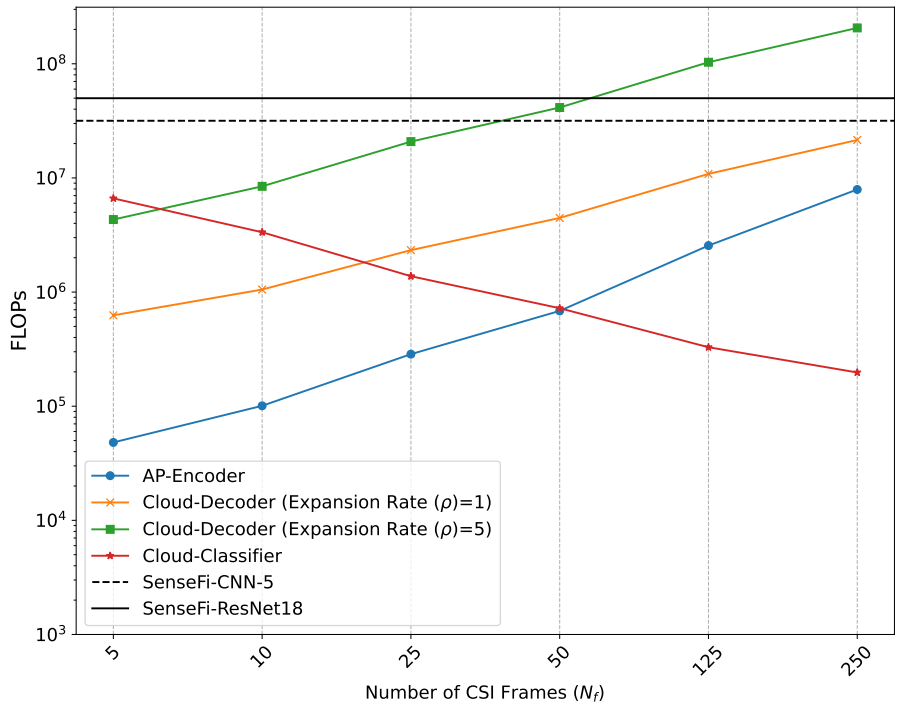


Figure 3.4: Comparative FLOP counts analysis for different CSI frame numbers with $\eta = 1/90$ compression and the Flop count of the baseline methods.

FLOP counts ranging from 77.9% to 99.7% on WiFi devices depending on N_f. This substantial reduction underscores the efficiency of RSCNet and demonstrates the potential of cloud-based sensing as an effective solution for enabling WiFi sensing on low-cost and low-power devices.

Compression Ratio vs NMSE and HAR Accuracy

The compression ratio, η, is primarily contingent on the data overhead the communication channel with the cloud can accommodate. Nonetheless, it is crucial to optimize η to ensure a reasonable HAR accuracy and CSI reconstruction performance (NMSE).

Fig. 3.5 displays the RSCNet framework with different compression ratios for the decoder. The HAR accuracy remains competitive with the SenseFi benchmarks, trailing by a mere 1-2%, except at the most extreme compression ratio of $\eta = 1/4500$, which reduces a 50-frame CSI window to a single value. This showcases that utilization of RSCNet which has much lower computational requirements on-device results in a comparable performance to fully on-device sensing models. Moreover, CSI reconstruction remains consistent across different η values, indicating the NMSE's reliability in various settings.

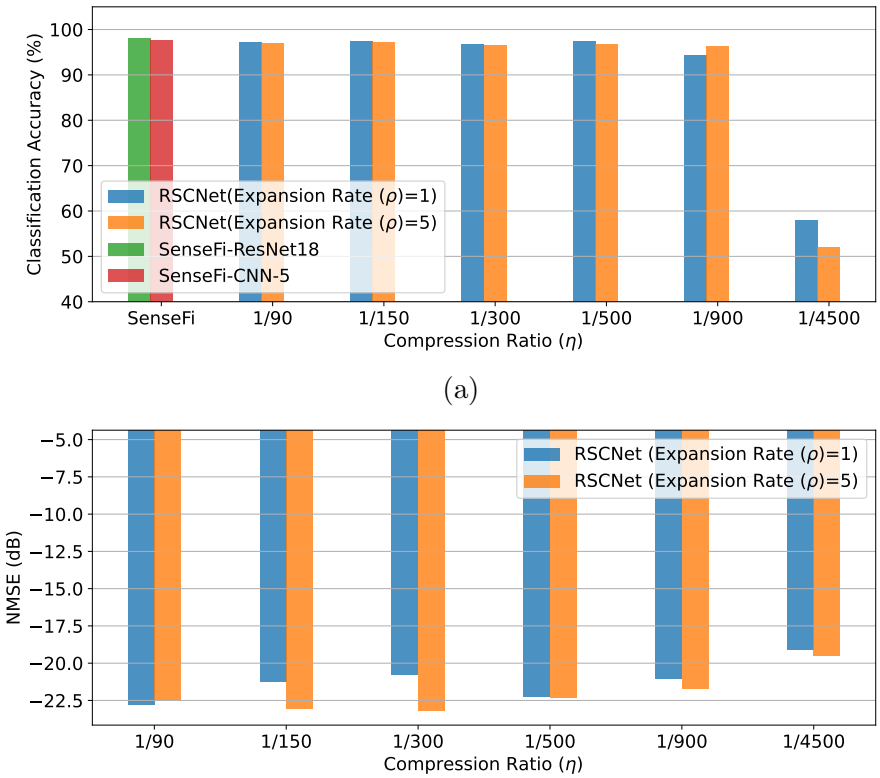
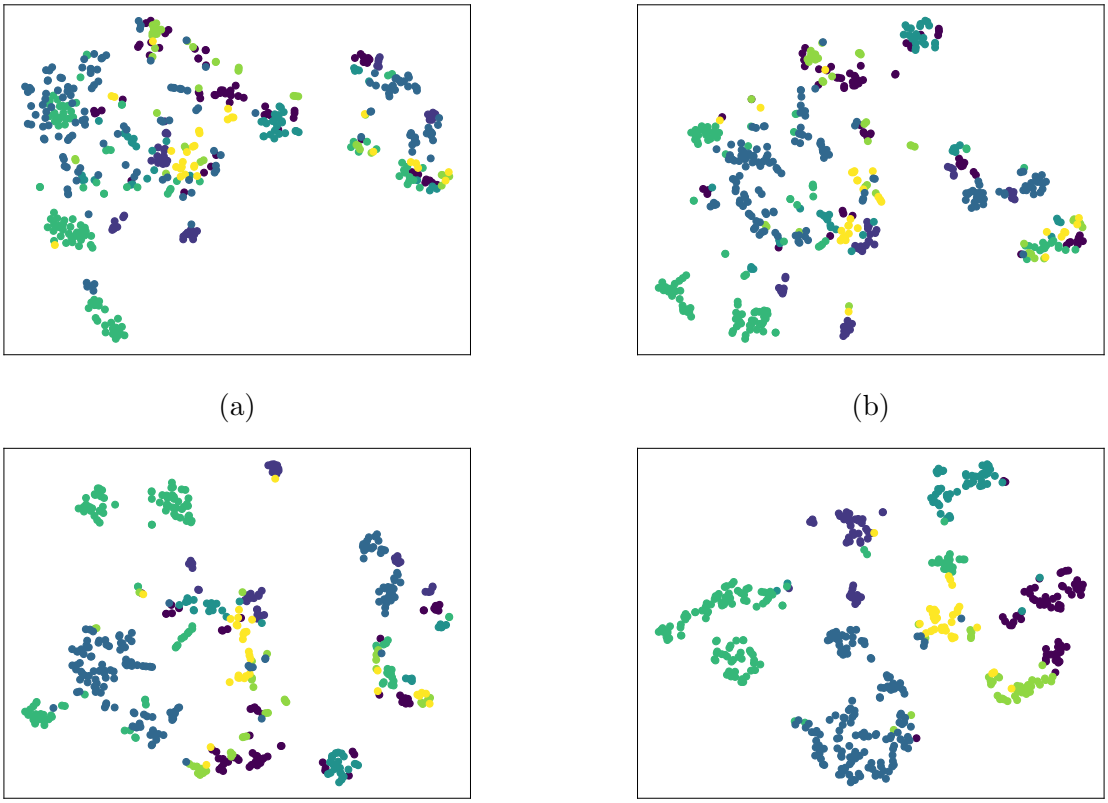


Figure 3.5: Comparing performance metrics across RSCNet with different compression ratios and expansion rates for $N_f = 50$ as well as baseline methods, with (a) illustrating sensing performance and (b) showcasing reconstruction error.

3.4.3 Expansion Rate vs NMSE

Fig. 3.5 displays the RSCNet framework with expansion rates $\rho = 1, 5$ for the decoder. The RSCNet's reconstruction error is intrinsically linked to its decoder's expansion ratio, ρ. As ρ rises, the decoder's convolutional layers broaden their channel dimensions, boosting the CSI data reconstruction ability. As depicted in Fig. 3.4, the increased performance of the decoder is counteracted by heightened computational requirements. For instance, a decoder at $\rho = 5$ can demand up to tenfold the computational resources than its $\rho = 1$ counterpart, contingent on N_f. Still, varying expansion rates offer a trade-off between reconstruction performance and computational efficiency, furnishing multiple deployment options for the decoder based on the application and the cloud service's resources. However, it is noteworthy that the encoder has a significantly lower computational cost than the decoder which makes it optimized for development on resource-limited devices.

3.4.4 T-SNE Analysis



(a) (b)

(c) (d)

Figure 3.6: Visualization via t-SNE for parameter settings $\eta = 1/500$ and $N_f = 50$ frames. (a) Initial raw CSI representation; (b) Compressed CSI embedding; (c) LSTM layer output embedding; (d) Final layer embedding within the classifier

In Fig. 3.6, we present a t-SNE visualization of the CSI at various network phases, differentiated by the labels of corresponding activities. The visualization of raw CSI is illustrated in Fig. 3.6a. Although there is a discernible separation among different activity types, the majority of samples appear dispersed throughout the embedding space. Conversely, Fig. 3.6b showcases the spatial separation of the compressed CSI, which is relayed to the cloud server. Notably, there is an enhanced distinction in

the feature space. This indicates that the encoder serves a dual purpose: facilitating CSI reconstruction and being effectively discriminative for the HAR task. Despite the refined spatial delineation, overlaps across certain classes within the compressed CSI remain apparent. Advancing to the LSTM layer's representation in Fig. 3.6c, a heightened discrimination towards class labels is evident, reflecting the LSTM's capability in capturing temporal features necessary for HAR. In our final representation, we delineate clusters of CSI in the classifier's concluding layer, revealing a distinct clustering contingent upon the activities. In summation, our visualizations highlight the encoder's role in both data compression and task-relevant feature discrimination, while also validating the LSTM block's effectiveness in temporal feature extraction. The ensuing representations manifest discernible, activity-centric clusterings, thereby substantiating our methodological propositions.

3.5 Summary

This work proposed RSCNet framework which facilitates real-time compression and sensing through adaptable small CSI windows. RSCNet incorporates an LSTM block to enhance accuracy and reconstruction by using previous CSI windows information. Our evaluations consistently underline RSCNet's effectiveness, highlighting its compatibility with different CSI window sizes, and compression ratios, and comparing its performance with state-of-the-art counterparts.

Chapter 4

CAPC: A Representation Learning Framework for WiFi Sensing

4.1 Introduction

The scarcity and challenge of obtaining labeled CSI data necessitate innovative approaches that leverage unlabeled data for effective model training. As mentioned before, SSL is a effective method for utilizing such data and has led to significant advances in developing robust and adaptable WiFi sensing systems. Traditional SSL approaches, primarily designed for image processing, are often not directly applicable to the temporal and contextual nature of CSI data. To address this gap, this chapter introduces a novel SSL framework specifically tailored for WiFi sensing, named **C**ontext-**A**ware **P**redictive **C**oding (**CAPC**). This framework integrates contrastive and non-contrastive learning mechanisms to harness the rich temporal dynamics and contextual information of CSI data, thereby enhancing the model's ability to generalize across various environmental conditions and reduce the amount of required labelled data. Additionally, we propose a novel augmentation technique, *dual view*,

which utilizes reciprocal links of wireless transceivers to isolate free space propagation from electronic distortions, thus enhancing the model's ability to capture the core characteristics of the over-the-air channel which contains the sensing information. The Figure 4.1 provides a summary of the CAPC framework.

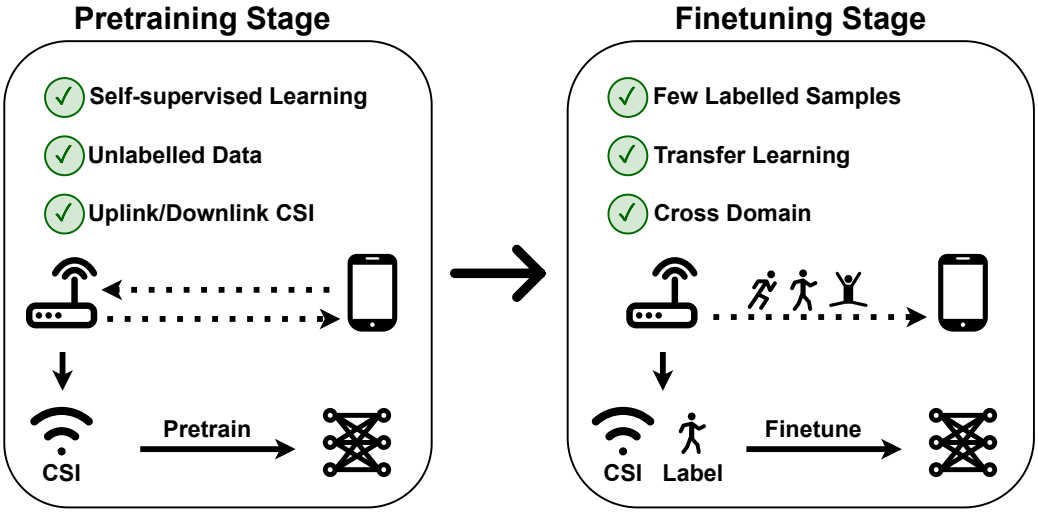


Figure 4.1: Illustration of the proposed CAPC framework. Initially, unlabelled uplink and downlink CSI are utilized to pretrain an encoder through an unsupervised approach. Subsequently, the model undergoes finetuning using a limited set of labelled CSI for the HAR task in the unseen environment.

CAPC uniquely combines the principles of Barlow Twins [51] and Contrastive Predictive Coding (CPC) [50] to create a twin-branch design. Each branch processes distorted versions of CSI samples, augmented using several augmentation methods including a novel technique that leverages both uplink and downlink CSI. These samples are then segmented into smaller windows. An encoder generates latent representations for each window and is followed by an autoregressive model that creates context embeddings by summarizing multiple windows' latent representations and predicts

future windows using CPC loss functions to capture temporal dynamics. Simultaneously, the Barlow Twins' non-contrastive loss function maximizes the agreement between the context embeddings across both branches with WiFi sensing specific augmentations. This includes enhancing the sensing context consistency of uplink and downlink CSI representations by extracting free space propagation effects, minimizing electronic distortions, and CSI estimation errors. This hybrid loss function ensures the encoder extracts essential temporal information and remains invariant to augmentations and distortions, thus making the representations robust and feature-rich.

Our key contributions can be summarized as follows:

1. We develop a novel SSL framework, CAPC, along with considerations for time-series and wireless propagation channel properties. CAPC utilizes a contrastive predictive method to capture temporal dynamics and a non-contrastive method for invariant, contextually consistent representations.

2. We introduce the dual view augmentation method that leverages both uplink and downlink CSI, enhancing the model's generalization capabilities by isolating free space propagation effects and minimizing electronic distortions.

3. We perform a quantitative comparative analysis of several common augmentations for time-series data as well as the proposed one to find the best combination suited for WiFi sensing and different SSL methods.

4. We demonstrate that CAPC requires fewer labelled samples while achieving superior performance compared to other SSL baselines, with an average improvement margin of 6.5%. Additionally, CAPC outperforms supervised learning by

an average margin of 30.53% in low-labelled data scenarios. These results are validated under both linear and semi-supervised evaluation in unseen environments using the SignFi dataset [19].

5. We evaluate the transfer learning performance of CAPC on the UT HAR dataset [1], where CAPC representations exhibit superior generalization to the new HAR task in an unseen environment. CAPC outperforms SSL baselines by 1.8% and supervised learning by 24.7% in accuracy, further showcasing its cross-domain capabilities.

6. We have made CAPC's source code available on GitHub at `https://github.com/bornabr/CAPC`.

4.2 Method

In this section, we describe the components of our novel CAPC framework designed to improve CSI-based WiFi sensing as well as the novel model-based augmentation.

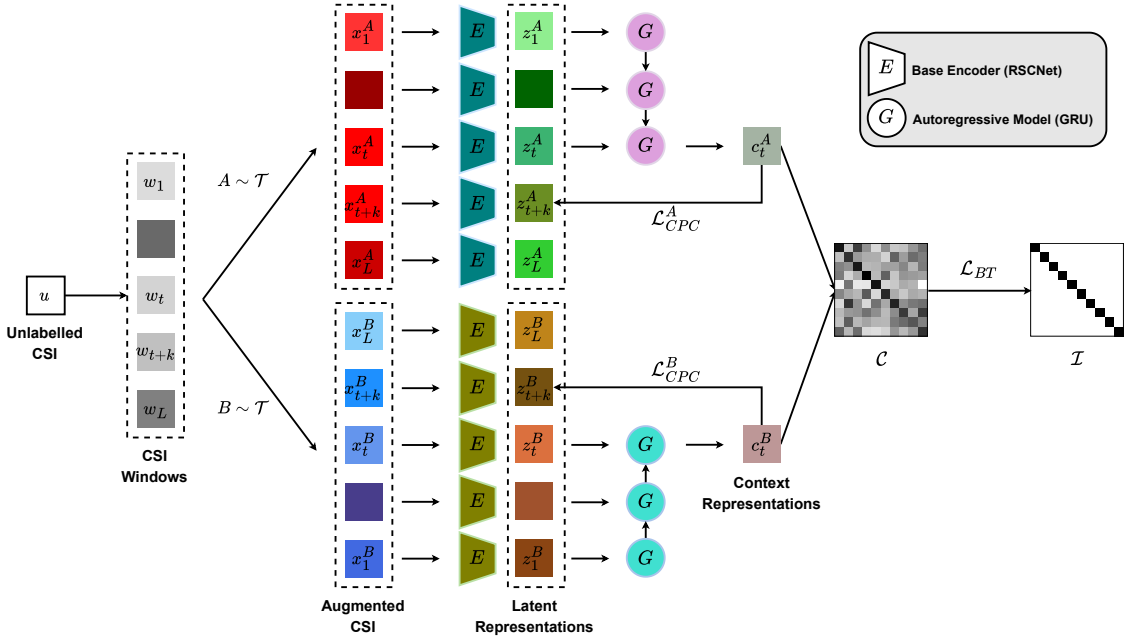


Figure 4.2: Overview of the CAPC's architecture. Here, w_t denotes a window of sample u. The symbols x_t, Z_t, and c_t represent the augmented CSI for window t, the latent representation of this window, and the accumulated context embedding up to window t, respectively. Different colours signify distinction in the windows, their representations, and model parameters between branches A and B.

4.2.1 Overview

An overview of the components of our proposed CAPC framework is given as follows:

- **CSI segmentation:** Each CSI sample u_i in the unlabeled batch U has dimensions $N_a \times N_s \times N_t$, where N_a denotes the number of antennas, N_s denotes the number of subcarriers, and N_t denotes the number of packets or timestamps. Each sample u_i undergoes a segmentation process, which results in a set of CSI windows $\{w_1^i, \ldots, w_L^i\}$. Each window w_t^i has dimensions $N_a \times N_s \times N_f$, where N_f represents the number of CSI frames in each window. The total number of windows L generated from each sample is calculated by $L = \frac{N_t}{N_f}$.

• **Stochastic augmentations:** In this phase, each CSI window w_t undergoes random transformations through a customized suite of augmentations designed specifically for CSI time-series data including *Gaussian noise*, *time flip*, *time mask*, *subcarrier mask*, and a novel augmentation, named *dual view*, proposed specifically for wireless sensing. This sequence of augmentations represented as $\mathcal{T}$ yields two unique distorted versions of the data sample, denoted as x_t^A and x_t^B.

• **Latent representation generation using base encoder:** A base encoder, $E_\theta(x)$ is employed to generate a sequence of latent representations from the windows. Each of the two augmented views is processed by two separate encoders with their respective learning parameters θ^A and θ^B. This yields latent representations $z_t^A = E_{\theta^A}(x_t^A)$ and $z_t^B = E_{\theta^B}(x_t^B)$, where $z_t^A, z_t^B \in \mathbb{R}^D$ represent the encoder's output in the embedding space of dimension D. The main objective of CAPC is to make the representations z to be as feature-rich and robust as possible as the base encoder and these representations are used in the downstream tasks.

• **Context embedding generation using an autoregressive model:** Following the generation of latent representations, an autoregressive model, G, is applied to condense the sequence of latent representations into a singular context embedding for each augmented view. Specifically, $c_t^A = G_{\gamma^A}(z_{\leq t}^A)$ and $c_t^B = G_{\gamma^B}(z_{\leq t}^B)$ are obtained, where both c_t^A and c_t^B reside in the $\mathbb{R}^H$ space. This space denotes the transformed embedding dimension utilized by the model's loss function. The parameters γ^A and γ^B represent the distinct learning parameters for the autoregressive model's head, corresponding to each view.

• **Hybrid contrastive loss function:** CAPC's core innovation is a hybrid contrastive loss function that combines the CPC ($\mathcal{L}_{CPC}$) [50] and Barlow Twins ($\mathcal{L}_{BT}$)

[51] loss functions. The CPC loss predicts future latent representations from context embeddings c_t^A and c_t^B, independently for each branch, enforcing the model to learn the underlying shared information between the windows. Meanwhile, $\mathcal{L}_{BT}$ enhances consistency between c_t^A and c_t^B, ensuring robustness against augmentations and preventing dimensional collapse by reducing redundancy within embeddings.

Remark: Note that we use the RSCNet encoder [86] which already includes components like CSI segmentation and an autoregressive model within our framework. However, we substituted the LSTM in the autoregressive block with a GRU [111] due to its efficiency over LSTM. Alternatively, any standard encoder and autoregressive model, such as Transformers [87], could be employed, potentially enhancing performance with extra computational resources.

4.2.2 Proposed Model-based Augmentation for Wireless Sensing

Data augmentation is crucial for the effectiveness of many SSL techniques [49]. These methods aim to create distorted views of the same sample and then maximize the agreement between the representations of these views, making the design of appropriate data augmentations essential. Typically, these augmentations are distortions that naturally occur in the input data but do not alter the inherent semantics of the data, namely their label in the downstream task. By applying these augmentations, we force the encoder to stay invariant to these distortions and to learn the fundamental characteristics—specifically, the useful features of the downstream task that are consistent across both views.

We propose a novel augmentation technique for SSL in wireless sensing, termed as *dual view*. This method leverages reciprocal links of wireless transceivers to isolate the

free space propagation channel from electronic distortions inherent in the transmission and reception processes. By randomly assigning uplink and downlink CSI data from these reciprocal links to two branches of the network, our technique aims to extract the core characteristics of the over-the-air channel.

In classical electromagnetics, the reciprocity theorem [112] asserts that the communication channel between the transmitter and receiver is the same in both directions, regardless of the signal's direction. However, applying this concept straightforwardly may miss a critical subtlety: the channel's reciprocity pertains only to the antenna-to-antenna interaction. Specifically, the theorem addresses the signal's behaviour once it enters the free space medium. Nevertheless, the CSI measured by standard WiFi devices does not solely reflect the free space channel. Instead, the measured CSI at either end, is the measurement of the channel, all the way from the digital representation of subcarriers in the transmitter, to their digital representation at the receiver. This includes the entire analog medium traversed by the signal, including base-band (low frequency) amplifiers, oscillators, mixers, (HF) power amplifiers, front-end filters, duplexers, matching networks etc. Thus, the signal path includes a lot of electronics processing inside the transmitter, and similarly at the receiver, where the signal again passes through a front-end filter, low noise amplifier (LNA), mixer, base-band channelizer filter, and baseband amplifiers. These electronics involved in the transmission/reception chain has a non-trivial effect on the signal. These electronics significantly alter the signal, influencing the CSI calculation, which combines the effects of transmitter electronics, free space propagation, and receiver electronics. However, to study phenomena such as sensing or HAR, which only affects the free space propagation channel, it is essential to isolate this segment from

the influences of transmitter and receiver effects.

An answer arises by generating a latent representation of the channel that discounts the electronic effects by maximizing the information between the uplink and downlink CSI using a SSL method. The rationale is that the impact of free space propagation represents the common information in the uplink and downlink CSI, and their discrepancies are due to electronic distortions. Therefore, by maximizing information between these two branches, the encoders are forced to overlook the artifacts attributable to the electronics processing chain and instead generate representations that are closely tied to the variations in the over-the-air channel, which are crucial for sensing applications. In Section 4.4, we demonstrate how this augmentation, coupled with our innovative loss function, achieves such representations highlighted by our superior performance in downstream sensing tasks.

4.2.3 Hybrid Contrastive Loss Function

We present a new hybrid contrastive loss function that enhances the encoders and autoregressive models within each branch. This is achieved by employing the CPC loss function for intra-branch temporal prediction and the Barlow Twins loss function to ensure inter-branch contextual consistency.

Temporal Prediction

Following the CPC methodology [50], at each iteration, we select a random timestamp $t \leq L - T$, where T is the number of future windows the model aims to predict based on the windows up to t. For each branch, given the augmented input sequence

x, we generate the latent representation $z_t = E_\theta(x_t)$ with potentially reduced dimensions. An autoregressive head G_γ then summarizes all $z_{\leq t}$ into a context latent representation $c_t = G_\gamma(z_{\leq t})$.

This context embedding is used to predict each future window $t+k$, where $k \leq T$. However, instead of making a direct prediction, we introduce a log-bilinear model f as a density ratio estimator to preserve the mutual information between x_{t+k} and c_t as:

$$f_k(x_{t+k}, c_t) = \exp(z_{t+k}^T W_k c_t), \tag{4.1}$$

where W_k is a linear layer predicting the future window $t + k$ via the transformation $W_k c_t$, with each W_k tailored for every step k. This linear transformation ensures the mapping aligns the dimensions of c_t with those of z_{t+k}.

The rationale for preferring implicit prediction over explicit approaches, such as mean squared error, is that explicit methods are often computationally intensive. They attempt to model complex relationships within windows x rather than extracting shared information between x and c. Conversely, the CPC method chooses to model the mutual information between the context embedding, which represents current information, and future latent representations through a non-linear mapping, denoted as $f(x, c)$. By maximizing the mutual information between the representations, we extract shared features inherent to both the current and future windows.

To maximize the mutual information and optimize the estimation f, we employ the InfoNCE contrastive loss function, which uses categorical cross-entropy to distinguish the positive sample—the correct prediction—from others within the context's latent distribution. For a given batch size N, let $X = \{x_{t+k}^1, \ldots, x_{t+k}^N\}$ represent the set of potential predictions at step k for each batch sample. The CPC loss is thus defined

as:

$$\mathcal{L}_{CPC} = -\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}_X \log \frac{f_k(x^i_{t+k}, c^i_{t_i})}{\sum_{x^j \in X} f_k(x^j, c^i_{t_i})} \tag{4.2}$$

The use of randomly selected current timestamp, denoted as t, for each batch aims to enhance the generalization capabilities of the representations. CPC is applied independently within each branch of the twin network but with the same current timestamp for both branches as well as the same transformation W, thereby predicting future latent representations z^A_{t+k} and z^B_{t+k} based on their respective context latent representations c^A_t and c^B_t, respectively, using the $\mathcal{L}_{CPC}$ loss.

Contextual Consistency

To further enhance representation robustness and reduce redundancy, the Barlow Twins loss function [51] is utilized to preserve the uniqueness of features while maintaining their invariant nature across different augmentations. The loss function is employed on the context latent representations c^A_t and c^B_t and is formulated as follows:

$$\mathcal{L}_{BT} = \underbrace{\sum_{i=1}^{H}(\mathcal{C}_{ii}-1)^2}_{\text{invariance term}} + \underbrace{\lambda \sum_{i,j=1}^{H} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \tag{4.3}$$

Here, λ is a positive constant that balances the importance of the two terms of the loss function, and $\mathcal{C}$ is the cross-correlation matrix computed between the outputs c^A and c^B of the two identical networks along the batch dimension. The cross-correlation matrix $\mathcal{C}$ is defined as:

$$\mathcal{C}_{i,j} = \frac{\sum_b c_i^{b,A} c_j^{b,B}}{\sqrt{\sum_b (c_i^{b,A})^2 \sum_b (c_j^{b,B})^2}} \tag{4.4}$$

In this equation, b indexes batch samples, and i, j indexes the feature embeddings of the context latent representations. The values of $\mathcal{C}$ range from -1 (perfect anti-correlation) to 1 (perfect correlation). The Barlow twins' invariance term aims to make the embedding invariant to any applied augmentations by setting the diagonal elements of the cross-correlation matrix to 1. Meanwhile, the redundancy reduction term seeks to decorrelate the various vector components of the embedding by setting the off-diagonal elements of the cross-correlation matrix to zero. This process of decorrelation minimizes redundancy among the output units, ensuring that they hold distinct information about the sample.

Contrary to the original Barlow Twins method, which incorporates a projection head to map the latent representations to a separate embedding space for loss function optimization, our approach applies the Barlow Twins loss directly to the context representations. In other words, the autoregressive model in our framework also serves as the projection map.

Hybrid Loss

Our hybrid loss function integrates these components, leveraging a hyperparameter β to align the scale of the CPC loss, $\mathcal{L}_{CPC}^{A}$ and $\mathcal{L}_{CPC}^{B}$, to the Barlow Twins loss $\mathcal{L}_{BT}$, resulting in a balanced and effective optimization criterion for our SSL framework:

$$\mathcal{L} = \mathcal{L}_{BT} + \beta(\mathcal{L}_{CPC}^{A} + \mathcal{L}_{CPC}^{B}) \tag{4.5}$$

Most SSL methods, including CAPC, extract valuable representations by minimizing the distances between embedding vectors of augmented samples. However,

without additional mechanisms, these methods could lead to a *complete collapse* solution, where the resultant representations become constant across different inputs. Furthermore, they may also experience *dimensional collapse*, where the embedding vectors are confined to a lower-dimensional subspace, failing to utilize the entire embedding space. In CAPC, several components aid in preventing both complete and dimensional collapse. The use of negative samples in the CPC loss function prevents the complete collapse by ensuring that negative samples are distanced from positive samples, thus introducing variance in the representations. Moreover, the redundancy reduction term of Barlow Twins not only prevents complete collapse, but also combats dimensional collapse by promoting the decorrelation of feature embeddings. This hybrid strategy effectively enhances the robustness and utility of the learned representations.

Algorithm 1: Pseudocode for CAPC

Input: Unlabeled CSI batch $U = \{u_i\}_1^N$
Output: Updated model parameters $\theta^A, \theta^B, \gamma^A, \gamma^B, W_{1\ldots k}$
while *step* $<$ *total steps* **do**
  Segment each input u to create CSI windows $\{w_t\}_1^L$;
  Apply augmentation $\mathcal{T}$ to each window w generating x^A, x^B;
  Select a random timestamp $t \leq L - T$;
  Extract latent representation $z^A = E_{\theta^A}(x^A)$ and $z^B = E_{\theta^B}(x^B)$ for $x_1, \ldots, x_{t+T}$;
  Summarize the representations of the first t windows, $c_t^A = G_{\gamma^A}(z_{\leq t}^A)$ and $c_t^B = G_{\gamma^B}(z_{\leq t}^B)$;
  Calculate the mutual information between the context prediction c_t^A or c_t^B, and each possible prediction x by $f_k(x, c_t) = \exp(E_\theta(x) W_k c_t)$;
  Update $\theta^A, \theta^B, \gamma^A, \gamma^B, W_{1\ldots k}$ by minimizing $\mathcal{L}_{BT} + \beta(\mathcal{L}_{CPC}^A + \mathcal{L}_{CPC}^B)$;
end

4.3 Experimental Settings

In this section, we present the considered dataset specifications, considered baselines for comparison, considered evaluation criteria, training configuration of the neural network architecture, and considered augmentations other than the proposed Dual view.

4.3.1 Datasets

SignFi

We employed the SignFi gesture recognition dataset [19], specifically designed for sign language gesture recognition tasks. This dataset features a significant volume of data instances but has a limited number of samples per class due to its extensive variety of sign language words (classes), totalling 276. The SignFi dataset, acquired through the Intel 5300 NIC [52], consists of 3 antennas, 30 subcarriers, and 200 packets per data instance, resulting in each instance possessing $3 \times 30 \times 200$ dimensions. Its key attributes include:

(1) Multiple environment setups: SignFi dataset comes from two different environments: a home and a lab. This variety helps us test how well our method adapts to new environments. We used the lab dataset as the unlabelled dataset for the SSL pretraining as it contained more samples and the home dataset with labels for the supervised evaluation.

(2) Dual view CSI: SignFi dataset includes synchronized uplink and downlink CSI for each sample in the dataset allowing us to utilize them for dual view augmentation in SSL pretraining. To the best of our knowledge, SignFi is the only database offering synchronized uplink and downlink CSI.

(3) Substantial Sample Volume: The SignFi dataset comprises 5520 instances from the lab environment (20 samples per class) and 2760 instances from the home environment (10 samples per class), making it a notably large dataset.

UT HAR

We also employed the UT HAR dataset [1] for evaluating transfer learning. Specifically, we tested the RSCNet backbone encoders, which were pretrained using unlabeled CSI data in the SignFi lab, on a subset of labelled samples from UT HAR. These experiments aimed to assess the effectiveness of the CAPC representations for adapting to new tasks and environments.

The UT HAR data was collected using the Intel 5300 NIC [52], similar to the setup used in SignFi. The data samples have dimensions of 3 antennas, 30 subcarriers, and 250 packets. Given that the antenna and subcarrier dimensions align with those of SignFi, and considering our use of a windowing size $N_f = 10$ in the encoders for both CAPC and all baseline models, we were able to use the same pretrained encoders from SignFi without requiring additional preprocessing. Additionally, this dataset categorizes human activities into seven types: lying down, falling, walking, running, sitting down, standing up, and empty room. The choice to use this dataset solely for fine-tuning and transfer learning purposes is due to its relatively small size, with only 3,977 samples in the training set.

4.3.2 Evaluation Criteria

Our evaluation criteria shown in Figure 4.3 are designed to assess how effectively a pretrained encoder with SSL adapts to downstream WiFi sensing tasks. Specifically,

we aim to understand the applicability of the encoder's representations when confronted with a limited number of labeled samples from the downstream task. We base our investigation on two main questions:

(1) Can the SSL encoder extract relevant features for the downstream task without additional adaptation? We analyze if the representations produced by the encoder are sufficient and generalizable for the downstream tasks without additional data-specific training. The encoder's weights, denoted as θ_A, are frozen to prevent the infusion of task-specific information. The latent representations, generated by the encoder for each data window, are concatenated and used as input for a linear classifier C_ϕ. This classifier is trained using the cross-entropy loss function, previously defined in Equation 3.3. This setup tests the hypothesis of how a well-trained encoder can enable a linear classifier to perform effectively on downstream tasks without the additional information of the task in the representations.

(2) Does partial fine-tuning of the encoder enhance its adaptability to the downstream tasks? We examine whether adjusting the weights of the encoder, θ_A, enhances its performance and adaptability to the specific characteristics of the downstream task. This creates a balance between leveraging the learned representations in the self-supervised mode and utilizing available labeled samples.

Unlike in linear evaluation where the encoder is completely frozen, in this **semi-supervised evaluation** setup, both the encoder and the classifier are trained, but the encoder is subjected to a lower learning rate compared to the classifier as well as fewer training epochs in general. This method allows the encoder to fine-tune its pre-learned representations to the specifics of the new task without significantly drifting from its original, generalizable features. Importantly, this approach helps

prevent catastrophic interference, where the model could otherwise forget previously learned information upon acquiring new, potentially biased information from the limited number of labeled samples in the downstream task. This strategy ensures a robust evaluation of the SSL method's effectiveness and adaptability.

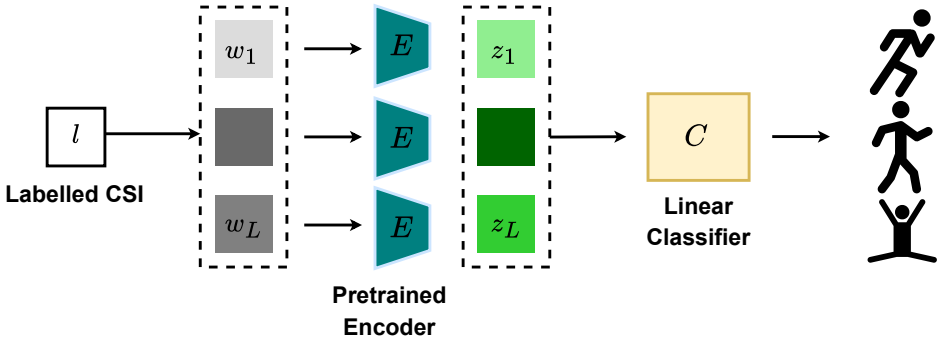


Figure 4.3: **Supervised evaluation:** A linear classifier C_ϕ is fine-tuned with labelled CSI based on the concatenated representations from all windows generated by the pretrained encoder E_{θ^A}. The pretrained encoder's weights θ^A are frozen in linear classification but fine-tuned in the semi-supervised evaluation.

4.3.3 Baselines

To thoroughly evaluate the effectiveness of our proposed method, CAPC, we conducted comparisons with four established SSL methods. Specifically, we compared CAPC against: *(1) SimCLR* [47], which serves as a standard contrastive SSL benchmark; *(2) Barlow Twins* [51] and *(3) CPC* [50], both included as ablation studies due to their integration within CAPC's framework; *(4) AutoFi* [41], a recent SSL method tailored for CSI WiFi sensing. Furthermore, we compare CAPC against a fully supervised training model where the encoder E_θ is randomly initialized and trained with the labelled data from scratch. This comparison is intended to highlight

the performance gains of SSL even with limited training labels.

Remark: To ensure a fair comparison, all methods utilized the same backbone encoder architecture, namely the RSCNet encoder [86]. Additionally, unlike CAPC and CPC, which employ an autoregressive model during pretraining, SimCLR, Barlow Twins, and AutoFi use a projector to independently map each window's representation into an embedding space for applying their respective loss functions. Specifically, we adopted the Barlow Twins' projector configuration, which consists of three fully connected layers paired with ReLU activation functions. By using the same backbone encoder across all methods, we maintained consistent model complexity, as detailed by the FLOP counts in Figure 4.8. This consistency ensures that any differences in representation quality are solely attributable to SSL methodology and augmentations rather than model complexity, thus guaranteeing a fair and balanced comparison among CAPC and the baseline methods.

4.3.4 Augmentations

In addition to the proposed dual view augmentation, we evaluated several common augmentations used in the context of time-series [113], [114] and WiFi sensing [43]:

• **Gaussian Noise:** introduces random Gaussian noise with zero mean and 0.1 standard deviation into data, simulating inherent noise in CSI, and enhancing model resilience.

• **Time Flip:** flips CSI samples along the time axis to accommodate palindromic activities, where time-reversed patterns represent the same activity.

• **Time Mask:** randomly masks a segment of the time dimension in each CSI window, varying its location. This teaches the model to predict missing temporal information,

simulating scenarios with temporal disruptions or signal losses.

• **Subcarrier Mask:** masks a set of subcarriers in each CSI sample, forcing the model to infer activities using the remaining subcarriers. This augmentation improves the handling of frequency-selective fading or interference and underscores different subcarriers' significance in activity recognition.

Choosing Best Augmentations for each Method: We conduct a comprehensive analysis of the aforementioned augmentations, both individually and in combination, on the performance of our method and the baselines. This allowed us to identify the optimal augmentation mix for each method and to demonstrate the robustness of each method in response to the various augmentations. The augmentations chosen for each method are in Figure 4.4. For CAPC, the chosen augmentations are dual view and noise. For SimCLR, it is time mask. The Barlow Twins use noise and time mask, while AutoFi employs noise and time flip. CPC relies solely on prediction for its SSL task, thus not using any augmentations.

4.3.5 Training Configuration

Our model, developed using PyTorch, employs the LARS optimizer during the SSL phase [115]. The training lasts for 300 epochs with a batch size of 128. We initialize the learning rate for weights at 0.2 and for biases and batch normalization parameters at 0.0048. The initial 10 epochs act as a warm-up period for the learning rate, which is then reduced following a cosine decay schedule [116]. The weight decay is configured at 1.5×10^{-6}. The trade-off parameter of Barlow Twins loss is set to $\lambda = 0.002$ and the trade-off parameter of CAPC β is set to 50 to scale the loss terms. These configurations largely follow those reported in [51]. In our model, unlike [51], the

weights between the twin networks are not shared to enhance performance. Regarding the number of future windows prediction T in CAPC and CPC, we chose 9 for CAPC and 2 for CPC, as these values show better performance for each method as shown in Figure 4.8.

During the evaluation phase, we switch to the Adam optimizer [117], maintaining a batch size of 512. For linear evaluation, the learning rate starts at 10^{-3} and follows a cosine decay schedule over 100 epochs, training only the linear classifier. In the semi-supervised evaluation, the configuration remains the same except the model is trained for only 20 epochs, and the encoder's learning rate is set at 5×10^{-3}, following a similar decay schedule.

For the model architecture, we use a window size $N_f = 10$, an encoder embedding size $d = 128$, and 128 nodes in the GRU autoregressive model's hidden layer, the projection hidden layers and the embedding size h. The number of nodes in the hidden layer of the linear classifier C_ϕ is half of its input size, which equals the embedding size d multiplied by the sequence length L.

4.4 Results and Discussions

In this section, we present the experiments conducted to validate our SSL framework by showcasing its superiority over the aforementioned baselines in different configurations as well as demonstrating the quality of our representations using t-SNE visualization.

4.4.1 Results under different labelled samples budgets

We conduct extensive experiments to evaluate the effectiveness of our proposed CAPC model across different portions of the labelled SignFi Home dataset. The evaluation involved fine-tuning the pretrained encoder using 2 to 12 samples per class, or *shots*. Table 4.1 presents the results of our fine-tuned CAPC model compared to baseline methods, for both linear and semi-supervised evaluations. Additionally, we include the results of fully supervised training for reference. Other than CAPC with noise and dual view augmentation, which proved to be the most effective combination, we also tested CAPC with noise and subcarrier mask augmentations. These experiments showcase the superiority of our proposed dual view augmentation and demonstrate that our model can also deliver strong performance with alternative transformations.

In terms of linear evaluation, our method consistently outperforms all other methods, achieving the highest average accuracy of 89.82%. In few-shot scenarios (2 and 4 shots), both versions of CAPC achieved the highest accuracies, with the dual view version reaching 65.67% and 88.50% for 2 and 4 shots respectively. The best baseline, SimCLR, achieved 59.6% and 82.25% for the same scenarios. For higher numbers of shots, the performance differences among all SSL methods, except CPC, become less pronounced, yet CAPC still records the highest accuracies in most scenarios,

Table 4.1: **Evaluation of representations fine-tuned on SignFi home** derived from pretrained CAPC and baseline SSL methods on: (1) linear classification using frozen representations; (2) semi-supervised classification with fine-tuned representations. Pretraining occurs in the lab environment of SignFi, and supervised evaluations are conducted in various fractions of the home environment. We report the accuracies (in %) of supervised evaluation, highlighting the best results in **bold** and the second-best results with underlining. All methods use RSCNet encoder thus having the same number of parameters and complexities.

		Shots						
Evaluation	Method	2	4	6	8	10	12	Avg.
Linear	Supervised	-	57.97	78.99	89.95	82.79	91.12	66.80
	SimCLR	59.6	82.25	92.57	95.02	96.01	**98.01**	87.24
	CPC	55.89	75.82	86.23	89.49	93.57	94.66	82.61
	BT†	54.08	76.00	92.66	92.39	95.29	96.92	84.56
	AutoFi	59.15	79.62	92.84	**95.92**	96.65	97.55	86.95
	CAPC*	63.41	85.51	92.48	95.38	96.56	97.83	88.53
	CAPC	**65.67**	**88.50**	**93.84**	95.83	**97.55**	97.55	**89.82**
Semi-Supervised	SimCLR	51.27	75.18	89.13	93.57	94.84	96.38	83.39
	CPC	46.01	67.03	78.62	87.59	91.03	94.47	77.46
	BT†	46.74	68.93	78.71	89.40	94.02	95.29	78.85
	AutoFi	46.56	72.19	86.50	92.93	96.11	96.47	81.79
	CAPC*	49.73	72.83	87.5	93.12	95.83	97.74	82.79
	CAPC	**57.52**	**82.52**	**92.57**	**96.47**	**97.19**	**97.92**	**87.36**

† Barlow Twins

* Refers to the CAPC model being trained with the second best combination of augmentations, noise and subcarrier mask, instead of noise and dual view.

falling short by less than 0.5% only in the 8 and 12 shots categories. Generally, CAPC with either augmentation combination outperforms other methods by approximately 2.6% in average accuracy. Furthermore, we observe that supervised training performs poorly across all shots when compared to SSL methods, particularly in few-shot scenarios. For instance, with just 2 shots, the supervised model fails to converge. Furthermore, the average accuracy of supervised training is approximately 23% lower

than that of CAPC.

In semi-supervised evaluations, CAPC with dual view augmentation distinctly surpasses all other SSL methods by a substantial margin of 4%. This augmentation also appears to enhance semi-supervised training significantly, indicating that the encoder weights from the CAPC with dual view are particularly well-suited for fine-tuning. In few-shot scenarios, the performance gap between our method and the top baseline, SimCLR, is even more pronounced, with about a 7% difference in the 2 and 4 shot scenarios. CAPC without dual view augmentation also exceeds all baselines except for SimCLR, which shows comparable performance with only a 0.6% gap.

Overall, the results underscore the critical role of leveraging both temporal dependencies and wireless channel characteristics for SSL, especially in few-shot scenarios.

4.4.2 Transfer learning for a different task

Table 4.2: **Transfer learning on UT HAR.** Evaluation of adapting SSL-pretrained RSCNet backbone encoders using CAPC and other SSL methods from SignFi's sign language detection task to the HAR task by linear evaluation on top of fixed representations with limited labelled data—10 and 20 shots, constituting about 1.8% and 3.5% of available samples, respectively.

	Shots		
Method	10	20	Avg.
Supervised	8.0	56.0	32.0
AutoFi	51.2	55.0	53.1
CPC	52.2	54.4	53.3
Barlow Twins	<u>52.8</u>	56.0	54.4
SimCLR	52.6	57.2	<u>54.9</u>
CAPC*	49.8	<u>57.4</u>	53.6
CAPC	**54.2**	**59.2**	**56.7**

Transfer learning involves adapting a pretrained model to new tasks across different domains, demonstrating the model's ability to generalize well beyond its original training context. We applied the same pretrained RSCNet backbone encoders of the previous part—originally trained on the unlabelled SignFi lab dataset for sign language gesture recognition—to the UT HAR dataset. For this new HAR task, we kept the encoders fixed and fine-tuned a linear classifier using only 10 and 20 shots subsets of UT HAR dataset.

Our results, presented in Table 4.2, highlight the superior performance of the CAPC approach over other SSL methods and traditional supervised learning for the HAR task. Notably, CAPC achieves comparable results to other baselines even without dual-view augmentation. However, integrating dual-view augmentation significantly boosts the model's generalizability and performance, surpassing the next best SSL method, SimCLR, by 1.8%, and achieving a 54.2% accuracy in the 10 shots scenario, where traditional supervised learning did not converge to a comparable performance level. These findings highlight the effectiveness of CAPC in cross-task and cross-environment WiFi sensing applications.

4.4.3 Augmentations Selection

The selection of appropriate augmentations is critical in SSL techniques, as these methods are highly sensitive to augmentation choices. While some studies have examined the impact of augmentations in computer vision [47], the exploration of effective augmentations for SSL that are well-suited to CSI data and sensing applications remains limited. Our study focuses on identifying suitable augmentations for both our CAPC method and established baselines. We explored five types of augmentations:

dual view, noise, time flip, time mask, and subcarrier mask. These were analyzed both in combination and individually, as demonstrated in Figure 4.4, and the average performance of each augmentation across all methods was summarized in Table 4.3.

Table 4.3: Summary of Figure 4.4: Presents the average accuracies achieved using various augmentations across different methods.

Method \ Aug.	**Dual View**	Time Mask	Subcarrier Mask	Time Flip	Noise	Avg.
CAPC	**91.16**	87.41	**89.67**	**89.11**	**91.89**	**89.85**
AutoFi	49.98	80.16	79.64	80.67	82.97	74.68
SimCLR	86.50	**89.33**	88.60	89.04	88.95	88.60
Barlow Twins	79.22	87.73	86.76	84.00	87.77	88.10
Avg.	76.72	86.16	86.17	85.71	**87.89**	84.53

Our comprehensive analysis reveals that noise augmentation consistently enhances model generalization by accommodating inherent data disturbances, making it the most effective augmentation on average. It was part of the best augmentation set for each method, except for SimCLR, which performed better with the time mask alone. In general, our CAPC method outperformed other approaches across all augmentations, except for time mask, where SimCLR showed superior performance. This is likely because the temporal prediction component of CAPC complicates the task when a time mask is also applied, hindering the model's ability to predict future windows.

Notably, CAPC excelled with the dual view augmentation, achieving an average accuracy of 91.16%, compared to 86.50% for SimCLR. This observation is significant as dual view augmentation generally underperformed across other baseline methods. CAPC's remarkable success with this augmentation indicates its unique capability to mitigate electronic distortions, a feature that was anticipated but not as effectively

harnessed by other methods such as AutoFi. The distinct advantage of CAPC with dual view augmentation underscores its potential to leverage complex augmentations that other models fail to utilize effectively.

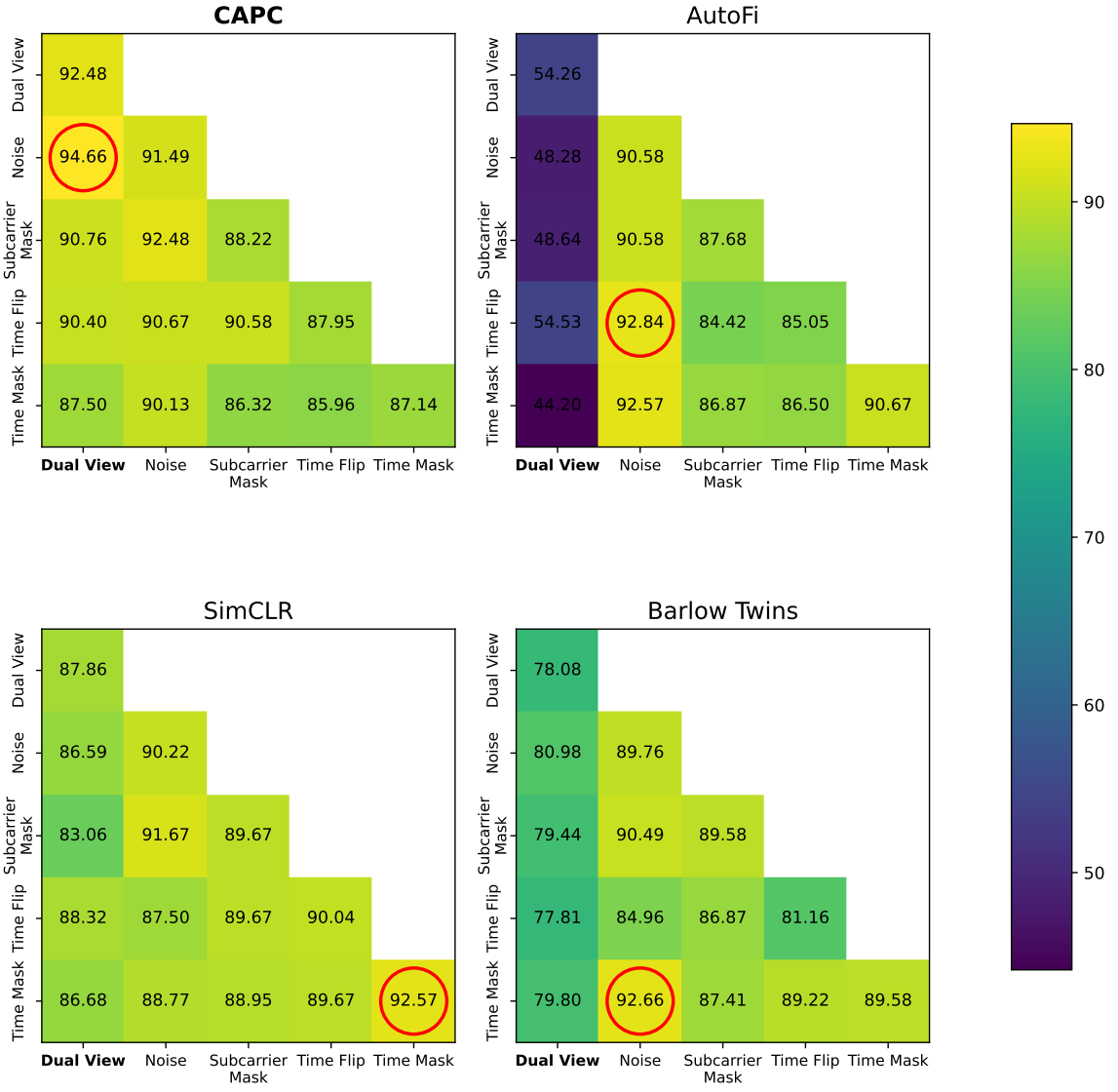


Figure 4.4: Linear evaluation of individual and compositional data augmentations. Each diagonal element represents the effect of a single transformation, while off-diagonal elements illustrate the combined impact of two sequentially applied transformations. We report the accuracies (in %) with 6 shots in the labelled dataset. Red circles indicate the best combination of augmentations.

4.4.4 Comparative Analysis of Contextual Loss

One of the key novelties of our CAPC method is the integration of hybrid temporal and contextual losses. The use of prediction and autoregressive models to capture temporal dependencies is well-recognized. However, the rationale behind choosing a specific type of contextual loss might not be immediately apparent as various contrastive [47] and non-contrastive methods [51], [100], [41] are available.

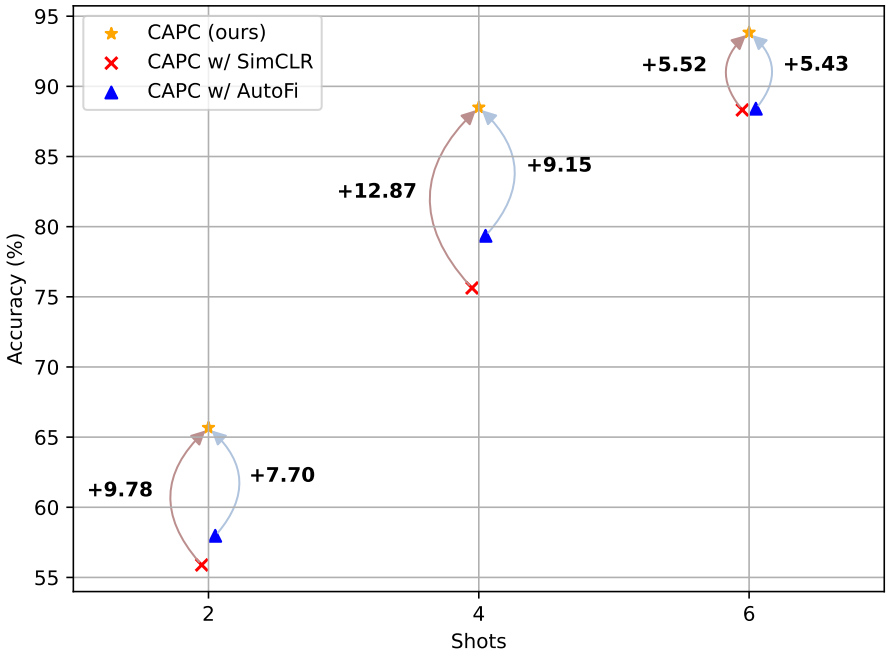


Figure 4.5: A comparative study of the proposed CAPC method. The CAPC w/ SimCLR and AutoFi mean that we have replaced the Barlow Twins loss function in our design with SimCLR and AutoFi, respectively. Showcasing that Barlow Twins has superior performance for enforcing context embedding consistency. We report the experiments under linear evaluation of SignFi Home dataset with 2, 4, and 6 shots.

Our investigation shown in Figure 4.5 included three distinct loss functions: Barlow Twins, as utilized in our framework, along with SimCLR, and AutoFi. Our

findings indicate that the non-contrastive loss from Barlow Twins consistently outperforms the contrastive loss used in CAPC with SimCLR as well as CAPC with the noncontrastive loss of AutoFi across various training scenarios, often by a significant margin. For example, in a scenario with four shots, CAPC equipped with Barlow Twins achieved an accuracy of 88.50%, markedly higher than the 79.35% and 75.63% seen with AutoFi and SimCLR, respectively. Additionally, AutoFi consistently outperforms SimCLR by an average of 2%. These results suggest that non-contrastive loss functions serve as more effective contextual losses for our proposed CAPC method.

4.4.5 Hyperparameter Selection and Sensitivity Analysis

We conducted sensitivity analyses on three primary hyperparameters within the CAPC framework: (1) we illustrate the impact of the number of future windows or timesteps, T, predicted by CAPC during SSL, on linear evaluation; (2) we demonstrate the effect of the trade-off parameter, β, which represents the weight of $\mathcal{L}_{CPC}$ in the loss function; (3) we depict the influence of the number of frames per CSI window, N_f, and encoder complexity comparing CAPC to baseline SSL methods.

The analysis in Figure 4.6 reveals the impact of T on the performance of CAPC and CPC. For both frameworks, we present the linear evaluation performance averaged across 2, 4, and 6 shot scenarios in SignFi Home. Both methods exhibit fluctuations; however, CAPC demonstrates greater robustness and fewer fluctuations across different T values, outperforming CPC by approximately 11.3% on average. Generally, CAPC achieved better results with higher T values, particularly at $T = 9$. Conversely, CPC's performance initially declined with increasing T values but improved upon reaching $T = 12$ and $T = 13$, with $T = 2$ being the optimal value. This

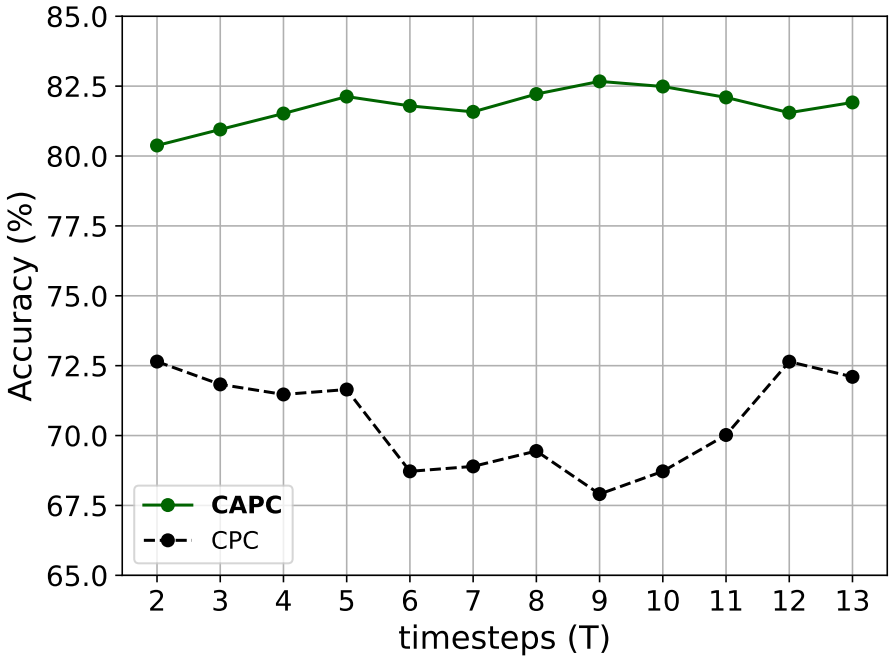


Figure 4.6: Illustrates how the accuracy of linear evaluation is affected by varying the number of predicted future windows (T) for 2, 4, and 6 samples per class, highlighting that CAPC is significantly more stable across different values of T compared to CPC.

implies that CAPC may be more effective at capturing common information over an extended number of windows.

We further investigated the sensitivity of CAPC to the hyperparameter β, which balances the significance of temporal and contextual consistency in the embeddings. We found that CAPC is relatively insensitive to variations in β, with values of 25 and 50 both demonstrating slightly improved performance overall, as illustrated in Figure 4.7.

We extensively examined SSL with different window sizes, N_f, for CAPC and all baselines within the flexible RSCNet framework, as shown in Figure 4.8. We specifically evaluated window sizes of 10, 20, and 50 for CAPC and CPC because larger

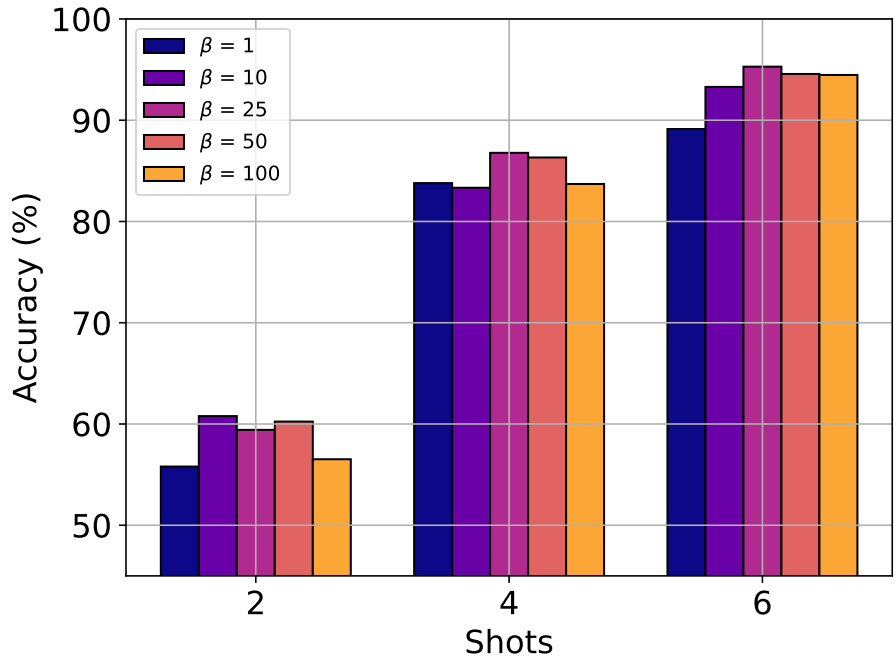


Figure 4.7: Depicts the influence of the coefficient β on CAPC's performance under linear evaluation.

N_f values reduce the sequence length L, and both CPC and CAPC require $L \geq 3$ to predict future windows effectively. For other methods, we extended our examination to include window sizes of 100 and 200, corresponding to the complete sequence of the CSI in SignFi. Our results indicate that our proposed CAPC, along with Barlow Twins and SimCLR, demonstrated robust performance across different N_f values, with CAPC showing superior accuracy in all tested cases. In contrast, AutoFi and CPC experienced significant performance declines as the number of windows increased.

This suggests that CSI segmentation is a viable approach for SSL in WiFi sensing, benefiting methods beyond our proposed model. Not only did all methods generally perform better with lower N_f values, but, as Figure 4.8 illustrates, these lower values

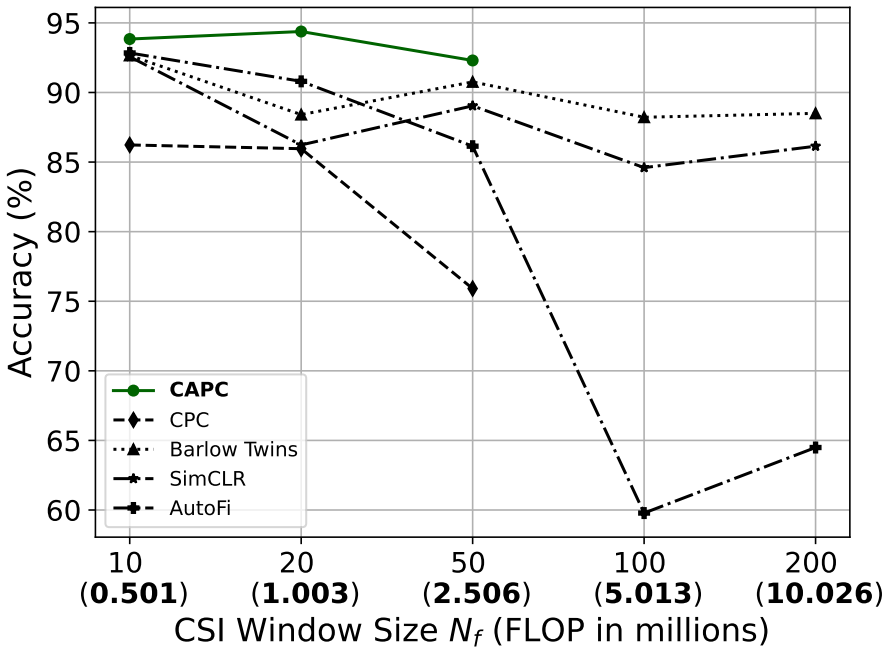


Figure 4.8: Examines the impact of different window sizes, N_f, on the accuracy of CAPC and baseline methods during linear evaluation for 6 samples per class. It also shows the computational complexity of the encoder with varying window sizes. Here, $T = 2$ for both CAPC and CPC due to constraints imposed by the limited number of windows at higher window sizes ($T \leq L - 2$).

also simplify the complexity of the encoder E due to smaller input sizes. This reduction enhances efficiency and reduces computational demands, making it particularly suitable for resource-limited edge devices where these models are deployed. Based on these findings, we selected $N_f = 10$ for subsequent experiments, as it generally yielded strong performance across all methods.

4.4.6 Collapse Analysis

To demonstrate that our method does not suffer from dimensional collapse, we present the singular value spectrum of the embedding space (Figure 4.9) of the pretrained

encoder of CAPC and other baselines. Complete collapse occurs when all singular values fall to zero, indicating that the representation has become constant. If only some singular values drop to zero, it suggests partial dimensional collapse, where the encoder has not fully utilized the embedding space [118]. As depicted in Figure 4.9, no singular values fall to zero for any of the methods, confirming that none, including ours, undergo dimensional collapse.

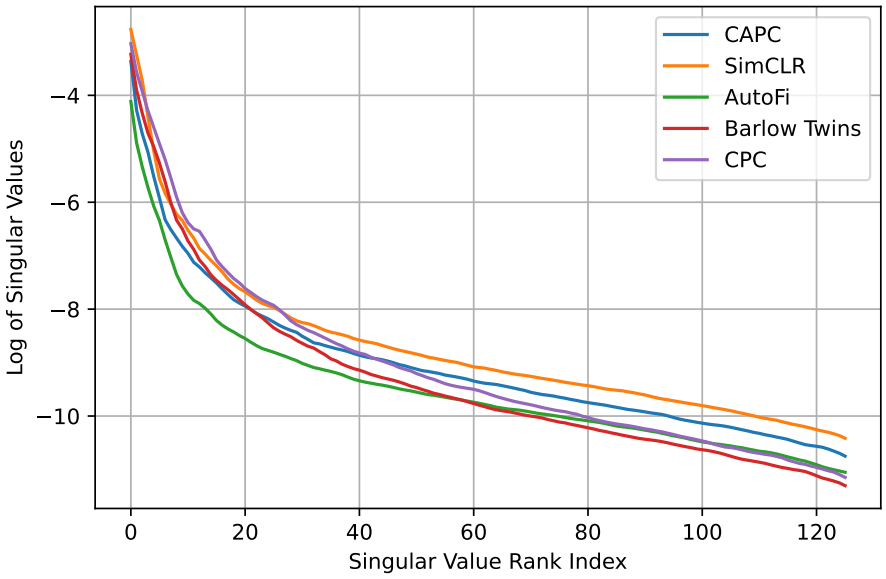


Figure 4.9: Singular value spectrum of the representation space (z) of CAPC compared to baselines on the SignFi Home dataset validation set. Each embedding vector is of size 128. The spectrum displays the singular values of the covariance matrix of these embedding vectors, sorted and plotted on a logarithmic scale. No singular values drop to zero, indicating that none of the methods, including ours, experience dimensional collapse.

4.4.7 T-SNE Visualization

In Figure 4.10, we present a t-SNE visualization [119] of the encoded CSI with the proposed CAPC pretrained encoder, marked by the colored labels in the SignFi Home

dataset. This visualization demonstrates the discriminative power of the embeddings for sign language recognition characteristics. Despite the large number of labels and the lack of supervision or predefined labels during training, the encoder effectively segregates the data into distinct clusters. Moreover, the input data originates from an unseen environment, further underscoring the capability of CAPC to not only extract relevant discriminative features for downstream tasks but also to generalize effectively in new environments.

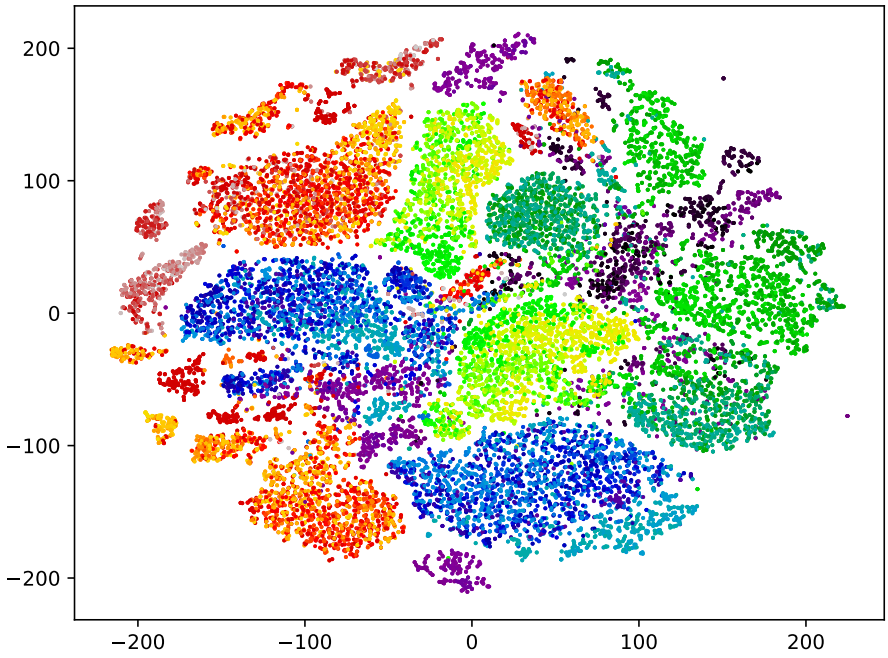


Figure 4.10: t-SNE visualization of SignFi Home dataset representations, trained using the CAPC SSL method on the SignFi Lab dataset. Each color corresponds to a distinct sign language label.

4.5 Summary

In this work, we proposed CAPC, a representation learning framework for CSI data and WiFi sensing, featuring a time-series-specific architecture. Specifically, we employed a hybrid contrastive loss function that combines future prediction and embedding consistency pretext tasks for SSL, ensuring the generated representations are both temporally informative and robust to data distortions inherent in downstream tasks. We also developed a novel augmentation technique to reduce electronic distortions from transceivers and isolate free space propagation effects on the channel. Extensive cross-domain experiments demonstrated that CAPC, with and without this augmentation, surpasses baseline SSL methods in downstream HAR tasks, particularly in few-shot scenarios. Furthermore, using the novel augmentation significantly boosts CAPC's performance.

Chapter 5

Conclusions and Future Directions

This chapter provides a conclusive summary of our work in Section 5.1 and discusses potential future developments and directions in the field of WiFi sensing in Section 5.2.

5.1 Conclusion

In this dissertation, we investigated WiFi sensing, which is the utilization of wireless WiFi signals captured which are transmitted and received throughout our everyday environments for sensing and monitoring physical characteristics of the environment, including human movements and activities. This study highlighted the challenges associated with the practical deployment of WiFi sensing solutions on a large scale. Specifically, we identified three primary obstacles that hinder widespread adoption:

1. The limited computational capabilities of WiFi devices constrain their ability to utilize complex deep learning systems required for sensing tasks.

2. The costs and complications associated with annotating CSI data make it difficult to train and adapt neural networks for sensing.

3. The bias and distribution shift of deep learning models towards the specific environment in which CSI training data was collected can limit the generalizability of these systems, particularly when applied to new, unseen domains.

We proposed two innovative representation learning frameworks—RSCNet and CAPC—to address these challenges. RSCNet provides a cloud-based neural network solution to overcome computational limitations, while CAPC introduces a self-supervised deep learning model that enhances generalization across different environments and reduces dependency on labeled CSI by utilizing unlabeled data. In the following, we provide a summary of each of the developed solutions.

Real-time Sensing and Compression Network

The Real-time Sensing and Compression Network (RSCNet) framework was designed as a two-part network. The first part of the system is deployed on WiFi access points such as smartphones, laptops, routers, and IoT devices. It consists of a lightweight encoder that compresses the CSI windows into a compressed representation while preserving the essential sensing features and information. This encoder achieves low complexity through the use of dilated convolutional layers, which offer a high receptive field while requiring fewer computational resources. This framework is proposed to tackle the limited computational power of WiFi devices. The compressed CSI is then transmitted to the cloud where the second part of RSCNet is deployed. The choice to compress the CSI instead of uploading raw data is due to the large dimensions of CSI and its transmission overhead. Upon receiving the compressed CSI window in the cloud, it passes through an LSTM autoregressive unit, which further enhances it by utilizing the temporal information captured from previous windows. Subsequently,

the representation of the window is processed by two systems: (1) a classifier that aggregates every window's representation to perform the sensing HAR task, and (2) a decoder that is used to restore the CSI for archival and logging purposes. The decoder is designed with the flexibility to adapt based on the precision required for the CSI reconstruction and the computational availability in the cloud. For example, in healthcare applications, higher precision might be required for legal reasons.

Our results showcase that RSCNet reduces the resources required for sensing by WiFi access points by up to 99.7% while maintaining almost the same performance. This suggests that cloud computing can significantly decrease the computational overhead on devices by shifting it to the cloud. Additionally, we show that using CSI segmentation and smaller CSI windows can also lower the size and complexity of the system. Moreover, the RSCNet compression method can reduce the overhead caused by the transmission of CSI to the cloud without significant damage to the accuracy of the model.

Context-Aware Predictive Coding

We introduced the Context-Aware Predictive Coding (CAPC) framework, designed to extract compact latent representations from unlabelled CSI for downstream WiFi sensing tasks through SSL. CAPC combines two sophisticated approaches: (1) predicting future windows using Contrastive Predictive Coding (CPC) [50] and (2) ensuring consistency between representations of the same sample under varying augmentations through the Barlow Twins methodology [51]. The training of CAPC involves a twin network architecture where each branch's encoder generates latent representations of each CSI window, subjected to different augmentations. An autoregressive

GRU unit then aggregates these representations to create a comprehensive context representation of the sequential windows.

The CPC is executed independently on each branch, utilizing the context embeddings to predict the latent representation of future windows. This predictive task is crucial as it helps the representations to capture valuable temporal information, enhancing their utility for subsequent sensing tasks. Simultaneously, the Barlow Twins loss function guarantees that the context embeddings across the branches remain consistent, thereby effectively mitigating the impacts of augmentations and enhancing the encoder's robustness to variations in CSI. Additionally, we introduced dual-view, a novel augmentation technique specific to CSI WiFi sensing. This method enhances the generalization of the encoder by isolating the free space propagation information—which contains the actual sensing data—from the transmitter and receiver hardware distortions in the CSI, thus improving the quality of the representations.

Our experimental evaluations compare the CAPC's representations with those from other SSL baselines and traditional supervised learning methods. Results demonstrate that CAPC consistently outperforms other methods, particularly in linear and semi-supervised evaluations in unseen test environments, and is especially effective when labelled samples are scarce. Experiments on transfer learning show the CAPC representations outperform other SSL methods for even a different database with a different task and environment showing the practicality of this method in real deployment. Furthermore, extensive testing on common augmentations for time series and CSI WiFi sensing, including our dual view, shows that CAPC not only performs better on average across all augmentation combinations but also uniquely benefits from the dual view augmentation. This capability allows it to effectively remove electronic

distortions and produce superior representations compared to those generated using other augmentations.

Overall, the CAPC framework, both with and without dual view augmentation, delivers high-quality, generalizable representations that significantly enhance performance in WiFi sensing tasks, surpassing existing methods. This success underscores the substantial potential of integrating predictive coding with augmentation consistency in SSL for WiFi sensing. Additionally, the dual view augmentation proves especially beneficial to WiFi sensing, amplifying CAPC's effectiveness by effectively reducing electronic distortions.

5.2 Future Directions

5.2.1 Multi-band WiFi Sensing

The upcoming generation of wireless networks will see a mix of low and high frequencies ranging from sub-6GHz to millimeter and sub-THz frequencies, referred to as multi-band networks [120, 121, 122]. Subsequently, sensing in multi-band WiFi networks will become critical. In this context, WiFi agile multi-band has been recently approved by WiFi Alliance. Furthermore, due to higher transmission frequencies, extremely-large antenna arrays can be supported on APs, thus the machine learning-enabled sensing solutions should be scalable and capable of incorporating a variety of blockage and mobility constraints [123, 124]. In order to further enhance the recognition accuracy in large-scale multi-band networks, edge learning and federated learning solutions with antenna selection and beamforming will be necessary [125, 126].

5.2.2 Autoregressive Model

In both RSCNet and CAPC frameworks, we have utilized RNN units like LSTM and GRU. However, as the number of CSI windows increases, RNNs tend to suffer from information loss due to compression of all past information into a single vector. This limitation restricts the potential for longer duration sensing or the use of smaller window sizes. As an alternative, Transformers, particularly their attention mechanisms, [87] can provide a more robust solution for capturing temporal dynamics and relationships between windows. Attention mechanisms utilize the hidden states of all windows through query, key, and value matrices, offering enhanced temporal awareness. However, it's important to note that while powerful, attention mechanisms require significant computational resources. Employing a cloud-based approach, as in RSCNet, could be essential to manage this computational demand effectively.

5.2.3 Multi-modal Learning

WiFi sensing is characterized by its ubiquity, cost-effectiveness, and ability to preserve privacy, functioning independently of lighting conditions and being partially immune to occlusion. These attributes make it a viable complement to visual sensing techniques. For robust, around-the-clock sensing, integrating multiple data modalities through multi-modal learning is crucial. Recent methods like WiVi [127], WiWeHAR [128] and recent datasets such as WiMANS [75] and MM-Fi [73], which includes video, WiFi CSI, depth frames, and LiDAR, pave the way for this integration. By leveraging joint features from various modalities and choosing reliable modalities for decision-making, multi-modal learning can significantly enhance the performance and versatility of HAR systems. Our CAPC framework, for instance, could be adapted

to train on these multi-modal datasets, aiming for consistent representations across all modalities, akin to the dual view augmentation.

5.2.4 Distributed Large Scale Cloud-Based Sensing

The proliferation of WiFi connectivity and the growth of IoT and smart home technologies mean that most environments are equipped with multiple WiFi devices. This availability of multiple wireless links, which provide varied propagations of WiFi signals, can be harnessed to generate more accurate and high-resolution representations of the environment. While multi-device CSI WiFi sensing approaches like Widar 3.0 [71] have utilized multiple wireless links for processing CSI, they have not fully addressed the challenges related to online CSI synchronization, aggregation, and the computational overhead of incorporating additional CSI streams. We believe that the RSCNet approach to CSI compression and transmission can be adapted to a more distributed architecture. In such a system, each device would compress and transmit its own CSI to the cloud, where a unified environmental representation could be constructed through the fusion of these CSIs using an autoregressive model, potentially enhancing overall sensing performance.

5.2.5 RIS-empowered WiFi Sensing

The complexity and unpredictability of wireless environments significantly impact the accuracy and flexibility of activity recognition due to unwanted multi-path fading and the limited number of independent channels in conventional RF sensing systems. Recently, reconfigurable intelligent surfaces (RIS) [129] have emerged as a promising

technology to actively customize propagation channels, creating favorable environments. By optimizing and programming RIS configurations, it is possible to generate a large number of independent paths, thereby enhancing activity recognition accuracy. The system can overcome the limitations of conventional RF sensing by actively customizing the environment to enhance propagation properties and provide diverse transmission channels.

Bibliography

[1] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.

[2] J. Yang, X. Chen, H. Zou, C. X. Lu, D. Wang, S. Sun, and L. Xie, "Sensefi: A library and benchmark on deep-learning-empowered wifi human sensing," *Patterns*, vol. 4, no. 3, 2023.

[3] S. M. Hernandez and E. Bulut, "Wifi sensing on the edge: Signal processing techniques and challenges for real-world systems," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 46–76, 2023.

[4] S. He, K. Shi, C. Liu, B. Guo, J. Chen, and Z. Shi, "Collaborative sensing in internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1435–1474, 2022.

[5] T. Long, T. Zeng, C. Hu, X. Dong, L. Chen, Q. Liu, Y. Xie, Z. Ding, Y. Li, Y. Wang, and Y. Wang, "High resolution radar real-time signal and information processing," *China Communications*, vol. 16, no. 2, pp. 105–133, 2019.

[6] F. Liu, C. Masouros, A. P. Petropulu, H. Griffiths, and L. Hanzo, "Joint radar and communication design: Applications, state-of-the-art, and the road ahead," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3834–3862, 2020.

[7] J. Hu, J. Yang, J.-B. Ong, D. Wang, and L. Xie, "Resfi: Wifi-enabled device-free respiration detection based on deep learning," in *2022 IEEE 17th International Conference on Control & Automation (ICCA)*, 2022, pp. 510–515.

[8] P. Li, H. Cui, A. Khan, U. Raza, R. Piechocki, A. Doufexi, and T. Farnham, "Deep transfer learning for wifi localization," in *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–5.

[9] R. Zhou, M. Hao, X. Lu, M. Tang, and Y. Fu, "Device-free localization based on csi fingerprints and deep neural networks," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[10] R. Zhou, H. Hou, Z. Gong, Z. Chen, K. Tang, and B. Zhou, "Adaptive device-free localization in dynamic environments through adaptive neural networks," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 548–559, 2020.

[11] L. Yang, T. Kamada, and C. Ohta, "Indoor localization based on csi in dynamic environments through domain adaptation," *IEICE Communications Express*, vol. 10, no. 8, pp. 564–569, 2021.

[12] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fido: Ubiquitous fine-grained wifi-based localization for unlabelled users via domain adaptation," in *Proceedings of The Web Conference 2020*, ser. WWW '20.

New York, NY, USA: Association for Computing Machinery, 2020, p. 23–33. [Online]. Available: https://doi.org/10.1145/3366423.3380091

[13] Y. Zhang, C. Wu, and Y. Chen, "A low-overhead indoor positioning system using csi fingerprint based on transfer learning," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18 156–18 165, 2021.

[14] N. Damodaran, E. Haruni, M. Kokhkharova, and J. Schäfer, "Device free human activity and fall recognition using wifi channel state information (csi)," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, pp. 1–17, 2020.

[15] Z. Shi, Q. Cheng, J. A. Zhang, and R. Yi Da Xu, "Environment-robust wifi-based human activity recognition using enhanced csi and deep learning," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 643–24 654, 2022.

[16] S. Tan and J. Yang, "Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 201–210. [Online]. Available: https://doi.org/10.1145/2942358.2942393

[17] Y. Zhao, R. Gao, S. Liu, L. Xie, J. Wu, H. Tu, and B. Chen, "Device-free secure interaction with hand gestures in wifi-enabled iot environment," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5619–5631, 2021.

[18] M. Raja, V. Ghaderi, and S. Sigg, "Wibot! in-vehicle behaviour and gesture recognition using wireless network edge," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 376–387.

[19] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, 3 2018. [Online]. Available: https://doi.org/10.1145/3191755

[20] Y. Zhao, R. Gao, S. Liu, L. Xie, J. Wu, H. Tu, and B. Chen, "Device-free secure interaction with hand gestures in wifi-enabled iot environment," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5619–5631, 2020.

[21] J. Liu, K. Liu, F. Jin, D. Wang, G. Yan, and K. Xiao, "An efficient csi-based pedestrian monitoring approach via single pair of wifi transceivers," in *International conference on neural computing for advanced applications*. Springer, 2021, pp. 685–700.

[22] X. Wang, Y. Wang, and D. Wang, "A real-time csi-based passive intrusion detection method," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 2020, pp. 1091–1098.

[23] J. Guo and H. Li, "Rswi: A rescue system with wifi sensing and image recognition," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–4.

[24] X. Chen, Z. Tian, M. Zhou, J. Yu, and B. Luo, "Phcount: Passive human number counting using wifi," in *International Conference in Communications, Signal Processing, and Systems*. Springer, 2020, pp. 1214–1223.

[25] B. Korany and Y. Mostofi, "Counting a stationary crowd using off-the-shelf wifi," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 202–214.

[26] D. Konings and F. Alam, "Lifecount: A device-free csi-based human counting solution for emergency building evacuations," in *2020 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2020, pp. 1–5.

[27] S. Di Domenico, M. De Sanctis, E. Cianca, and G. Bianchi, "A trained-once crowd counting method using differential wifi channel state information," in *Proceedings of the 3rd International on Workshop on Physical Analytics*, 2016, pp. 37–42.

[28] S. M. Hernandez and E. Bulut, "Adversarial occupancy monitoring using one-sided through-wall wifi sensing," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[29] Y. Zeng, D. Wu, J. Xiong, E. Yi, R. Gao, and D. Zhang, "Farsense: Pushing the range limit of wifi-based respiration sensing with csi ratio of two antennas," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–26, 2019.

[30] J. Liu, Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng, "Tracking vital signs during sleep leveraging off-the-shelf wifi," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 267–276. [Online]. Available: https://doi.org/10.1145/2746285.2746303

[31] W. Liu, S. Chang, Y. Liu, and H. Zhang, "Wi-psg: Detecting rhythmic movement disorder using cots wifi," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4681–4696, 2020.

[32] B. Korany and Y. Mostofi, "Nocturnal seizure detection using off-the-shelf wifi," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6996–7008, 2021.

[33] X. Wang, C. Yang, and S. Mao, "Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 1230–1239.

[34] I. Shirakami and T. Sato, "Heart rate variability extraction using commodity wifi devices via time domain signal processing," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 2021, pp. 1–4.

[35] D. Wu, D. Zhang, C. Xu, H. Wang, and X. Li, "Device-free wifi human sensing: From pattern-based to model-based approaches," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 91–97, 2017.

[36] Y. Ren, Z. Wang, S. Tan, Y. Chen, and J. Yang, "Tracking free-form activity using wifi signals," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 816–818. [Online]. Available: https://doi.org/10.1145/3447993.3482857

[37] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, 6 2019. [Online]. Available: https://doi.org/10.1145/3310194

[38] D. C. Halperin, "Simplifying the configuration of 802.11 wireless networks with effective snr," *arXiv preprint arXiv:1301.6644*, 2013.

[39] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fidora: Robust wifi-based indoor localization via unsupervised domain adaptation," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9872–9888, 2022.

[40] H. Zou, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Robust wifi-enabled device-free gesture recognition via unsupervised adversarial domain adaptation," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, 2018, pp. 1–8.

[41] J. Yang, X. Chen, H. Zou, D. Wang, and L. Xie, "Autofi: Towards automatic wifi human sensing via geometric self-supervised learning," *IEEE Internet of Things Journal*, 2022.

[42] M. J. Bocus, H.-S. Lau, R. McConville, R. J. Piechocki, and R. Santos-Rodriguez, "Self-supervised wifi-based activity recognition," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 552–557.

[43] 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 *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–10.

[44] D. Liu and T. Abdelzaher, "Semi-supervised contrastive learning for human activity recognition," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2021, pp. 45–53.

[45] K. Xu, J. Wang, L. Zhang, H. Zhu, and D. Zheng, "Dual-stream contrastive learning for channel state information based human activity recognition," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 1, pp. 329–338, 2023.

[46] A. K. Koupai, M. J. Bocus, R. Santos-Rodriguez, R. J. Piechocki, and R. McConville, "Self-supervised multimodal fusion transformer for passive activity recognition," *IET Wireless Sensor Systems*, vol. 12, no. 5-6, pp. 149–160, 2022.

[47] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[48] X. Chen and K. He, "Exploring simple siamese representation learning," 2020.

[49] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

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

[51] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021.

[52] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, 1 2011. [Online]. Available: https://doi.org/10.1145/1925861.1925870

[53] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 53–64. [Online]. Available: https://doi.org/10.1145/2789168.2790124

[54] M. Schulz, D. Wegemer, and M. Hollick. (2017) Nexmon: The c-based firmware patching framework. [Online]. Available: https://nexmon.org

[55] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your csi: A channel state information extraction platform for modern wi-fi chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH '19, 2019, p. 21–28. [Online]. Available: https://doi.org/10.1145/3349623.3355477

[56] A. Sharma, J. Li, D. Mishra, G. Batista, and A. Seneviratne, "Passive wifi csi sensing based machine learning framework for covid-safe occupancy monitoring," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[57] J. Schäfer, B. R. Barrsiwal, M. Kokhkharova, H. Adil, and J. Liebehenschel, "Human activity recognition using csi information with nexmon," *Applied Sciences*, vol. 11, no. 19, p. 8860, 2021.

[58] S. M. Hernandez and E. Bulut, "Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility," in *21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (WoWMoM 2020)*, Cork, Ireland, Jun. 2020.

[59] T. Ropitault, C. R. da Silva, S. Blandino, A. Sahoo, N. Golmie, K. Yoon, C. Aldana, and C. Hu, "Ieee 802.11 bf wlan sensing procedure: Enabling the widespread adoption of wifi sensing," *IEEE Communications Standards Magazine*, vol. 8, no. 1, pp. 58–64, 2024.

[60] R. Du, H. Hua, H. Xie, X. Song, Z. Lyu, M. Hu, Y. Xin, S. McCann, M. Montemurro, T. X. Han *et al.*, "An overview on ieee 802.11 bf: Wlan sensing," *arXiv preprint arXiv:2310.17661*, 2023.

[61] J. K. Brinke and N. Meratnia, "Dataset: Channel state information for different activities, participants and days," in *Proceedings of the 2nd workshop on data acquisition to analysis*, 2019, pp. 61–64.

[62] P. F. Moshiri, R. Shahbazian, M. Nabati, and S. A. Ghorashi, "A csi-based human activity recognition using deep learning," *Sensors*, vol. 21, no. 21, p. 7225, 2021.

[63] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo, "Wiar: A public dataset for wifi-based activity recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, 2019.

[64] J. Yang, Y. Liu, Z. Liu, Y. Wu, T. Li, and Y. Yang, "A framework for human activity recognition based on wifi csi signal enhancement," *International Journal of Antennas and Propagation*, vol. 2021, pp. 1–18, 2021.

[65] A. Baha'A, M. M. Almazari, R. Alazrai, and M. I. Daoud, "A dataset for wi-fi-based human activity recognition in line-of-sight and non-line-of-sight indoor environments," *Data in Brief*, vol. 33, p. 106534, 2020.

[66] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity wi-fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.

[67] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, "Joint activity recognition and indoor localization with wifi fingerprints," *IEEE Access*, vol. 7, pp. 80 058–80 068, 2019.

[68] S. Xu, Z. He, W. Shi, Y. Wang, T. Ohtsuki, and G. Guiy, "Cross-person activity recognition method using snapshot ensemble learning," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[69] S. Ding, Z. Chen, T. Zheng, and J. Luo, "Rf-net: A unified meta-learning framework for rf-enabled one-shot human activity recognition," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 517–530.

[70] P. Hu, C. Tang, K. Yin, and X. Zhang, "Wigr: a practical wi-fi-based gesture recognition system with a lightweight few-shot network," *Applied Sciences*, vol. 11, no. 8, p. 3329, 2021.

[71] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0: Zero-effort cross-domain gesture recognition with wi-fi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8671–8688, 2022.

[72] M. J. Bocus, W. Li, S. Vishwakarma, R. Kou, C. Tang, K. Woodbridge, I. Craddock, R. McConville, R. Santos-Rodriguez, K. Chetty *et al.*, "Operanet, a multimodal activity recognition dataset acquired from radio frequency and vision-based sensors," *Scientific data*, vol. 9, no. 1, p. 474, 2022.

[73] J. Yang, H. Huang, Y. Zhou, X. Chen, Y. Xu, S. Yuan, H. Zou, C. X. Lu, and L. Xie, "Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[74] F. Meneghello, C. Chen, C. Cordeiro, and F. Restuccia, "Toward integrated sensing and communications in ieee 802.11 bf wi-fi networks," *IEEE Communications Magazine*, vol. 61, no. 7, pp. 128–133, 2023.

[75] S. Huang, K. Li, D. You, Y. Chen, A. Lin, S. Liu, X. Li, and J. A. McCann, "Wimans: A benchmark dataset for wifi-based multi-user activity sensing," *arXiv preprint arXiv:2402.09430*, 2024.

[76] C. Xiao, D. Han, Y. Ma, and Z. Qin, "Csigan: Robust channel state information-based activity recognition with gans," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 191–10 204, 2019.

[77] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[79] R. Ayachi, M. Afif, Y. Said, and M. Atri, "Strided convolution instead of max pooling for memory efficiency of convolutional neural networks," in *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Vol. 1.* Springer, 2020, pp. 234–243.

[80] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[81] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[82] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986," *Biometrika*, vol. 71, pp. 599–607, 1986.

[83] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[84] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[85] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[86] B. Barahimi, H. Singh, H. Tabassum, O. Waqar, and M. Omer, "Rsc-net: Dynamic csi compression for cloud-based wifi sensing," *arXiv preprint arXiv:2402.04888*, 2024.

[87] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[88] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[89] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 286–293.

[90] A. Zhuravchak, O. Kapshii, and E. Pournaras, "Human activity recognition based on WiFi CSI data - a deep neural network approach," *Procedia Computer Science*, vol. 198, pp. 59–66, 2022.

[91] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi csi based passive human activity recognition using attention based blstm," *IEEE Trans. on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2019.

[92] C. Xiao, Y. Lei, Y. Ma, F. Zhou, and Z. Qin, "Deepseg: Deep-learning-based activity segmentation framework for activity recognition using wifi," *IEEE Internet of Things Jrnl.*, vol. 8, no. 7, pp. 5669–5681, 2021.

[93] Y. Gu, X. Zhang, Y. Wang, M. Wang, H. Yan, Y. Ji, Z. Liu, J. Li, and M. Dong, "WiGRUNT: WiFi-enabled gesture recognition using dual-attention network," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 4, pp. 736–746, 2022.

[94] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "Deepsense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *2018 IEEE Intl. Conf. on Commun. (ICC)*, 2018, pp. 1–6.

[95] Z. Shi, Q. Cheng, J. A. Zhang, and R. Y. Da Xu, "Environment-robust wifi-based human activity recognition using enhanced csi and deep learning," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 643–24 654, 2022.

[96] S. Ji, Y. Xie, and M. Li, "Sifall: Practical online fall detection with rf sensing," in *Proceedings of the 20th ACM Conf. on Embedded Networked Sensor Systems*, 2022, pp. 563–577.

[97] L.-H. Shen, K.-J. Chen, A.-H. Hsiao, and K.-T. Feng, "Bts: Bifold teacher-student in semi-supervised learning for indoor two-room presence detection under time-varying csi," 2023.

[98] R. Balestriero and Y. LeCun, "Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 671–26 685, 2022.

[99] R. Song, D. Zhang, Z. Wu, C. Yu, C. Xie, S. Yang, Y. Hu, and Y. Chen, "Rf-url: Unsupervised representation learning for rf sensing," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, ser. MobiCom '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 282–295. [Online]. Available: https://doi.org/10.1145/3495243.3560529

[100] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," 2022.

[101] S. Tang, J. Xia, L. Fan, X. Lei, W. Xu, and A. Nallanathan, "Dilated convolution based csi feedback compression for massive mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11 216–11 221, 2022.

[102] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges," in *2015 IEEE international conference on services computing*. IEEE, 2015, pp. 285–292.

[103] J. Yang, X. Chen, H. Zou, D. Wang, Q. Xu, and L. Xie, "Efficientfi: Toward large-scale lightweight wifi sensing via csi compression," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 086–13 095, 2022.

[104] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[105] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[106] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "Bm3d-amp: A new image recovery algorithm based on bm3d denoising," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3116–3120.

[107] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[108] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[109] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed deep convolutional compression for massive mimo csi feedback," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2621–2633, 2020.

[110] S. Tang, J. Xia, L. Fan, X. Lei, W. Xu, and A. Nallanathan, "Dilated convolution based CSI feedback compression for massive MIMO systems," *IEEE Trans. Veh. Tech.*, vol. 71, no. 10, pp. 11 216–11 221, 2022.

[111] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[112] P. Vaidyanathan, S.-M. Phoong, and Y.-P. Lin, *Signal processing and optimization for transceiver systems*. Cambridge University Press, 2010.

[113] M. N. Mohsenvand, M. R. Izadi, and P. Maes, "Contrastive representation learning for electroencephalogram classification," in *Proceedings of the Machine Learning for Health NeurIPS Workshop*, ser. Proceedings of Machine Learning Research, E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy, and S. L. Hyland, Eds., vol. 136. PMLR, 12 2020, pp. 238–253. [Online]. Available: https://proceedings.mlr.press/v136/mohsenvand20a.html

[114] K. Davaslioglu, S. Boztaş, M. C. Ertem, Y. E. Sagduyu, and E. Ayanoglu, "Self-supervised rf signal representation learning for nextg signal classification with deep learning," *IEEE Wireless Communications Letters*, vol. 12, no. 1, pp. 65–69, 2023.

[115] Y. You, I. Gitman, and B. Ginsburg, "Scaling SGD batch size to 32k for imagenet training," *CoRR*, vol. abs/1708.03888, 2017. [Online]. Available: http://arxiv.org/abs/1708.03888

[116] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with restarts," *CoRR*, vol. abs/1608.03983, 2016. [Online]. Available: http://arxiv.org/abs/1608.03983

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

[118] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," *arXiv preprint arXiv:2110.09348*, 2021.

[119] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Jrnl. of Machine Learning Research*, vol. 9, no. 11, 2008.

[120] M. A. Saeidi, H. Tabassum, and M.-S. Alouini, "Multi-band wireless networks: Architectures, challenges, and comparative analysis," *IEEE Communications Magazine*, 2023.

[121] S. Aboagye, M. Amin Saeidi, H. Tabassum, Y. Tayyar, E. Hossain, H.-C. Yang, and M.-S. Alouini, "Multi-band wireless communication networks: Fundamentals, challenges, and resource allocation," *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 4333–4383, 2024.

[122] M. T. Hossan and H. Tabassum, "Mobility-aware performance in hybrid rf and terahertz wireless networks," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1376–1390, 2022.

[123] M. Alizadeh and H. Tabassum, "Power control with qos guarantees: A differentiable projection-based unsupervised learning framework," *IEEE Transactions on Communications*, vol. 71, no. 8, pp. 4605–4619, 2023.

[124] M. Alizadeh, X. Mootoo, O. Waqar, and H. Tabassum, "Qos-aware deep unsupervised learning for star-ris assisted networks: A novel differentiable projection framework," *IEEE Wireless Communications Letters*, 2024.

[125] S. Zarandi and H. Tabassum, "Federated double deep q-learning for joint delay and energy minimization in iot networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[126] S. Asaad, H. Tabassum, C. Ouyang, and P. Wang, "Joint antenna selection and beamforming for massive mimo-enabled over-the-air federated learning," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2024.

[127] H. Zou, J. Yang, H. Prasanna Das, H. Liu, Y. Zhou, and C. J. Spanos, "Wifi and vision multimodal learning for accurate and robust device-free human activity recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 6 2019.

[128] M. Muaaz, A. Chelli, A. A. Abdelgawwad, A. C. Mallofré, and M. Pätzold, "Wiwehar: Multimodal human activity recognition using wi-fi and wearable sensing modalities," *IEEE Access*, vol. 8, pp. 164 453–164 470, 2020.

[129] T. Shafique, H. Tabassum, and E. Hossain, "Stochastic geometry analysis of irs-assisted downlink cellular networks," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1442–1456, 2022.