Toward Robust and Deployable Representation Learning for AI-Native Wireless Sensing

Ahmed Youssef Radwan

A Thesis Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

Graduate Program in Electrical Engineering and Computer Science

YORK UNIVERSITY

TORONTO, ONTARIO

April 2026

© Ahmed Youssef Radwan, 2026

Abstract

Device-free Wi-Fi sensing offers a privacy-preserving approach to human activity recognition (HAR), yet practical deployment faces two key challenges: domain shifts across sensing environments and the gap between CSI compression and prediction in dynamic wireless systems. Existing Unsupervised Domain Adaptation (UDA) methods rely on labeled source data and fail in multi-user scenarios due to signal entanglement, permutation invariance, and severe class imbalance. AI-driven CSI feedback methods treat compression and prediction as independent problems, leaving channel aging insufficiently addressed. This thesis proposes two complementary frameworks. First, MU-SHOT-Fi is a Source-Free UDA framework for multi-user Wi-Fi sensing employing a permutation-invariant set-prediction architecture trained via Hungarian matching. It introduces an occupancy-weighted information maximization objective to handle class imbalance and integrates rotation-based spatial self-supervision to align cross-domain representations using the frequency-time structure of Channel State Information (CSI). A single-user extension, SU-SHOT-Fi, incorporates temporal self-supervision via Contrastive Predictive Coding (CPC). Second, a unified compression-prediction framework integrates CPC into a 3GPP-compliant

CSI pipeline, jointly optimizing reconstruction fidelity and temporal predictive coherence to address channel aging without increasing feedback overhead. Two variants, CPC-before-Compression and CPC-after-Compression, offer different trade-offs between temporal modeling and device complexity. Evaluations on WiMANS and Widar 3.0 demonstrate consistent improvements over state-of-the-art baselines under cross-room and cross-frequency shifts. Experiments on Nokia, Oppo, and CATT datasets confirm favorable complexity-performance trade-offs, advancing AI-native wireless sensing toward robust, practical deployment.

Acknowledgments

I would like to thank Prof. Hina Tabassum for her continuous support throughout my studies, for guiding me, for sharing her expertise, and for giving me the opportunity to join the NGWN Research Lab. I would also like to thank my thesis committee members, Prof. Laleh Seyyed-Kalantari and Prof. Mehdi Nourinejad. I want to extend my gratitude to those who supported me throughout my career and believed in me: Prof. Slim Alouini, Prof. Tareq Al-Naffouri, Dr. Naeemullah Khan, and Dr. Emad Al-Ghamdi. And to Dr. Shaina Raza, who guided me in expanding my knowledge in AI and never stopped supporting me along the way.

I also cannot forget the friends who were there for me through the long nights when I was overwhelmed and just needed someone to talk to. If I started naming everyone, we would be here all day, so let's just say: Barqeen's group. And especially Ahmad Sait, who not only stood by me throughout our Master's journey and stayed connected with me, but who also constantly pushed me to be a better version of myself. I would also like to thank Faisal Alzahrani, whom I had the pleasure of working with during my first research internship. His support and collaboration during that time meant a great deal to me.

Last but not least, my family, who supported me through all these years and were the reason I came here in the first place. Whatever I have achieved, and whatever I

will achieve, is by the grace of Allah and then because of them.

This is also dedicated to the memory of my father, Youssef Ragab Radwan, may Allah grant him Jannah. He always told me: *whatever you want to do in life, whatever you believe is best for you, do it. Because when you choose your own path, you will succeed, and you will become one of the greatest.* And this is for every Palestinian, who carries their identity as both a wound and a pride. For a people who have endured displacement, loss, and relentless hardship, yet have never stopped dreaming, building, and pushing forward. May Allah grant them peace, dignity, and the home they deserve.

And finally, this work is for everyone who ever felt they did not have the opportunity to make something of themselves, because of their nationality, their background, or where they come from. For those who felt they had nowhere to belong. Your chance will come. And when it does, give it everything you have. You will see the return.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Wireless Sensing and Its Applications

The rapid spread of Wi-Fi infrastructure has enabled applications that go well beyond connectivity. Wi-Fi (RF) signals propagate throughout built environments, reflecting, diffracting, and scattering from people and objects, creating rich multipath patterns that carry environmental information such as human mobility, location, and item state [2, 3]. These effects are commonly decoded using two core metrics: *Channel State Information (CSI)* and *Received Signal Strength Indicator (RSSI)*. CSI captures the complex-valued channel frequency response (CFR), which varies with path length, multipath, and Doppler shifts, whereas RSSI measures received signal power influenced by distance and obstructions. Motion sensitivity can be enhanced with *CSI similarity* which often computed via cross-correlation between CSI matrices to distinguish static from dynamic scenes. In addition, *channel coherence time* and *coherence bandwidth*, the temporal and spectral intervals over which the channel impulse response (CIR) remains stable—are widely used to infer device mobility.

These signal features underpin a broad set of applications (**Table 1.1**), including

crowd counting, occupancy detection, pose estimation, vital-sign monitoring, human activity recognition, and gesture recognition [4].

1.1.1 Localization

Wi-Fi sensing can localize targets by inferring position from ambient signals. Early approaches required the target to carry a transmitter or receiver, but recent work demonstrates *device-free* localization using CSI-based fingerprints that capture the spatial layout and scattering characteristics of a site [5–9]. Modern systems fuse amplitude, phase, and Doppler features (e.g., via short-time Fourier transforms of subcarrier streams) and leverage multi-AP diversity or antenna arrays to estimate time-of-flight and angle-of-arrival. To remain accurate across deployment sites and over time, methods frequently employ domain adaptation and transfer learning to mitigate environmental drift, and self-calibration schemes to reduce labor-intensive site surveys.

1.1.2 Human Activity and Gesture Recognition

Human Activity Recognition (HAR). HAR maps temporal CSI/RSSI dynamics to semantic labels such as sitting, standing, walking, or running. Typical pipelines denoise and unwrap phase, extract time–frequency or Doppler signatures, and train sequence models (e.g., CNNs, RNNs, or Transformers) to capture motion primitives [2, 10]. Beyond single-person activities, multi-person HAR aggregates spatial streams across APs to resolve overlapping motions. Datasets like *UT-HAR* promote reproducibility and cross-environment evaluation, enabling standardized comparisons and robustness studies (e.g., leave-one-room-out, leave-one-subject-out) [11]. Practical uses include

occupancy awareness for smart buildings and privacy-preserving fall detection for elder care without cameras.

Gesture Recognition. Fine-grained hand and finger movements induce micro-Doppler and phase perturbations in CSI that can be learned for gesture vocabularies and continuous trajectories. Systems often use subcarrier selection and reference linking to stabilize phase, then classify gestures with temporal convolution or attention modules. This enables touchless interaction in homes and vehicles and supports applications such as sign-language components and mid-air controls [12–15]. For example *Widar 3.0* highlights the generalization of the cross-domain (e.g.main (e.g. new users, positions, and environments), offering protocols and benchmemphasize robustness to changes in deploymentes in deployment [16].

1.1.3 Crowd Counting and Occupancy Detection

Subtle, involuntary human motions (e.g., postural sway, fidgeting, respiration) modulate multipath in ways that scale with the number of people. By modeling variance, entropy, and Doppler spread in CSI streams—or by using supervised regressors over subcarrier features—Wi-Fi sensing can estimate headcounts and occupancy even when individuals are stationary [2, 17]. These capabilities support safety monitoring, queue analytics, and demand-aware operations in venues such as retail stores, airports, hospitals, and theme parks, and can inform HVAC control and space planning in smart buildings.

1.1.4 Health Monitoring

Contact-free vital-sign monitoring exploits micro-motions of the chest wall and cardiac activity that imprint periodic structure on CSI. Systems track respiration rate, detect irregular breathing (e.g., apnea, tachypnea) during sleep, estimate heart rate, and derive heart-rate variability by stabilizing phase across subcarriers and applying spectral peak tracking or adaptive filtering [18–22]. Multi-AP diversity and beamforming can improve SNR and mitigate body orientation effects, while personalization or transfer learning helps maintain accuracy across subjects and rooms.

1.1.5 Practical Considerations.

Because CSI is highly sensitive to hardware differences, room layout changes, and user variability, models can degrade when moved across domains. In practice, lightweight domain adaptation (e.g., calibration-free phase handling and test-time/statistics-based alignment) helps maintain accuracy without new labels. For real-time use, compact CSI feedback and short-horizon prediction mitigate channel aging [23–25] and reduce latency, enabling stable closed-loop operation on commodity Wi-Fi. We discuss these challenges—domain shift and fast, reliable CSI feedback—in more detail in the *Challenges* subsection.

1.2 Evolution of Wi-Fi Standards

The journey of Wi-Fi began with the IEEE 802.11 standard in 1997, operating at 2.4 GHz with data rates up to 2 Mbps, laying the foundation for wireless communication. The subsequent introduction of IEEE 802.11b (Wi-Fi 1) and IEEE 802.11a (Wi-Fi 2) in 1999 brought enhanced speeds—up to 11 Mbps at 2.4 GHz with 802.11b and up

Table 1.1: Wi-Fi Sensing Applications

Application	**References**	**Description**
Localization	[5], [6], [7], [8], [9]	Tracking a target's location within a space using ambient Wi-Fi signals. Traditional methods require the target to have transmitting or receiving hardware. Recent studies focus on device-free tracking using CSI data to create environmental signal fingerprints. Techniques like Domain Adaptation and Transfer Learning are used to address accuracy issues due to environmental changes.
Human Activity Recognition (HAR)	[2], [10]	Recognizes activities such as sitting, standing, walking, and running. Useful for monitoring room occupancy in smart homes. Fall detection safeguards elderly or ill individuals without invading privacy, unlike camera-based systems.
Gesture Recognition	[12], [13], [14], [15]	Identifies intricate human movements focusing on hand and finger gestures. Enables innovative gesture-based interactions in smart homes and in-vehicle controls. Sign Language detection is another use case.
Crowd Counting and Occupancy Detection	[2], [17]	Estimates the number of people in a given area. Detects stationary crowd sizes by subtle, random fidgeting movements. Valuable for safety-critical and non-critical situations. Used to monitor human queues or tally individuals in waiting rooms. Helps improve services in various locations like retail stores, airports, hospitals, and theme parks. Optimizes transportation schedules and boarding/payment procedures based on passenger numbers.
Health Monitoring	[18], [19], [20], [21], [22]	Continuous health monitoring in private residences. Common applications include respiration tracking and detecting irregular breathing patterns, such as apnea or tachypnea, especially in sleep. Tracks heart rates to reveal heart rhythm variability.

to 54 Mbps at 5 GHz with 802.11a improving accessibility and reliability. By 2003, IEEE 802.11g (Wi-Fi 3) maintained the 54 Mbps data rate at 2.4 GHz, with improved range and compatibility.

A major leap occurred in 2009 with IEEE 802.11n (Wi-Fi 4), which supported both 2.4 GHz and 5 GHz frequencies and introduced MIMO technology, enabling speeds up to 600 Mbps. MIMO's ability to enhance capacity and reliability was crucial for the development of environmental mapping in Wi-Fi sensing. In 2013, IEEE 802.11ac (Wi-Fi 5) advanced Wi-Fi sensing by achieving data rates up to 3.5 Gbps at 5 GHz, enabling higher resolution for detailed motion and presence detection. The 2019 release of IEEE 802.11ax (Wi-Fi 6) further optimized performance in high-density environments, reaching speeds up to 9.6 Gbps across 2.4 GHz and 5 GHz

bands. The extension Wi-Fi 6E, introduced in 2020, expanded these capabilities to the 6 GHz band, offering increased capacity and reduced congestion.

As Wi-Fi technology evolved, standards-based technologies emerged to bring an intelligent approach and interoperability across networks. The IEEE 802.11k standard enables access points and clients to exchange information about the Wi-Fi environment, allowing devices to optimize their connections. IEEE 802.11v uses this network information to influence client roaming decisions, facilitating improved network management, while IEEE 802.11u enables devices to gather information from other networks before connecting. IEEE 802.11r further enhances network efficiency by enabling fast transitions between access points within a Wi-Fi network, supporting applications that require seamless mobility. Together, these standards support Wi-Fi Agile Multiband, a technology that improves network management and interoperability across various vendors. Additionally, Wi-Fi Alliance-defined technologies supplement these standards by exchanging supplementary information, and identifying preferred channels, bands, or access points to further enhance intelligent Wi-Fi network management [26]. The introduction of IEEE 802.11bf in 2020 marked a significant milestone by supporting sensing across multiple frequency bands, including 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz. This standard builds on the protocols of IEEE 802.11ad and IEEE 802.11ay for the 60 GHz band, minimizing communication overhead and making Wi-Fi sensing more accessible and efficient [27]. Further advancements are expected with the forthcoming IEEE 802.11be (Wi-Fi 7), which aims to deliver data rates up to 30 Gbps across multiple frequency bands. With enhancements like multi-link operation and wider channel bandwidths. Wi-Fi 7 is poised to support even more accurate environmental data collection for advanced sensing

applications. Beyond Wi-Fi 7, the next major evolution—Wi-Fi 8, based on IEEE 802.11bn and anticipated by 2028—will operate across the 2.4, 5, and 6 GHz bands, introducing multiple access point coordination and transmission over millimetre wave (mmWave) frequencies. To reach speeds up to 100 Gbps, far surpassing copper Ethernet's 40 Gbps limit, Wi-Fi 8 promises low-latency connectivity but may experience signal attenuation in rainy conditions, similar to satellite communications. Deploying Wi-Fi 8 will require retrofitting ceiling-mounted access points to support these advancements [28, 29].

The IEEE 802.11bf standard represents a pivotal advancement, formally integrating Wi-Fi sensing capabilities such as gesture recognition, presence detection, and environmental monitoring into mainstream wireless protocols, while ensuring coexistence with legacy Wi-Fi deployments [30, 31]. This amendment bridges the gap between Wi-Fi sensing research and practical deployment, standardizing features for bistatic and multistatic sensing across the 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz bands. Building upon this foundation, Wi-Fi 7 (802.11be) introduces wider bandwidths, multi-link operation (MLO), and lower latencies, enabling more accurate and real-time environmental sensing [32]. Looking ahead, Wi-Fi 8 (802.11bn) is poised to revolutionize high-demand sensing applications by prioritizing ultra-high reliability (UHR) and integrating millimeter-wave (mmWave) communications [29, 32]. These enhancements will enable future sensing use cases such as augmented reality (AR) and holographic communications, where immediate feedback, extremely high throughput, and minimal latency are critical [33]. Wi-Fi 8's capabilities will not only enhance conventional sensing but also open new avenues for immersive applications in the metaverse and real-time digital twin systems, overcoming many hardware limitations

that constrained Wi-Fi 7.

1.3 The Generalization Challenge in Data-Driven Sensing

Deep learning has revolutionized wireless sensing by replacing manual feature engineering with automated feature extraction from complex CSI [34]. However, these data-driven models fundamentally rely on the assumption that training (source) and testing (target) data share identical statistical distributions. In real-world wireless deployments, this assumption rarely holds. Environmental changes, such as furniture repositioning, user diversity, or varying carrier frequencies, introduce severe domain shifts that degrade model performance. While Unsupervised Domain Adaptation (UDA) has emerged to bridge these gaps by aligning source and target distributions, traditional UDA approaches necessitate simultaneous access to labeled source data and unlabeled target data. This dependency poses severe privacy risks and storage bottlenecks, particularly in edge-computing scenarios where data transmission is constrained. Consequently, there is a compelling need for Source-Free Unsupervised Domain Adaptation (SFUDA) frameworks that can adapt pre-trained models to new environments using only unlabeled target data, thereby ensuring privacy, reducing communication overhead, and enabling robust deployment in dynamic sensing environments.

1.4 Deployment Considerations for AI-Native Wireless Sensing

While SFUDA addresses adaptation without access to source data, practical wireless sensing deployment introduces additional constraints beyond recognition performance alone. In real-world operation, CSI must be acquired, represented, transmitted, and

processed under limited communication budgets and resource-constrained devices. This makes deployment not only an adaptation problem, but also a system-level design problem.

This perspective is consistent with the evolution toward AI-native wireless systems, where recent Third Generation Partnership Project (3GPP)[1] studies consider CSI feedback enhancement through standardized training and deployment pipelines. Within this framework, CSI compression and CSI prediction are both critical for practical deployment [35]. A large number of deep learning-based prediction techniques have been discussed here [36–38]. Compression reduces feedback overhead and storage burden, while prediction mitigates the effect of channel aging by compensating for delays between CSI acquisition and usage. Existing works, however, often treat these objectives separately, which limits their effectiveness in dynamic environments.

Motivated by this, the thesis considers deployment from two complementary perspectives. On one hand, SFUDA enables robust adaptation of sensing models under domain shift without relying on source data. On the other hand, contrastive predictive coding (CPC) with compression improves deployment efficiency by integrating temporal prediction into a 3GPP-compliant CSI feedback pipeline, thereby supporting robust generalization across deployment scenarios while reducing model complexity for resource-constrained user equipment (UEs).

[1] 3GPP is a global collaboration of telecom standards organizations that develops unified specifications for mobile networks (from 3G to 5G and beyond), ensuring worldwide interoperability and innovation in wireless communications.

1.5 Limitations and Research Gaps

While deep learning has emerged as a prominent approach for Wi-Fi sensing, its application to robust, privacy-preserving multi-user scenarios faces fundamental challenges. Understanding these limitations is essential for developing frameworks that can meet the dynamic requirements of real-world deployments. Existing learning-based sensing approaches exhibit several critical limitations:

- **Domain Sensitivity and Negative Transfer:** CSI is inherently sensitive to environmental changes; modest changes in room layout, user placement, or carrier frequency substantially alter CSI statistics, causing abrupt domain shifts [39–41]. Enforcing domain invariance in multi-user settings often leads to negative transfer, as environment-dependent variations are tightly coupled with the cues required for user separation [42, 43].

- **Privacy Concerns in Adaptation:** Most existing Unsupervised Domain Adaptation (UDA) methods rely on access to labeled source data during the adaptation process [44–46]. This reliance raises significant privacy concerns and logistical overhead when data is confidential or transmission is bandwidth-constrained, limiting practical deployment [47].

- **Inability to Handle Multi-User Dynamics:** Current Source-Free Unsupervised Domain Adaptation (SFUDA) frameworks focus exclusively on single-user cases [48,49]. They fail in multi-user sensing scenarios due to three fundamental gaps:

  - *Signal Entanglement:* Multi-user CSI reflects a superposition of activities where identical features may correspond to different user configurations,

violating the stable feature mapping assumptions of standard adaptation methods [50].

  - *Class Imbalance:* In variable occupancy scenarios, "no person" slots dominate the data distribution. Standard information maximization objectives cause model collapse toward this dominant class [51].

  - *Permutation Invariance:* Clustering-based pseudo-labeling assumes fixed class assignments [52], but multi-user outputs are permutation-invariant (e.g., {Walk, Sit} is equivalent to {Sit, Walk}), rendering standard pseudo-labeling techniques ineffective.

- **Separation of CSI Compression and Prediction:** Existing AI-driven CSI feedback works predominantly treat CSI compression and CSI prediction as separate problems [53–60]. This separation is also reflected in current 3GPP studies, where CSI feedback enhancement is discussed through distinct sub-use cases, such as compression with two-side models and prediction with UE-side models [35,61]. As a result, existing frameworks do not jointly address feedback efficiency and channel aging, leaving an important gap for deployment-oriented wireless systems operating under dynamic channel conditions. The only work which considered both CSI compression and prediction is [34].

These limitations motivate the development of robust, source-free adaptation frameworks capable of handling the entanglement, imbalance, and permutation invariance inherent in multi-user Wi-Fi sensing, while also highlighting the need for unified CSI feedback frameworks that jointly address compression and prediction under practical deployment constraints.

1.6 Scope of the Thesis

Given the challenges of domain shift in CSI data, the privacy limitations of existing UDA methods, the gaps in multi-user adaptation, and the growing need for deployment-efficient CSI feedback under dynamic wireless conditions, this thesis develops frameworks for both robust source-free adaptive sensing and practical AI-native wireless deployment. The thesis addresses two complementary research directions: robust adaptation of sensing models without source-data access, and unified CSI feedback enhancement through joint compression and prediction.

For the sensing component, we propose *MU-SHOT-Fi*, a framework that adopts a set-based prediction approach to formulate multi-user recognition as a permutation-invariant problem [62, 63]. Its key contributions can be summarized as follows:

- **Multi-User SFUDA Problem Analysis:** We identify and formalize the fundamental challenges that prevent existing SFUDA methods from extending to multi-user sensing: class imbalance from unoccupied slots, permutation invariance invalidating pseudo-labeling, and signal entanglement violating stable feature-to-label mappings. We further demonstrate that CPC, while beneficial for single-user adaptation, degrades multi-user performance due to conflicts between temporal consistency and permutation-invariant slot predictions, which is a finding with broader implications for temporal SSL is structured prediction settings.

- **Occupancy-Weighted Information Maximization:** To address the critical issue of class imbalance, we introduce a novel adaptation objective that weights diversity regularization by slot occupancy probability. This approach prevents

the model from collapsing toward the dominant "no person" class while ensuring diverse predictions for occupied slots.

- **Spatial and Temporal Self-Supervision:** We integrate rotation-based spatial self-supervision to align source and target representations by exploiting the frequency-time structure of CSI without requiring labels [64]. Furthermore, for single-user cases, we incorporate Contrastive Predictive Coding (CPC) to leverage temporal dependencies for enhanced domain invariance.

- **Unified Framework and Comprehensive Evaluation:** We demonstrate that the proposed framework generalizes to single-user scenarios (SU-SHOT-Fi) as a special case. We validate the framework on the WiMANS and Widar 3.0 datasets [51, 65], demonstrating superior performance over state-of-the-art baselines across cross-room and cross-frequency domain shifts.

For the deployment-oriented CSI feedback component, this thesis further investigates AI-driven CSI feedback under dynamic wireless conditions. Existing works largely treat CSI compression [53–58] and CSI prediction [59, 60] as separate problems, while recent joint approaches remain limited in how temporal prediction is incorporated into compact representations [66]. Motivated by this gap, this thesis develops a unified framework within a 3GPP-oriented pipeline. The main contributions in this direction are as follows:

- **Novel integration of CPC into a standardized compression pipeline.** We develop a joint compression–prediction framework that integrates CPC into a standardized CSI feedback pipeline. Unlike conventional CSI compression schemes that focus solely on reconstruction, our design incorporates CPC within

a 3GPP-compliant architecture featuring a quantized linear bottleneck. The proposed approach jointly optimizes reconstruction fidelity and temporal prediction, addressing a challenge not considered in prior works.

- **Two architecturally distinct variants addressing a real deployment tradeoff.** We propose two architecturally distinct variants:
  CPC-before-Compression, which employs a GRU-based autoregressive module on raw encoded features before dimensionality reduction, and *CPC-after-Compression*, which defers temporal modeling to the BS decoder, reducing UE-side complexity. Both variants learn robust representations that remain stable across prediction horizons. This tradeoff between feature preservation and computational efficiency has not been explored in prior CSI feedback literature [57, 58].

- **Joint SGCS and InfoNCE training objective.** The joint 1-SGCS and InfoNCE loss function goes beyond standard reconstruction objectives by embedding a self-supervised contrastive predictive objective directly into the compression training loop, encouraging temporally coherent latent representations without requiring labeled data.

- **Evaluation of computational power and accuracy trade-offs.** We evaluate our methods on 3GPP-compliant datasets from Nokia, Oppo, and CATT, demonstrating 32× lower decoder GFLOPs for *CPC-before-Compression* and an identical encoder footprint for *CPC-after-Compression* compared to the 3GPP baseline, while *CPC-before-Compression* consistently achieves reconstruction accuracy exceeding 90%. Also, model compression techniques like pruning and

low-rank factorization [67, 68] are implemented to reduce inference costs.

1.7 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews Wi-Fi sensing fundamentals, preprocessing techniques, and the state-of-the-art in deep learning and domain adaptation for CSI-based sensing. Chapter 3 presents the proposed MU-SHOT-Fi framework, detailing the set-based formulation, occupancy-weighted adaptation objectives, and self-supervised learning components. It also presents the SU-SHOT-Fi special case and validates the frameworks through extensive numerical results under various domain shift scenarios. Chapter 4 presents the deployment-oriented CSI feedback framework, detailing how CPC can be integrated with CSI compression to jointly address feedback efficiency and channel aging within a 3GPP-aligned pipeline. Chapter 5 concludes the thesis and discusses future research directions.

1.8 Research Outcomes

- A. Y. Radwan, M. Yildirim, N. Hasanzadeh, H. Tabassum, and S. Valaee, "A Tutorial-cum-Survey on Self-Supervised Learning for Wi-Fi Sensing: Trends, Challenges, and Outlook," *IEEE Communications Surveys & Tutorials*, accepted, 2025. Code: `https://github.com/AhmedRadwan02/ssl-wifi-sensing`.

- A. Y. Radwan and H. Tabassum, "MU-SHOT-Fi: Self-Supervised Multi-User Wi-Fi Sensing with Source-Free Unsupervised Domain Adaptation," *IEEE Internet of Things Journal*, accepted, 2026.
  Code: `https://github.com/AhmedRadwan02/mu-shot-fi`.

- A. Y. Radwan, H. Tabassum, F. S. Muhammad, and M. Baker, "Contrastive Predictive Coding with Compression for Enhanced Channel State Feedback in Wireless Networks," *IEEE Transactions on Neural Networks and Learning Systems*, under review. Code: `https://github.com/AhmedRadwan02/cpc-3gpp`.

Table 1.2: Evolution of Wi-Fi Standards and Their Specifications

IEEE Standard	**Year**	**Frequency Bands**	**Channel Width**	**Data Rate**	**Notes**
802.11 (Wi-Fi 0)	1997	2.4 GHz	20 MHz	Up to 2 Mbps	• Laid the foundation for wireless communication.
802.11b (Wi-Fi 1)	1999	2.4 GHz	20 MHz	Up to 11 Mbps	• Increased data rates, making Wi-Fi more widely adopted.
802.11a (Wi-Fi 2)	1999	5 GHz	20 MHz	Up to 54 Mbps	• Operated in 5 GHz, providing higher data rates and less interference.
802.11g (Wi-Fi 3)	2003	2.4 GHz	20 MHz	Up to 54 Mbps	• Combined speed of 802.11a with 802.11b's range and compatibility.
802.11n (Wi-Fi 4)	2009	2.4 GHz and 5 GHz	40 MHz	Up to 600 Mbps	• Introduced MIMO for improved speed and reliability. • Enabled better performance on both 2.4 GHz and 5 GHz bands.
802.11ad (WiGig)	2012	60 GHz	2.16 GHz	Up to 8 Gbps	• Utilized 60 GHz for short-range, high-speed, low-latency applications.
802.11ac (Wi-Fi 5)	2013	5 GHz	80/160 MHz	Up to 3.5 Gbps	• Enhanced with wider channels, more channels, and higher modulation density.
802.11ax (Wi-Fi 6)	2019	2.4 GHz and 5 GHz	Up to 160 MHz	Up to 9.6 Gbps	• Optimized for dense environments with OFDMA and better spectral efficiency.
802.11ax (Wi-Fi 6E)	2020	6 GHz	Up to 160 MHz	Up to 9.6 Gbps	• Extended Wi-Fi 6 into the 6 GHz band for increased capacity and reduced congestion.
802.11bf	2020	2.4 GHz, 5 GHz, 6 GHz, and 60 GHz	N/A	N/A	• Focused on Wi-Fi sensing, enabling applications in motion and presence detection.
802.11ay	2021	60 GHz	2.16/4.32/6.48/ 64 GHz	Up to 176 Gbps	• Enhanced 802.11ad with higher data rates and channel bonding.
802.11be (Wi-Fi 7)	2024	2.4 GHz, 5 GHz, and 6 GHz	Up to 320 MHz	Up to 30 Gbps	• Aims for higher data rates with multi-link operation and wider channels.
802.11bn (Wi-Fi 8)	2028	2.4 GHz, 5 GHz, and 6 GHz	TBD	Up to 100 Gbps	• Utilizes multiple access point coordination and mmWave frequencies. • Optimized for low-latency applications and very high data rates. • Rain attenuation similar to satellite communications may impact performance. • Retrofit of ceiling-mounted access points required for deployment.

Abbreviations: MIMO: Multiple-Input Multiple-Output, OFDMA: Orthogonal Frequency-Division Multiple Access, NDPs: Null Data Packets

Chapter 2

Literature Review and Preliminaries

2.1 Wi-Fi Sensing Fundamentals

Wi-Fi sensing has emerged as a complementary alternative to camera-based indoor sensing and tracking. It operates without line-of-sight (LoS) or ambient lighting and can be more privacy-preserving. Because it relies on radio propagation, however, it is inherently sensitive to noise and environmental perturbations: small changes in people or objects can alter the multipath structure and thus the received signal [40].

2.1.1 Channel State Information (CSI)

In a MIMO-OFDM[1] system, the received signal $\mathbf{y}$ can be modeled as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\eta}, \tag{2.1}$$

[1]MIMO employs multiple antennas at both the transmitter and receiver ends, creating a matrix of connections between every transmitter and receiver. Orthogonal Frequency Division Multiplexing (OFDM) divides the channel bandwidth into multiple smaller, non-overlapping frequency bands, called subcarriers, which transmit bit streams in parallel.

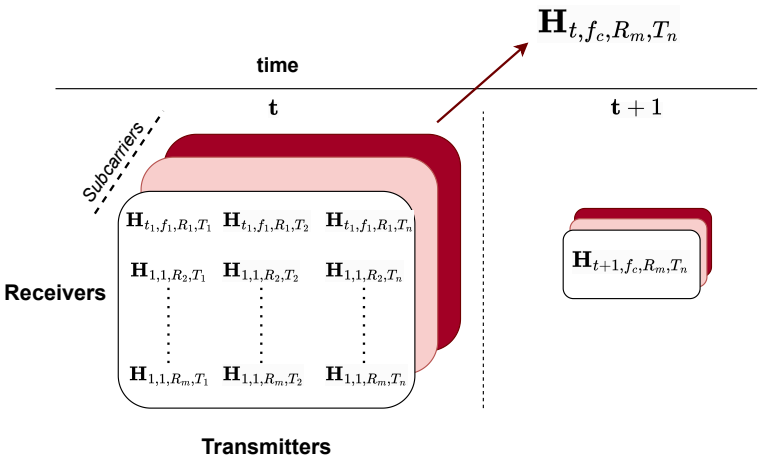


Figure 2.1: Structure of the CSI matrix, illustrating dimensions across receivers, transmitters, subcarriers, and time.

ewhere $\mathbf{x}$ is the transmitted signal vector, $\boldsymbol{\eta}$ is the additive noise vector, and $\mathbf{H}$ is the CSI matrix, capturing the properties of the wireless channels between the transmitter and receiver. Each element $H_{m,n}(f_c;t)$ in $\mathbf{H}$ represents the channel impulse response (CIR) between the n-th transmitter antenna and the m-th receiver antenna at a given frequency f_c and time t, encapsulating environmental effects such as path loss, obstacles, and interference. The CSI matrix element $H_{m,n}(f_c;t)$ can be expressed in terms of the CIR as follows [34]:

$$H_{m,n}(f_c;t) = \sum_{l=1}^{L} a_l(t) e^{-j2\pi f_c \tau_l(t)}, \tag{2.2}$$

where L is the number of multipath components, $a_l(t)$ represents the amplitude attenuation, and $\tau_l(t)$ is the propagation delay for each path. The amplitude $|H_{m,n}(f_c;t)|$ and phase $\angle H_{m,n}(f_c;t)$ of CSI provide detailed information about multipath propagation. The structure of the CSI matrix at a given receiver depends on the number of receiving and transmitting antennas, subcarriers, and timestamps in the wireless

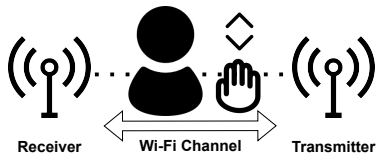
communication system. It can be visualized as a multidimensional array, where each element provides detailed channel information for a specific combination of time, subcarrier, transmit antennas, and receive antennas. Wi-Fi devices obtain accurate CSI measurements by utilizing *Long Training Fields* (LTFs). By comparing the received LTFs with the known transmitted patterns, the receiver can estimate the CSI, effectively producing a "Wi-Fi image" of the surrounding environment [40, 69].

As shown in **Fig. 2.1**, the hierarchical structure of the CSI matrix is depicted, illustrating how it evolves and frequency. At each time step t, the matrix H_{t,f_c,R_m,T_n} captures wireless channel measurements between multiple transmit antennas T_n and receive antennas R_m across different subcarrier frequencies f_c. Each element of the matrix represents a complex channel coefficient containing both amplitude and phase information, characterizing the wireless channel between a specific transmitter-receiver pair at a specific frequency and time.

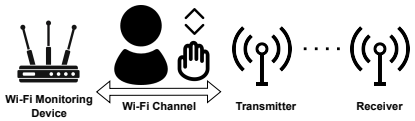
Fig. 2.1 demonstrates that the CSI matrix can be thought of as layers stacked over time and frequency. Each layer corresponds to a different subcarrier frequency, and within each layer, the grid formed by the receivers and transmitters provides a detailed snapshot of the channel characteristics at that specific time and frequency. Although the figure presents an example with two time instances, this structure can extend across any number of timestamps and subcarriers.

2.1.2 Active and Passive CSI Measurements

Collecting CSI data involves configuring transmitters and receivers, with at least one Wi-Fi network interface card (NIC) running modified firmware to support CSI extraction. This can be done using either active or passive sensing methods, depending on



(a) Active Wi-Fi Sensing



(b) Passive Wi-Fi Sensing

Figure 2.2: Active sensing and passive sensing setups for CSI data collection.

the application and setup. In active sensing, a transmitter sends predefined packets to a receiver, which extracts and stores CSI data. This approach requires only two devices and offers precise control over the signal, as the devices' positions are known and packet rates can be adjusted for greater accuracy. Active sensing is ideal for scenarios needing consistent and predictable signal patterns. However, it consumes bandwidth, rendering the transmitter unavailable for other tasks. **Fig. 2.2(a)** illustrates an active Wi-Fi sensing setup, where CSI captures the channel between the transmitter and receiver.

Passive sensing, on the other hand, leverages existing Wi-Fi traffic without introducing additional transmissions. A monitoring device collects CSI from ongoing communications and estimates the channels to transmitters with known MAC addresses. This setup typically involves at least three devices. Passive sensing minimizes interference and is suitable for scenarios where active transmission is impractical or disruptive. Passive sensing is advantageous due to its lower bandwidth usage and reduced power consumption, making it ideal for applications like HAR and smart

environments where non-intrusiveness is critical. However, it poses challenges, such as the need for precise device synchronization, lower packet rates, and the possibility of missed packets due to dependence on existing traffic. **Fig. 2.2(b)** shows a passive Wi-Fi sensing configuration, where CSI captures the channel between a monitoring device and a transmitter.

2.1.3 Wi-Fi CSI for Moving Users

To understand the effect of a moving point, such as a human hand, on the Wi-Fi CSI phase, we can consider a scenario in which an object is positioned between a Wi-Fi transmitter and receiver, see **Fig. 2.3**. As the object moves upward at a constant velocity v over a duration of Δt seconds, the altered signal path introduces a time delay $\Delta\tau$, which in turn causes a phase shift. Accounting for this delay, the CSI $h(f;t)$ for a subcarrier with carrier frequency f_c at time $t+\Delta t$ can be expressed as follows:

$$h\left(f_c; t+\Delta t\right) = h\left(f_c; t\right) \exp\left(-j2\pi f_c \Delta\tau\right), \tag{2.3}$$

in which $\Delta\tau$ is

$$\Delta\tau = \frac{\Delta d(v)}{c}, \tag{2.4}$$

where c represents the propagation speed of Wi-Fi signals through the air and $\Delta d(v)$ corresponds to the displacement resulting from the object's motion [1].

If there are L moving points in the space that reflect the transmitted signal to the receiver, the resulting CSI signal can be expressed as follows

$$h\left(f_c; t_0+\Delta t\right) = \sum_{i=1}^{L} h_i\left(f_c; t_0\right) \exp\left(-j4\pi f_c\left(\frac{v_i \Delta t}{c}\right)\right), \tag{2.5}$$

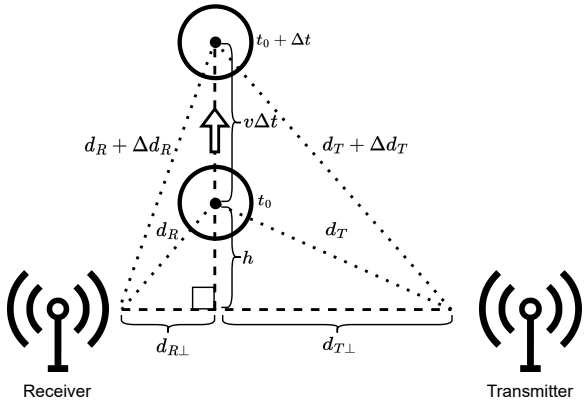


Figure 2.3: The spatial arrangement of a Wi-Fi transmitter and receiver where an object moves upward at a velocity of v [1].

where $h_i(f_c;t)$ represents the CSI from the i^{th} path at time t and carrier frequency f_c, while v_i denotes the velocity of the i^{th} object. Suppose that M samples of CSI are obtained with a sampling interval of Δt, and the instantaneous velocities of the moving points are constant. Under these conditions, **Eq. 2.5** for the CSI vector $\mathbf{h}_{f_c}$, consisting of M elements, can be represented in matrix form as follows:

$$\mathbf{h}_{f_c} = \underbrace{[\mathbf{d}(v_1),\ldots,\mathbf{d}\,(v_L)]}_{\triangleq \mathbf{D}_{M\times L}} \underbrace{[h_1(t_0,f_c),\ldots,h_L(t_0,f_c)]^\top}_{\triangleq \mathbf{a}_{L\times 1}}, \tag{2.6}$$

where, for each moving point indexed by i, the velocity vector $\mathbf{d}(v_i)$, composed of M elements, is given by

$$\mathbf{d}(v_i) = \left[1, e^{-j4\pi f_c(\frac{v_i\Delta t}{c})},\ldots, e^{-j4\pi(M-1)f_c(\frac{v_i\Delta t}{c})}\right], \tag{2.7}$$

Thus, $\mathbf{D}$ represents the $M \times L$ matrix of velocity vectors. Thus, considering a random

noise vector $\mathbf{n}$ with M elements, **Eq. 2.6** can be reformulated as follows:

$$\mathbf{h}_{f_c} = \mathbf{D}\mathbf{a} + \mathbf{n}. \tag{2.8}$$

where $\mathbf{a}$ represents the vector of initial CSI at the reference time t_0. The equation above establishes a relationship between the collected CSI samples and the velocity of objects moving in the environment. This suggests that velocities, referred to as Doppler velocity, can be estimated from CSI samples $\mathbf{h}_{f_c}$, thus relating CSI values to the human activities performed within the environment.

2.1.4 Domain Shift in CSI-Based Wi-Fi Sensing

Wi-Fi sensing exploits CSI from commodity devices for privacy-preserving HAR. In a MIMO-OFDM system with N_t transmit and N_r receive antennas over N_{sc} subcarriers, the received signal on subcarrier k at time t is $\mathbf{y}_k(t) = \mathbf{H}_k(t)\,\mathbf{s}_k(t) + \boldsymbol{\eta}_k(t)$, where $\mathbf{s}_k(t) \in \mathbb{C}^{N_t}$ is the transmitted symbol vector, $\boldsymbol{\eta}_k(t) \in \mathbb{C}^{N_r}$ denotes additive noise, and $\mathbf{H}_k(t) \in \mathbb{C}^{N_r \times N_t}$ is the CSI matrix. Let $H_{m,n}(k,t)$ denotes the CSI from transmit antenna $n \in \{1,\ldots,N_t\}$ to receive antenna $m \in \{1,\ldots,N_r\}$ on subcarrier k at time t, so that $\mathbf{H}_k(t) = [H_{m,n}(k,t)]_{m=1,n=1}^{N_r,N_t}$. Each CSI entry representing the complex channel frequency response (CFR) is $H_{m,n}(k,t) = |H_{m,n}(k,t)|\, e^{j\phi_{m,n}(k,t)}$, where $\phi_{m,n}(k,t) \triangleq \angle H_{m,n}(k,t)$. Commodity NICs expose CSI as tensors, i.e.,

$$\mathbf{x}_{\text{raw}} \triangleq [H_{m,n}(k,t)]_{t,m,n,k} \in \mathbb{C}^{T \times N_r \times N_t \times N_{sc}}, \tag{2.9}$$

across time, antennas, and subcarriers [59]. We use $\mathbf{x} \in \mathbb{R}^{1 \times F \times T}$ to denote its pre-processed representation (e.g., amplitude or phase extraction with spatial flattening,

detailed in Section 3), where $F = N_{sc} \cdot N_r \cdot N_t$ is the flattened dimension.

Indoor propagation is dominated by multipath effects. A common model that expresses the CFR as a superposition of L propagation paths is given below:

$$H_{m,n}(k,t) = \sum_{\ell=1}^{L} \alpha_{\ell,mn}(t)\, e^{-j2\pi f_k \tau_{\ell,mn}(t)}\, e^{j2\pi f_{D,\ell} t}, \tag{2.10}$$

where f_k is the center frequency of subcarrier k, subscript mn denotes the transmit–receive antenna pair (m,n), $\alpha_{\ell,mn}(t) \in \mathbb{C}$ and $\tau_{\ell,mn}(t)$ are the time-varying complex attenuation and delay of path ℓ, respectively, and $f_{D,\ell}$ (Hz) is the Doppler shift producing the time-varying phase term $e^{j2\pi f_{D,\ell} t}$. Small changes in path delay induce large phase rotations, i.e., a delay perturbation $\Delta\tau$ yields

$$\Delta\phi_k \approx -2\pi f_k \Delta\tau \pmod{2\pi}. \tag{2.11}$$

Consequently, modest changes in room layout, materials, user placement, carrier frequency, or device calibration substantially alter CSI statistics, causing abrupt domain shifts in both amplitude and phase [39–41].

In multi-user scenarios, Eq. (2.10) aggregates contributions from all active users simultaneously. Unlike single-user settings where the L paths primarily reflect one person's movements, multi-user environments superpose path sets from each individual, where path parameters $\{\alpha_{\ell,mn}(t), \tau_{\ell,mn}(t), f_{D,\ell}\}$ reflect the combined influence of multiple people performing different activities [50]. This signal entanglement makes it fundamentally difficult to attribute observed CSI variations to specific users, as one person's movements directly affect how another person's activities appear in the wireless channel [51].

Domain shift compounds these multi-user challenges by altering both individual activity signatures and their interactions. Consider the cross-frequency scenario in WiMANS where models trained at 2.4 GHz must generalize to 5 GHz. The wavelength changes from 12.5 cm to 5 cm, fundamentally altering how overlapping user activities interfere in the multipath channel. From Eq. (2.11), phase rotations scale directly with carrier frequency: a given path delay $\Delta\tau$ induces phase shifts $\Delta\phi_{5\text{GHz}}/\Delta\phi_{2.4\text{GHz}} = 5.0/2.4 \approx 2.08$ times larger at the higher frequency. This frequency-dependent phase sensitivity means that entangled multi-user patterns exhibit domain shifts that cannot be addressed by single-user adaptation techniques [70].

2.2 Preprocessing Techniques for CSI Data

Data preprocessing is a critical step in ML as it directly impacts the performance and generalization of the model. Real-world data such as CSI, often contains noise, missing values, outliers, and irrelevant or redundant features, which can lead to issues like overfitting bias, and poor deployment results. Preprocessing techniques such as cleaning, normalization, feature selection, and transformation address these challenges by reducing dimensionality, enhancing data quality, and ensuring compatibility with algorithms. These steps make learning algorithms more efficient and improve the accuracy and the interpretability of results [71].

Although DL algorithms can inherently learn the underlying patterns within CSI data, their performance is significantly enhanced when the data is preprocessed to highlight relevant features and reduce noise. Preprocessing also helps mitigate the inherent nature of CSI data, such as noise, multi-path effects, and interference, which can obscure the information of interest. The complex nature of CSI often leads

to signal distortions, making it challenging to focus on specific features, such as body movement in motion recognition applications. To overcome these challenges, a variety of preprocessing techniques have been developed. These techniques range from signal transformation methods that create new feature spaces to outlier removal and dimensionality reduction strategies. Each technique is tailored to enhance or suppress irrelevant features, depending on the specific application and data characteristics. The following subsections explore some of the most commonly used preprocessing methods and their practical applications in Wi-Fi sensing.

2.2.1 Outlier Removal

Outlier removal is essential in preprocessing signals like CSI and internal measurement (IMU)[2] data, which are often affected by noise and variability. Different techniques are used to identify and remove outliers as described below.

2.2.1.1 Simple Smoothing Techniques

Simple smoothing techniques reduce random noise by averaging data points within a set window. For example, the Moving Average (MA) method replaces each data point x_i with the average of neighbouring points in a window of size w:

$$\hat{x}_i = \frac{1}{w} \sum_{j=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} x_j \tag{2.12}$$

The median filter further enhances robustness against outliers by substituting each data point with the median of its neighbours. However, these methods can sometimes

[2]An IMU is a sensor device that provides raw motion data such as linear acceleration, angular velocity, and orientation, typically via embedded accelerometers and gyroscopes.

overly smooth critical signal variations. To counter this, Weighted Moving Average (WMA) and Exponentially Weighted Moving Average (EWMA) apply weights w_j to emphasize recent data points as formulated below:

$$\hat{x}_i = \sum_{j=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} w_j x_j \quad \text{where} \quad \sum_j w_j = 1. \tag{2.13}$$

2.2.1.2 Advanced Filtering Methods

Advanced filtering methods focus on specific frequency components in the signal. Low-pass filters (LPF) reduce high-frequency noise by selecting an appropriate cutoff frequency f_c, which helps preserve essential signal components. Wavelet filters, using the Discrete Wavelet Transform (DWT), break down the signal into various frequency bands, enabling precise noise reduction by retaining only selected wavelet coefficients.

2.2.1.3 Anomaly Detection Techniques

Anomaly detection methods identify and correct data points that deviate from expected patterns. The Hampel Filter flags outliers by comparing each data point x_i to the median m_i of neighbouring points and replacing points where $|x_i - m_i|$ exceeds a threshold. Another method, the Local Outlier Factor (LOF) [4] assesses the local density around each point, marking outliers as those with substantially lower density compared to their neighbours.

2.2.2 Phase offset Compensation

In practical Wi-Fi systems, various sources of noise arise from the transmitter and receiver processing stages, as well as hardware and software imperfections. These noise

factors significantly affect the phase of the measured CSI between transmitting and receiving antennas, introducing complexities such as Cyclic Shift Diversity (CSD), Sampling Time Offset (STO), Sampling Frequency Offset (SFO), and beamforming-induced phase variations. Since Wi-Fi sensing applications rely on extracting the underlying multipath channel information to detect changes in the environment, addressing these phase distortions through appropriate signal processing techniques is crucial [72]. A few important techniques are outlined below.

2.2.2.1 Common Phase Error (CPE) Compensation

In CPE compensation, it is assumed that each subcarrier's phase is affected by a common phase error θ_{CPE} in addition to the actual phase θ_n of subcarrier n:

$$\tilde{\theta}_n = \theta_n + \theta_{\text{CPE}} \tag{2.14}$$

CPE compensation estimates θ_{CPE} and subtracts it from the measured phase $\tilde{\theta}_n$ across all subcarriers:

$$\hat{\theta}_n = \tilde{\theta}_n - \hat{\theta}_{\text{CPE}} \tag{2.15}$$

where $\hat{\theta}_n$ denotes the compensated phase, aligning phase data across subcarriers to improve accuracy in subsequent analysis.

Phase Difference Method (PDM) calculates phase differences between adjacent subcarriers or time samples. For adjacent subcarriers with phases $\tilde{\theta}_n$ and $\tilde{\theta}_{n+1}$, the phase difference $\Delta\tilde{\theta}_{n,n+1}$ is given by:

$$\Delta\tilde{\theta}_{n,n+1} = \tilde{\theta}_{n+1} - \tilde{\theta}_n \tag{2.16}$$

This method cancels out constant phase offsets, leaving stable phase differences $\Delta\theta_{n,n+1}$ that are beneficial for applications like motion detection and device-free localization [73, 74].

2.2.2.2 CSI Ratio Model

The CSI Ratio Model is another technique for mitigating common phase noise and compensating for phase offsets in Wi-Fi routers, particularly for those equipped with multiple antennas [75]. In such systems, antennas are placed in close proximity, causing them to experience nearly identical phase shifts due to common noise sources like oscillator drift or environmental factors. By computing the ratios of the CSI measurements between pairs of antennas, the common phase components cancel out, isolating the relative phase differences caused by unique path effects to each antenna. Mathematically, the CSI ratio model is expressed as:

$$\frac{H_i(f)}{H_j(f)} = \frac{|H_i(f)|\, e^{j[\theta_i(f)+\phi_c(f)]}}{|H_j(f)|\, e^{j[\theta_j(f)+\phi_c(f)]}} = \frac{|H_i(f)|}{|H_j(f)|}\, e^{j[\theta_i(f)-\theta_j(f)]} \tag{2.17}$$

In this equation, $H_i(f)$ and $H_j(f)$ are the CSI measurements at frequency f for antennas i and j, respectively. The magnitudes of these measurements are $|H_i(f)|$ and $|H_j(f)|$, while $\theta_i(f)$ and $\theta_j(f)$ represent the unique phase shifts for each antenna. The term $\phi_c(f)$ denotes the common phase noise experienced by all antennas, and j is the imaginary unit. By taking the ratio, the common phase term $\phi_c(f)$ cancels out, allowing for better phase offset compensation based on the relative phase differences $\theta_i(f) - \theta_j(f)$. Though this technique does not provide absolute phase values, they effectively capture relative phase changes, enhancing the utility of phase information for ML algorithms in Wi-Fi sensing.

2.2.3 Dimension Reduction / Compression

Dimension reduction is crucial for managing the high dimensionality of CSI data. It reduces noise by eliminating irrelevant features, simplifying models to be less prone to overfitting and more computationally efficient.

2.2.3.1 Principal Component Analysis (PCA)

PCA transforms high-dimensional data into a smaller set of orthogonal components while retaining as much variance as possible. Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, PCA finds a projection matrix $\mathbf{W} \in \mathbb{R}^{p \times k}$ to map $\mathbf{X}$ into a lower-dimensional space:

$$\mathbf{Z} = \mathbf{X}\mathbf{W} \tag{2.18}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ represents the reduced data. To improve data quality, it's often useful to remove outliers before applying PCA, as it is sensitive to extreme values [10].

2.2.3.2 Low-Rank Factorization (LRF)

Low-Rank Factorization approximates the data using lower-rank matrices. For a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, LRF identifies matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{p \times r}$ such that:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^{\top} \tag{2.19}$$

This technique is useful for applications like matrix completion, offering greater flexibility than PCA for targeted dimensionality reduction. These dimension reduction techniques enhance CSI data processing, improving both model performance and computational efficiency.

2.2.4 Signal Transform Methods

Signal transforms are fundamental for uncovering activity-related patterns or generating more informative representations suitable for training ML models. Key techniques include the Fast Fourier Transform (FFT), Short-Time Fourier Transform (STFT), and Wavelet Transform (WT) for the time-frequency analysis of CSI measurements. Additionally, specialized approaches such as Doppler Velocity Extraction and the Body-coordinate Velocity Profile (BVP) for estimating movement velocity and providing a robust representation of static environmental factors, as well as Random Convolutional Kernels for comprehensive feature extraction, are employed.

2.2.4.1 Fast Fourier Transform (FFT)

FFT is widely used to convert a time-domain signal $x(t)$ into its frequency-domain representation $X(f)$:

$$X(f) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi fn/N} \tag{2.20}$$

where N is the number of points, f is the frequency, and j is the imaginary unit. FFT reveals dominant frequencies in the signal and, when combined with Low Pass Filters (LPF) or Bandpass Filters (BPF), effectively removes high-frequency noise or targets specific frequency ranges, making it useful for applications like motion detection and breath estimation [10].

2.2.4.2 Short-Time Fourier Transform (STFT)

STFT analyzes the frequency content of a signal over time by applying FFT to short, overlapping segments of the signal $x(t)$ with a windowing function $w(t)$:

$$\text{STFT}(t, f) = \sum_{n=-\infty}^{\infty} x(n)w(n-t)e^{-j2\pi fn} \tag{2.21}$$

This method provides a time-frequency representation, useful for identifying patterns that vary over time. However, STFT encounters trade-offs between time and frequency resolution.

2.2.4.3 Wavelet Transform (WT)

The Wavelet Transform offers better frequency resolution for low frequencies and better time resolution for high frequencies. The Discrete Wavelet Transform (DWT) decomposes a signal into multiple resolution levels using high-pass and low-pass filters:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \phi_k(t) + \sum_{k=-\infty}^{\infty} \sum_{j=1}^{J} d_{j,k} \psi_{j,k}(t) \tag{2.22}$$

where c_k are approximation coefficients, $d_{j,k}$ are detail coefficients, $\phi_k(t)$ is the scaling function, and $\psi_{j,k}(t)$ are the wavelet functions at scale j. DWT is effective for noise reduction and offers robustness beyond techniques relying solely on Doppler phase shift [76].

2.2.4.4 MUSIC

CSI and the velocity of movement or activity are theoretically related, as shown in Eq.2.8. This relationship suggests that velocities of moving objects within the

environment can be determined from CSI using AoA estimation methods, such as Multiple Signal Classification (MUSIC) [77]. MUSIC leverages the eigen structure of the covariance matrix of the CSI data to estimate the movements Doppler velocity vector by separating the signal subspace from the noise subspace. The Doppler velocity pseudo-spectrum $P_{\text{MUSIC}}(v)$ is given by:

$$P_{\text{MUSIC}}(v) = \frac{1}{\mathbf{d}^H(v)\mathbf{Q}_n\mathbf{Q}_n^H\mathbf{d}(v)} \tag{2.23a}$$

where $\mathbf{d}$ is the steering vector for velocity v, and $\mathbf{Q}_n$ is the matrix of all eigenvectors forming the noise subspace. The denominator becomes zero when v corresponds to the velocity of one of the moving points. Therefore, the sharp peaks in the pseudo-spectrum provide the estimated velocities of the moving points in the environment.

2.2.4.5 Body-coordinate Velocity Profile (BVP)

BVP, introduced in systems like Widar3.0 [16], is designed for human motion analysis by leveraging unique velocity distributions across body parts during activities. A key parameter in BVP is the Doppler Frequency Shift (DFS), calculated as:

$$f_D = \frac{v \cdot f_c}{c} \tag{2.24}$$

where f_D is the Doppler shift, v is relative velocity, f_c is the carrier frequency, and c is the speed of light. While DFS provides valuable insights, it is sensitive to position and orientation. BVP addresses this by mapping signal power over body-coordinated velocity components, making it a robust indicator of activity types.

2.2.4.6 MiniRocket

Due to the typically small size of datasets in HAR tasks, deep neural networks may struggle to effectively learn patterns from limited data. Unlike DL models that require large labeled datasets, MiniRocket, a highly effective method for time-series analysis, enables feature extraction without the need for labels, making it particularly advantageous in scenarios with scarce data [78].

In the MiniRocket method, convolution is applied to extract features from time-series data, such as CSI signals, by using random kernels. Each kernel k shifts along the input signal x, generating an output $(x * k)_i$, which captures localized patterns. These diverse transformed outputs are summarized using the Proportion of Positive Values (PPV), calculated as:

$$\text{PPV} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left((x * k)_i > 0\right) \tag{2.25}$$

where N is the number of elements in the convolution output, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 when the condition inside is true. These extracted features, which in the default setting amount to $9,996$ features, can then be used as inputs to a simple linear classifier.

2.3 Existing CSI Datasets

Publicly available datasets play a vital role in advancing CSI-based Wi-Fi sensing research, offering diverse environments, hardware setups, and applications. These datasets generally fall into three categories: *HAR*, which targets broad, complex actions involving the whole body; *Gesture Recognition and Sign Language*, focused

on discrete, repetitive movements of specific body parts; and *Health Monitoring and Specialized Datasets*, designed for applications like health tracking or unique environments. Most datasets consist of time-series data with labeled entries corresponding to activities or gestures. Typically, this data includes CSI values, capturing signal amplitude and phase across multiple antennas and subcarriers, often supplemented by RSSI values and timestamps. In the following, we list the most popular datasets and their specifications. **Table 2.1** provides a comparative summary of these datasets in terms of the hardware, extraction tool, and specifications.

2.3.1 HAR Datasets

In what follows, we organize datasets mainly designed for HAR applications.

- **UT-HAR Dataset:** The UT-HAR dataset [11] was collected in an indoor office environment using an Intel 5300 NIC with the Linux 802.11n CSI tool. It includes six activities (e.g., walking, sitting) performed by six individuals under LOS conditions. Data columns contain timestamps, 90 amplitude values across 30 subcarriers and 3 R_m antennas operating at 5 GHz frequency, followed by 90 corresponding phase values, resulting in a CSI format of 1 x 250 x 90 with an approximate 4 GB size. Following the IEEE 802.11n/ac standard, this dataset, despite challenges with noise and variability, is valuable for developing activity recognition models and is available on GitHub.

- **WiAR Dataset:** The WiAR dataset [79] comprises 16 activities performed by 10 volunteers across three indoor environments, collected with an Intel 5300 NIC operating on IEEE 802.11n standard at 5 GHz with 20 MHz bandwidth. It includes raw RSSI and CSI data across 30 subcarriers, 1 T_n and 3 R_m antennas

with an approximate 5 GB size. Each row represents time-specific data consisting of two periods, active (human activity is being performed) and empty (background data with no activity). This structure helps in smart homes and healthcare environments by enabling activity recognition in a dynamic environment.

- **WiCount Dataset:** The WiCount dataset [80] captures CSI data with detailed amplitude and phase information across 30 subcarriers, 2 T_n and 3 R_m antennas, collected using an Intel NUC laptop and mini R1C router operating on IEEE 802.11n standard at 5 GHz with 20 MHz bandwidth. This dataset allows volunteers free movement, enhancing its applicability in scenarios where precise movement tracking is essential. It is available on GitHub.

- **WiMANS Dataset:** The WiMANS dataset [51] is designed for multi-user activity sensing in smart homes and security contexts, capturing interactions of up to five users in dynamic environments. Using an Intel 5300 NIC operating on IEEE 802.11n standard with 3 T_n transmitting and 3 R_m receiving antennas, it collects CSI data across 30 subcarriers at both 2.4 GHz and 5 GHz frequencies. The dataset provides over 9.4 hours of CSI data synchronized with video recordings. The dataset, 1.44 GB in size, is accessible on GitHub.

- **FallDeFi Dataset:** FallDeFi [81] supports activity and fall detection applications. It includes CSI traces for activities such as falling, walking, and sitting, collected with Intel 5300 NICs operating on IEEE 802.11n standard at 5 GHz with 20 MHz bandwidth, using 2 T_n transmitting and 2 R_m receiving antennas across 30 subcarriers in typical indoor spaces (bedrooms, corridors). This

time-sequence data is valuable for fall detection research and is available on GitHub.

2.3.2 Gesture Recognition and Sign Language Datasets

In what follows, we organize datasets mainly designed for gesture recognition and sign language applications.

- **Widar3.0 Dataset:** Introduced in 2019, the Widar3.0 dataset [65] focuses on gesture recognition with CSI data collected from 16 users across various indoor environments using Intel 5300 NIC utilizing 1 T_n, and 18 R_m across 30 subcarriers at 5.8 GHz. Its Body-coordinate Velocity Profile (BVP) feature enhances recognition accuracy in diverse settings, making it suitable for applications needing robust performance. The dataset is 3.4 GB and available online[3].

- **SignFi Dataset:** The SignFi dataset [15] is the first to enable sign language recognition using CSI. It includes 276 gestures from 5 users, captured in lab and home environments with a 5 ms sampling rate and noise preprocessing to ensure accuracy. Collected using Intel 5300 NIC following IEEE 802.11n at 5 GHz and 20 MHz frequency, utilizing 3 T_n and 1 R_m antennas. The dataset is 1.44 GB and accessible online[4].

- **Wi-SL Dataset:** The Wi-SL dataset [82] enables fine-grained gesture recognition through Wi-Fi CSI, without requiring wearable devices. Collected using TP-Link WDR4310 routers operating on IEEE 802.11n standard at 5 GHz with

[3] `http://tns.thss.tsinghua.edu.cn/widar3.0/index.html`
[4] `https://yongsen.github.io/SignFi/`

40 MHz bandwidth, utilizing 2 T_n transmitting and 3 R_m receiving antennas across 114 subcarriers, with CSI extracted using the Atheros tool. It provides a non-invasive, cost-effective option suitable for smart home and assistive applications. Preprocessing may be necessary to mitigate challenges such as noise and multipath effects.

- **CrossSense Dataset:** The CrossSense Dataset [83] comprises 1,200,000 activity samples from 100 users performing 40 gestures, captured using both Intel 5300 NIC and TP-Link WDR7500 devices operating on IEEE 802.11n at 5 GHz. The system utilizes 3 T_n transmitting and 3 R_m receiving antennas across 30 subcarriers. Focused on gesture recognition and human-computer interaction, its robust design enables reliable performance across diverse scenarios and large-scale implementations. The dataset is available on GitHub.

- **DeepSeg Dataset:** DeepSeg [84] supports both fine-grained and coarse-grained HAR. Collected in a meeting room setting using a Linux-based Intel 5300 NIC operating on IEEE 802.11n standard with 1 T_n transmitting and 3 R_m receiving antennas across 30 subcarriers, it includes data for 10 activities performed by 5 volunteers, making it versatile for model evaluation. The dataset is available on GitHub.

2.3.3 Health Monitoring and Specialized Datasets

In what follows, we organize datasets mainly designed for health monitoring applications.

- **eHealth CSI Dataset:** The eHealth CSI dataset [85] provides high-resolution

CSI data combined with phenotype and heartbeat information for health monitoring applications. Using a Raspberry Pi 4B operating on IEEE 802.11n standard at 5 GHz with 80 MHz bandwidth, the system captures data using 1 T_n transmitting and 1 R_m receiving antenna across 234 subcarriers. Collected in a controlled setting, this dataset supports applications in physical analysis and activity recognition. Access requires a request form.

- **IEEE 802.11ac 80 MHz CSI Dataset:** This dataset [86] includes channel measurements over an 80 MHz and 5 GHz bandwidth, suited for HAR and person identification. Using 3 T_n and 3 Rm antennas across 242 subcarriers have been collected across seven diverse environments, it offers time diversity and spans 23.6 GB. Available for download via GitHub.

- **Wi-Fi CSI-based HAR Dataset:** Supporting smart home automation and security, this dataset [87] captures activities in three rooms, with each activity labeled with an image and bounding box. Data collection utilized 2 TP-Link TL-WDR4300 routers with 3 T_n transmitting and 3 R_m receiving antennas, operating on dual bands: 2.4 GHz with 20 MHz bandwidth across 56 subcarriers, and 5 GHz with 40 MHz bandwidth across 114 subcarriers. The dataset is available on GitHub in two versions: 1.2 GB without images or 9.1 GB with images.

- **CSI Human Activity Dataset:** Collected in rooms of approximately 4 m $\times$ 4.5 m, this dataset [88] supports healthcare and smart living applications. Data collection employed Raspberry Pi 3B+ and 4B with ASUS RT-AC86U

routers, utilizing 2 T_n transmitting and 2 R_m receiving antennas, with CSI extracted using the Nexmon tool. The system operates in two configurations: 128 subcarriers at 40 MHz bandwidth and 256 subcarriers at 80 MHz bandwidth. Although inherently noisy, this 23.6 GB dataset provides comprehensive data for activity recognition tasks and is available on GitHub.

- **WiADG Dataset:** The WiADG dataset [89] focuses on gesture recognition using device-free Wi-Fi, featuring two TP-Link TL-WDR4300 routers on IEEE 802.11n at 5 GHz with 40 MHz bandwidth. The setup incorporates 3 T_n transmitting and 3 R_m receiving antennas, spanning 114 subcarriers, with CSI extracted using the Atheros tool. Collected in two indoor environments, it is optimized for gesture recognition via adversarial domain adaptation and is accessible on GitHub.

2.3.4 Other Datasets

- **ZigFi Dataset:** ZigFi [90] was collected using Intel 5300 NICs and TelosB motes in environments with varying noise, including offices during day and night. It is valuable for studying environmental impacts on CSI. The data is available on GitHub.

- **EyeFi Dataset:** The EyeFi dataset [91] combines vision data from a Bosch Flexidome IP Panoramic Camera with Wi-Fi data from an Intel 5300 NIC. Collected in lab and kitchen settings, it contains over 1.2 million Wi-Fi packets and is valuable for multi-modal research. Available on Zenodo.

Table 2.1: Publicly Available Datasets

Dataset Name	**Ref.**	**Year**	**Extraction Tool**	**Hardware**	**Specifications**
UT-HAR (OP)	[11]	2017	Linux 802.11n	Intel 5300 NIC	3 R_m Antennas, 30 Subcarriers Per Antenna, IEEE 802.11n/ac (5 GHz)
WiCount (OP)	[80]	2017	Linux 802.11n	Intel 5300 NIC, Mini R1C	2 T_n, 3 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5 GHz, 40 MHz)
CrossSense (OP)	[83]	2018	Linux 802.11n	Intel 5300 NIC, TP-Link WDR7500	3 T_n, 3 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5 GHz)
WiADG (OP)	[89]	2018	Atheros CSI	2 TP-Link TL-WDR4300	3 T_n, 3 R_m Antennas, 114 Subcarriers, IEEE 802.11n (5 GHz, 40 MHz)
FallDeFi (OP)	[81]	2018	Linux 802.11n	Intel 5300 NIC	2 T_n, 2 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5 GHz, 20 MHz)
SignFi (OP)	[15]	2018	OpenRF 802.11n	Intel 5300 NIC	3 T_n, 1 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5 GHz, 20 MHz)
CSI Human Activity	[88]	2019	Nexmon CSI	Raspberry Pi 3B+/4B, ASUS RT-AC86U	2 T_n, 2 R_m Antennas, 128 Subcarriers (40 MHz), 256 Subcarriers (80 MHz)
WiAR	[79]	2019	Linux 802.11n	Intel 5300 NIC	1 T_n, 3 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5 GHz, 20 MHz)
Wi-SL	[82]	2020	Atheros CSI	TP-Link WDR4310	2 T_n, 3 R_m Antennas, 114 Subcarriers, IEEE 802.11n (5 GHz, 40 MHz)
DeepSeg (OP)	[84]	2020	Linux 802.11n	Intel 5300 NIC	1 T_n, 3 R_m Antennas, 30 Subcarriers, IEEE 802.11n
ZigFi (OP)	[90]	2020	Linux 802.11n	Intel 5300 NIC	3 T_n, 3 R_m Antennas, 64 Subcarriers, IEEE 802.11n (2.4 GHz, 20 MHz)
EyeFi (OP)	[91]	2020	Linux 802.11n	Intel 5300 NIC	3 R_m Antennas, 30 Subcarriers, IEEE 802.11n
Widar3.0 (OP)	[65]	2020	Linux 802.11n	Intel 5300 NIC	1 T_n, 18 R_m Antennas, 30 Subcarriers, IEEE 802.11n (5.8 GHz)
Wi-Fi CSI-based HAR	[87]	2022	Atheros CSI	2 TP-Link TL-WDR4300	3 T_n, 3 R_m Antennas, 56 Subcarriers (2.4 GHz, 20 MHz), 114 Subcarriers (5 GHz, 40 MHz)
eHealth CSI	[85]	2023	Nexmon CSI	Raspberry Pi 4B	1 T_n, 1 R_m Antenna, 234 Subcarriers, IEEE 802.11n (5 GHz, 80 MHz)
IEEE 802.11ac 80 MHz	[86]	2023	Nexmon CSI	Netgear X4S, ASUS RT-AC86U	3 T_n, 3 R_m Antennas, 242 Subcarriers, IEEE 802.11ac (5 GHz, 80 MHz)
WiMANS (OP)	[51]	2024	Linux 802.11n	Intel 5300 NIC	3 T_n, 3 R_m Antennas, 30 Subcarriers, IEEE 802.11n (2.4 GHz, 5 GHz)

Table 2.2: Recommended DL Methods for CSI-based Wi-Fi Sensing Tasks

Task Type	**Data Characteristics**	**Recommended Methods**
Temporal and Sequential Analysis	Sequential CSI Data	RNNs, LSTMs, and Transformers for capturing temporal patterns in CSI sequences [92–96]; Classification tasks that rely on temporal dependencies can also leverage these methods.
Spatial Analysis	CSI Amplitude Maps	CNNs for extracting spatial features from CSI amplitude Attention mechanisms to focus on relevant antennas or subcarriers [97–99]; Classification tasks with strong spatial dependencies can also use these methods.
Generative Modeling	Amplitude and Phase Distributions	Autoencoders for learning representations and compression of CSI data [100]; Generative usage for data augmentation or anomaly detection [101–103]
Unsupervised Learning	Unlabeled CSI Data	Clustering methods to discover patterns in CSI amplitude and phase [104]; SSL for pretraining representations [105]

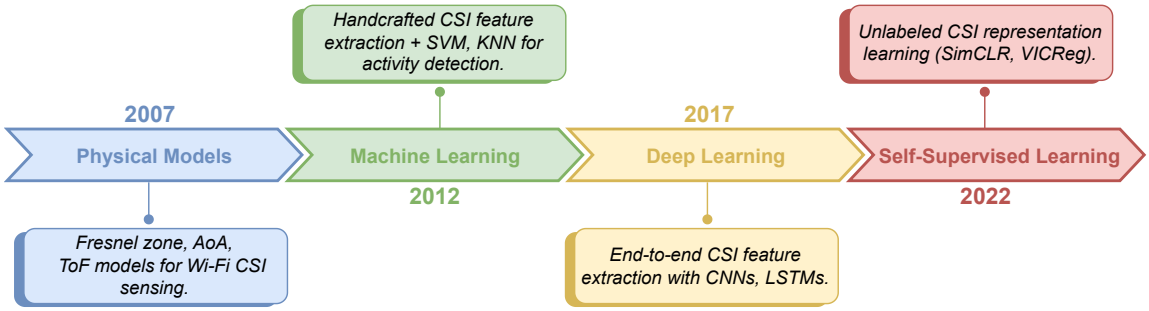


Figure 2.4: Evolution of Methodologies for Wi-Fi Sensing

2.4 Conventional Deep Learning for Wi-Fi Sensing

In this section, we provide a brief overview of the existing DL methods for CSI-based Wi-Fi sensing. As summarized in **Table 2.2**, time-independent models (e.g., CNNs, DNNs) suited for spatial feature extraction and time-dependent models (e.g., RNNs, LSTMs) are suitable for capturing temporal dynamics. Sequential tasks benefit from models like RNNs, GRUs, and Transformers, which handle temporal dependencies effectively. In contrast, spatial tasks leverage CNNs for modeling spatial correlations

across antennas and subcarriers. For scenarios with limited labeled data, unsupervised methods such as autoencoders and SSL approaches are invaluable for learning meaningful representations.

2.4.1 Time-Independent Models

Wi-Fi sensing environments are highly dynamic and require models capable of adapting to temporal shifts. Time-series datasets are influenced by external factors such as movement and interference, making time-dependent models essential for accurately capturing temporal changes. However, time-independent models process data without considering temporal dependencies, making them well-suited for tasks where sequence order does not influence outcomes.

2.4.1.1 Deep Neural Networks (DNNs)

DNNs [106] model complex nonlinear interactions through multiple layers of connected neurons, enabling them to effectively extract patterns from high-dimensional CSI data. Each layer l computes an activation $\mathbf{h}^{(l)}$ as follows:

$$\mathbf{h}^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \tag{2.26}$$

where $\mathbf{h}^{(0)} = \mathbf{x}$ (input layer) and $\mathbf{h}^{(L)} = \mathbf{y}$ (output layer). By stacking layers, DNNs extract hierarchical features, reducing the reliance on manual feature engineering. WiCount [107] employed a fully connected feed-forward neural network with two hidden layers trained on the phase and amplitude features of CSI data. This system achieved an accuracy of 82.3% in estimating up to five individuals in diverse indoor environments.

2.4.1.2 Convolutional Neural Networks (CNNs)

CNNs [108] are powerful feature extractors for spatial patterns in CSI data. The convolution operation applies a filter $\mathbf{w}$ across the input $\mathbf{x}$, producing feature maps $\mathbf{y}$:

$$\mathbf{y} = f(\mathbf{x} * \mathbf{w} + \mathbf{b}), \tag{2.27}$$

where $*$ denotes convolution, $\mathbf{b}$ is a bias term, and f is an activation function. CNNs are particularly effective in Wi-Fi sensing tasks, where spatial dependencies in CSI are critical for understanding human activity. Their ability to extract spatial features makes them indispensable for such applications. Typically, CNNs serve as feature extractors, with their output passed to a linear classifier for final prediction.

In device-free health monitoring, [97] utilized CNNs as a preprocessing technique connected to an SVM classifier for health status prediction, achieving improved accuracy. Similarly, [98] demonstrated that using CNNs as a backbone for feature extraction in HAR achieved state-of-the-art accuracy of 97.14% on the UCI HAR dataset. This outperformed traditional models such as DNNs, LSTMs, and CNN-LSTMs. The CNN-MLP hybrid approach showcased its ability to reduce computational overhead while achieving superior results in low-resource environments.

For edge-cloud scenarios, [100] proposed RSCNet, which employs dilated convolutional layers with residual connections for CSI compression at IoT devices. The lightweight encoder design enables real-time compression with ratios up to 1/90 while maintaining 97.4% sensing accuracy, demonstrating CNNs' effectiveness in resource-constrained environments for cloud-based Wi-Fi sensing.

2.4.2 Time-Dependent Models

Time-dependent models analyze sequential CSI data to uncover temporal dependencies, which are essential for dynamic tasks such as HAR.

2.4.2.1 Recurrent Neural Networks (RNNs)

RNNs [109] are designed to learn sequential patterns by maintaining internal states over time. The hidden state $\mathbf{h}_t$ at time t is computed as:

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}_h), \tag{2.28}$$

where σ is an activation function, $\mathbf{W}_h$ and $\mathbf{W}_x$ are weight matrices, and $\mathbf{b}_h$ is a bias vector. [110] proposed a fingerprinting approach for Wi-Fi-based localization using RNNs, leveraging sequential RSSI patterns to achieve an average localization error of 0.75 m, outperforming KNN and probabilistic models by 30%. Similarly, [111] developed a passive fall detection system using Wi-Fi signals. By combining RNNs with discrete wavelet transforms for preprocessing, their system significantly enhanced elder healthcare by achieving higher accuracy than traditional ML models. However, standard RNNs struggle with vanishing gradients, limiting their ability to learn long-term dependencies [112].

2.4.2.2 Long Short-Term Memory (LSTM)

LSTM networks [113] are specifically designed to overcome the limitations of standard RNNs by introducing sophisticated memory mechanisms. The LSTM architecture

uses three gates to control information flow:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad \text{(Forget gate)},$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad \text{(Input gate)},$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad \text{(Output gate)},$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \text{ (Cell state)},$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad \text{(Hidden state)}.$$

Recent applications highlight LSTM's versatility in Wi-Fi sensing. [92] improved LSTM performance by preprocessing Wi-Fi signals with Independent Component Analysis (ICA) and Continuous Wavelet Transform (CWT), achieving 97% accuracy in distinguishing multiple users' activities. For network security, [93] developed an LSTM-based system for real-time jamming detection in IoT networks, achieving 99.5% accuracy and outperforming traditional time-independent models. Furthermore, [94] introduced Wi-SensiNet, combining CNN feature extraction with Bidirectional LSTM (BiLSTM) [95] to provide forward and backward temporal context. This hybrid architecture achieved 99% accuracy in through-wall activity recognition, setting a new state-of-the-art performance.

2.4.2.3 Advanced Temporal Models

Recent architectural advancements have further refined temporal modeling, enabling better capture of short- and long-term dependencies. TCD-FERN [96] introduced a dual-network approach where one network processes immediate spatial features and

another captures movement patterns, with selective attention mechanisms determining the most relevant temporal information. This architecture achieved breakthrough performance in multi-room human presence detection by distinguishing between static and dynamic Wi-Fi signal patterns. Similarly, BiVTC [114] adapted vision transformers to Wi-Fi sensing by treating CSI measurements as image-like data, enabling precise robotic activity recognition through attention mechanisms that identify subtle signal variations. These approaches build on the fundamental attention mechanism:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.29}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are query, key, and value matrices. While attention-based models show immense promise, their computational demands pose challenges [115], prompting the development of efficient alternatives like xLSTM [116], which balance sophisticated temporal modeling with resource efficiency.

2.4.3 Unsupervised Learning Approaches

Unsupervised learning is critical when labeled data is limited or expensive to obtain. These approaches excel at tasks like anomaly detection, clustering, and feature extraction by identifying patterns in unlabeled data. Their versatility makes them well-suited for real-world deployments, requiring less preprocessing. Techniques such as clustering, autoencoders, and dimensionality reduction have significantly improved the efficiency of Wi-Fi sensing systems.

2.4.3.1 Clustering

Clustering algorithms group data points based on shared features, enabling automatic pattern discovery in CSI data without requiring labeled samples. Common applications include detecting anomalous activities, such as unusual human movements, and segmenting environmental conditions like office or home settings. A notable method, k-means clustering, minimizes within-cluster variance:

$$\arg\min_{\{\boldsymbol{\mu}_i\}_{i=1}^{k}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2, \tag{2.30}$$

where $\boldsymbol{\mu}_i$ represents the centroid of cluster C_i. Similarity between data points $\mathbf{x}$ and $\mathbf{x}'$ is often measured using the Euclidean distance:

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{j=1}^{n} (x_j - x'_j)^2}. \tag{2.31}$$

Clustering has been employed in applications such as noise identification [117] and human behavior analysis [104]. In [104], clustering is applied across three dimensions—time, person, and location—to extract distinct patterns from CSI data. Clustering by time uncovers periodic trends, clustering by person segments users based on behavioral traits, and clustering by location identifies spatial usage patterns.

2.4.3.2 Autoencoders

Autoencoders learn compact representations by encoding input $\mathbf{x}$ into a latent space $\mathbf{z}$ and reconstructing it as $\hat{\mathbf{x}}$:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}), \tag{2.32}$$

$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}). \tag{2.33}$$

These models effectively denoise, detect anomalies, and learn robust representations of environmental changes. In Wi-Fi sensing, autoencoders filter noise from CSI data, enhancing feature quality for downstream tasks. Advanced architectures, such as CNNs and DNNs are often integrated with autoencoders to process spatial and temporal patterns in CSI data.

Autoencoders are typically paired with methods like few-shot learning. SwinFi [102] incorporates an autoencoder architecture to compress and reconstruct CSI data, reducing computational and communication overhead. Similarly, AutoSen [101] utilizes a cross-modal autoencoder to enhance Wi-Fi sensing by linking CSI amplitude and phase, eliminating noise, and extracting robust features. These approaches demonstrate the versatility of autoencoders in optimizing both raw data quality and downstream processing efficiency.

Table 2.3: Summary of Existing Works on Domain Adaptation for Wi-Fi Sensing

Reference	Task	Environment	Input	Multi-User	SSL at Target	Agnostic Source Model	Agnostic Preprocessing	Open Access Data
Unsupervised Domain Adaptation (UDA)								
[118]	Gesture Recognition	Lab (Classroom, Hall, Office)	Phase Ratio (BVP)	✗	✗	✓	✓	Widar 3.0
[44]	User Authentication & HAR	Residential Apartment & Office	Amplitude (Time Domain& Spectrogram)	✗	✗	✓	✗	✗
WiAi-D [46]	Person Recognition	Indoor Floor	Amplitude	✗	✗	✗	✗	✗
Fidora [45]	Localization	Office	Amplitude	✗	✗	✗	✗	✗
DF-Loc [47]	Localization	Classroom & Office	Amplitude & Phase	✗	✗	✗	✗	✗
Source-Free Unsupervised Domain Adaptation (SFUDA)								
Wi-SFDAGR [52]	Wi-Fi Gesture Recognition	Lab (Classroom, Hall, Office)	Phase Ratio	✗	✗	✓	✗	XRF55, Widar 3.0
MU-SHOT-Fi	**HAR & Gesture Recognition**	**Lab (Classroom, Hall, Meeting Room)**	**Amplitude,✓ Phase Ratio**	✓	✓	✓	✓	WiMANS, Widar 3.0

2.5 Domain Adaptation in Wi-Fi Sensing

To date, a handful of research works addressed domain generalization in single-user Wi-Fi sensing using unsupervised domain adaptation (UDA) techniques. UDA employs labeled source data and unlabeled target data for training, reducing source–target discrepancy while preserving task-discriminative structure. Most UDA solutions follow a "learn-and-align" paradigm where a backbone is trained with labeled source data while the loss function encourages target features to match the source distribution.

Recently, [44] proposed an environment-independent user authentication system

using CSI amplitude and spectrograms processed by CNN-based feature extractors. They train two classifiers (user identity and HAR) with adversarial domain adaptation: a domain discriminator predicts the data domain from learned features while the feature extractor is simultaneously trained to prevent discriminator success. This minimax objective retains identity- and activity-relevant information while suppressing environment-specific cues. Similarly, WiAi-ID [46] adopted adversarial domain adaptation for passive person identification, making identity features robust to appearance changes (clothing, carried items) rather than environmental changes, using CSI amplitude from single-person walking traces. Both methods evaluate on single-user scenarios.

Adversarial domain adaptation is most effective when domain shifts alter irrelevant signal characteristics while label-relevant features remain stable across domains. In multi-user HAR, however, CSI reflects an entangled superposition of multiple users' motions, and environment- or frequency-dependent variations are tightly coupled with the cues required for user separation [42, 43]. Enforcing domain invariance in this setting can suppress information essential for disentangling users, leading to negative transfer[5]. In contrast, single-user settings better satisfy the assumptions of adversarial adaptation, as identity- and activity-relevant features can be isolated.

Beyond adversarial alignment, reconstruction-based objectives have been explored for UDA. Fidora [45] addressed indoor localization by classifying which of eight predefined locations a person occupies based on CSI fingerprints, using a variational autoencoder (VAE) for CSI data augmentation and a joint classification-reconstruction

[5]Negative transfer occurs when adaptation degrades performance: the adapted model performs worse than a non-adapted baseline (e.g., source-only) because the adaptation objective suppresses label-relevant information along with domain-specific factors.

network for domain adaptation across environmental changes. However, reconstruction methods assume stable, deterministic mappings between CSI patterns and spatial/activity representations. Multi-user scenarios violate this assumption through signal entanglement [50]—the same CSI pattern can correspond to different user-activity configurations depending on the number of users, their relative positions, and movement synchronization.

DF-Loc [47] improved localization accuracy with Multi-Source UDA (MUDA), leveraging multiple environments as source domains. Using both CSI amplitude and phase, DF-Loc employed two-stage alignment: first aligning feature distributions, then aligning regressor outputs for prediction consistency, plus adversarial learning with domain discriminators to enhance domain-invariant feature extraction. However, multi-source transfer risks negative transfer when sources poorly match the target. In multi-user scenarios, variable occupancy across source domains introduces biased activity distributions that degrade adaptation performance.

Lastly, [118] presented one of the first UDA methods for RF-based gesture recognition on the Widar 3.0 dataset. They extracted body-coordinate velocity profiles (BVP) from CSI phase ratios and employed (1) pseudo-labeling to generate pseudo-labels for unlabeled target data, enabling cross-entropy training, and (2) consistency regularization that enforces prediction consistency between original BVP features and augmented versions. However, pseudo-labeling can lead to class imbalance bias—converging toward dominant classes when the dataset is skewed. This is also a challenge for multi-user scenarios, where variable user counts create imbalanced activity distributions where empty-room or single-user samples may dominate, causing the model to under-represent occupancy states and degrade performance on classes

with less samples.

Overall, existing UDA solutions focus exclusively on single-user cases and assume access to source domain data, raising privacy concerns when data is confidential. Moreover, adversarial learning while effective for single-user alignment—does not guarantee applicability in multi-user settings due to signal entanglement that violates domain discriminator assumptions.

More recently, source-free unsupervised domain adaptation (SFUDA) has emerged as an alternative that adapts pretrained source models using only unlabeled target data [119]. SFUDA methods typically employ entropy minimization, pseudo-labeling, or neighborhood consistency [119, 120]. However, applying these to multi-user Wi-Fi sensing introduces three fundamental challenges: (1) class imbalance: standard information maximization treats all output dimensions uniformly, but "no person" slots dominate when users are less than M [51], causing collapse toward the dominant class; (2) permutation invariance: clustering-based pseudo-labeling cannot specify which activity belongs to which slot, while Hungarian matching requires explicit slot-level targets unavailable during UDA; (3) signal entanglement: neighborhood consistency assumes stable feature-to-label mappings, but multi-user CSI reflects superposed activities where identical features may correspond to different user configurations [50].

Wi-SFDAGR [52] addresses single-user gesture recognition through clustering-based adaptation with an attraction–dispersion objective and uncertainty-weighted neighbor selection based on prediction entropy. While effective for single-user settings with balanced gesture distributions, this approach assumes (i) samples with similar features belong to the same class, enabling direct cluster-to-label assignment, and (ii) each sample maps to a single activity label. Both assumptions break down in

multi-user scenarios due to class imbalance and permutation invariance, i.e., similar features can correspond to different slot orderings of the same underlying activity set (e.g., [walk, sit, $\emptyset$] and [sit, walk, $\emptyset$] are equivalent but produce different feature-label pairs).

Overall, existing UDA and SFUDA methods primarily target single-user classification under moderate domain gaps [48, 49]. Multi-user sensing suffers from: (i) class imbalance from unoccupied slots, (ii) signal entanglement invalidating stable feature mappings, and (iii) variable occupancy requiring structured set predictions rather than single-label outputs.

Chapter 3

MU-SHOT-Fi: Self-Supervised Multi-User Wi-Fi Sensing with Source-free Unsupervised Domain Adaptation

3.1 MU-SHOT-FI: Set-Based Multi-User HAR Formulation and Source Training

The proposed MU-SHOT-Fi (Fig. 3.1) addresses multi-user sensing through two key components: (1) permutation-invariant set-based prediction across M slots during source training via Hungarian matching (Section 3.1.3), and (2) target adaptation with frozen classifier while updating the backbone through rotation-based spatial self-supervision (Section 3.2). Specifically, this section first details the preprocessing of the CSI data, then presents the set-based HAR problem formulation, and finally describes the source-domain training mechanism for multi-user sensing.

3.1.1 Data Preprocessing

As established in Section 2.1.4, multi-user Wi-Fi sensing presents fundamental challenges due to signal entanglement, where the multipath model (Eq. 2.10) aggregates contributions from all active users simultaneously, making it difficult to attribute observed CSI variations to specific individuals [50, 51]. Unlike single-user settings, multi-user scenarios involve predicting an unordered activity set $\mathbf{y} = \{y_1, \ldots, y_{N_p}\}$, where $y_i \in \mathcal{A}$ represents an activity from K possible classes and N_p denotes the true number of people present. Critically, N_p is unknown and varies across samples, creating a variable-size prediction problem. This inherent permutation invariance—where one person walking while another sits has the same semantic meaning as one person sitting while another walks—motivates formulating the problem as set prediction rather than ordered sequence prediction.

From commodity Wi-Fi devices, we obtain raw CSI measurements as complex-valued tensors $\mathbf{x}_{\text{raw}} \in \mathbb{C}^{T \times N_r \times N_t \times N_{sc}}$ (Eq. 2.9), representing temporal samples, receive antennas, transmit antennas, and subcarriers. We reshape the spatial dimensions, yielding the input representation $\mathbf{x} \in \mathbb{R}^{1 \times F \times T}$, where $F = N_{sc} \cdot N_r \cdot N_t$ is the flattened spatial dimension and T is the temporal length. To handle variable occupancy, we pad the ground-truth activity set $\mathbf{y}$ to a fixed-size vector $\tilde{\mathbf{y}} \in (\mathcal{A} \cup \{\emptyset\})^M$ of length M using the "no person" token: $\tilde{\mathbf{y}} = [y_1, \ldots, y_{N_p}, \emptyset, \ldots, \emptyset]$, where $\tilde{\mathbf{y}}$ contains the N_p actual activities and $M - N_p$ instances of $\emptyset$ (the ordering is considered up to permutation, since training uses permutation-invariant bipartite matching). This allows the model to predict $\emptyset$ for unoccupied slots, reducing the hypothesis space from $(K)^{N_p}$ ordered sequences to $\binom{K+N_p-1}{N_p}$ unordered sets.

3.1.2 Source Model and Set-Based Multi-User HAR Formulation

We consider SFUDA for CSI-based multi-user sensing, where a model trained on a labeled source domain must adapt to an unlabeled target domain without access to the source dataset. Formally, let $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ denotes n_s labeled source samples drawn from distribution $P_s(\mathbf{X}, \mathbf{Y})$, and let $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ denotes n_t unlabeled target samples drawn from marginal distribution $P_t(\mathbf{X})$. The source and target domains exhibit distribution shift, i.e., $P_t(\mathbf{X}, \mathbf{Y}) \neq P_s(\mathbf{X}, \mathbf{Y})$. During adaptation, only the pre-trained source model and target data $\mathcal{D}_t$ are available; the source data $\mathcal{D}_s$ is inaccessible.

The source model consists of three components parameterized by $\theta_s = \{\eta_s, \psi_s, \phi_s\}$, i.e., a feature extractor $F_{\eta_s} : \mathbb{R}^{1\times F\times T} \to \mathbb{R}^{d_f}$, a bottleneck (linear) layer $B_{\psi_s} : \mathbb{R}^{d_f} \to \mathbb{R}^{d_b}$, and a classifier $C_{\phi_s} : \mathbb{R}^{d_b} \to \mathbb{R}^K$ (single-user) or $C_{\phi_s} : \mathbb{R}^{d_b} \to \mathbb{R}^{M\times(K+1)}$ (multi-user). The complete forward pass is $f_{\theta_s}(\mathbf{x}) = C_{\phi_s}(B_{\psi_s}(F_{\eta_s}(\mathbf{x})))$. For single-user tasks, the classifier outputs K activity class logits. For multi-user sensing, it outputs M slots with $K+1$ classes each (K activities plus "no person"), where M is the maximum occupancy.

We then formulate multi-user activity recognition as a set prediction problem, inspired by [62] and [63] in Wi-Fi sensing. The classifier $C_\phi : \mathbb{R}^{d_b} \to \mathbb{R}^{M\times(K+1)}$ outputs logits for M slots with $K+1$ classes (where d_b is the bottleneck dimension):

$$\hat{\mathbf{y}} = C_\phi(B_\psi(F_\eta(\mathbf{x}))) \in \mathbb{R}^{M\times(K+1)}. \tag{3.1}$$

Each slot $m \in \{1, \ldots, M\}$ is processed with softmax to produce a probability distribution over $K{+}1$ classes (where class $k = K$ represents "no person"):

$$p_{m,k} = \frac{\exp(\hat{y}[m,k])}{\sum_{k'=0}^{K} \exp(\hat{y}[m,k'])}, \quad k \in \{0, \ldots, K\}. \tag{3.2}$$

This formulation naturally represents any combination of up to M simultaneously active users, with variable occupancy captured by slots predicting $\emptyset$ (class K) versus actual activities (classes $0, \ldots, K{-}1$).

3.1.3 Hungarian Matching: Multi-User Source-Domain Training

During source training where ground-truth labels are available, we face a permutation invariance challenge: users can appear in any slot order, making direct index-based matching infeasible. For example, if two users perform "walk" and "sit," the ground-truth padded set $\tilde{\mathbf{y}} = \{\text{walk}, \text{sit}, \emptyset, \ldots, \emptyset\}$ is equivalent to $\tilde{\mathbf{y}} = \{\text{sit}, \text{walk}, \emptyset, \ldots, \emptyset\}$, yet they correspond to different slot-wise labels. We employ the Hungarian algorithm to obtain optimal permutation-invariant matching between predicted slots and ground-truth annotations [62, 63, 121].

For each sample, we construct a cost matrix $\mathbf{Q} \in \mathbb{R}^{M \times M}$ where each element represents the negative log-probability of assigning predicted slot i to ground-truth slot j:

$$Q_{ij} = -\log p_{i,\,\tilde{y}_j}, \tag{3.3}$$

where $\tilde{y}_j \in \mathcal{A} \cup \{\emptyset\}$ denotes the j-th element of the padded ground-truth vector $\tilde{\mathbf{y}}$, and $p_{i,\tilde{y}_j}$ is the predicted probability that slot i outputs activity class $\tilde{y}_j$ (from Eq. 3.2). The Hungarian algorithm efficiently finds the optimal permutation $\sigma^* \in \mathcal{S}_M$ (where

$\mathcal{S}_M$ is the set of permutations over M elements) that minimizes the total assignment cost:

$$\sigma^* = \arg\min_{\sigma \in \mathcal{S}_M} \sum_{i=1}^{M} Q_{i,\sigma(i)}. \tag{3.4}$$

The source training loss uses this optimal matching to compute cross-entropy loss function as shown below:

$$\mathcal{L}_{\text{matched-CE}} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{\text{CE}}(\hat{\mathbf{y}}[i], \tilde{\mathbf{y}}[\sigma^*(i)]), \tag{3.5}$$

where $\mathcal{L}_{\text{CE}}(\cdot, \cdot)$ is the cross-entropy loss between predicted logits and the ground-truth class index, and $\tilde{\mathbf{y}}[\sigma^*(i)] \in \{0, \ldots, K\}$ denotes the class index of the $\sigma^*(i)$-th element in the padded ground-truth vector. This matching-based supervision enables permutation-invariant learning during source training, i.e., the model is free to assign any activity to any slot, as the loss automatically finds the best alignment.

3.2 MU-SHOT-Fi: SFUDA in the Target Domain

MU-SHOT-Fi's SFUDA framework combines (1) *occupancy-weighted information maximization* (Section 3.2.1), which prevents collapse toward the dominant class by focusing diversity regularization on likely-occupied slots, and (2) *rotation-based spatial self-supervision* (Section 3.2.2), which exploits CSI's frequency-time structure for domain-invariant feature learning. The classifier remains frozen while the backbone adapts.

3.2.1 Occupancy-Weighted Information Maximization

In the source-free adaptation phase, target-domain labels are not available. As a result, the model cannot be fine-tuned using supervised cross-entropy. Instead, methods such as SHOT [119] rely on information maximization (IM), which uses prediction entropy as a self-training signal: it pushes the model toward confident decisions on target inputs while encouraging balanced use of classes through a marginal-distribution regularizer (GENT). MU-SHOT-Fi adopts this idea and designs a permutation-invariant entropy-based objective suitable for multi-user slot outputs. For a categorical distribution $\mathbf{p}$, Shannon entropy is defined as follows:

$$H(\mathbf{p}) = -\sum_{k} p_k \log(p_k + \epsilon), \tag{3.6}$$

where ϵ is a small constant used for numerical stability. High entropy indicates uncertainty (nearly uniform predictions) and low entropy indicates confident predictions.

3.2.1.1 Standard IM

The standard IM formulation is:

$$\mathcal{L}_{\text{IM}} = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{gent}}^{\text{std}}, \tag{3.7}$$

where $\mathcal{L}_{\text{ent}}$ drives confident predictions and $\mathcal{L}_{\text{gent}}^{\text{std}}$ maintains class diversity via conditional and marginal entropy terms.

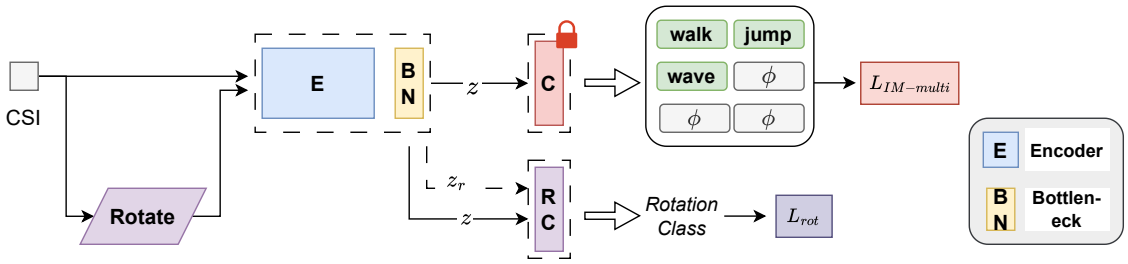


Figure 3.1: Architecture of the proposed MU-SHOT-Fi adaptation framework with spatial self-supervised losses for HAR.

3.2.1.2 Conditional entropy minimization (confidence)

Let $p_{i,m,k}$ be the softmax probability of class k at slot m for target sample i in a mini-batch of size N, where $m \in \{1, \ldots, M\}$ and $k \in \{0, \ldots, K\}$ (with $k = K$ denoting "no person"). Denote the slot-wise class distribution by $\mathbf{p}_{i,m} \in \mathbb{R}^{K+1}$ (Eq. 3.2). We encourage confident slot-wise predictions by minimizing the average conditional entropy:

$$\mathcal{L}_{\text{ent}} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} H(\mathbf{p}_{i,m}). \tag{3.8}$$

3.2.1.3 GENT: Marginal entropy maximization (diversity)

Entropy minimization alone can collapse to predicting the same class everywhere. SHOT-IM therefore uses GENT [120], which maximizes the entropy of the batch marginal class distribution. Standard SHOT-IM computes the marginal by uniformly averaging over all N samples and all M slots:

$$\bar{p}_k^{\text{std}} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} p_{i,m,k}, \quad k \in \{0, \ldots, K\}, \tag{3.9}$$

defining the diversity loss as

$$\mathcal{L}_{\text{gent}}^{\text{std}} = -H(\bar{\mathbf{p}}^{\text{std}}) = \sum_{k=0}^{K} \bar{p}_k^{\text{std}} \log(\bar{p}_k^{\text{std}} + \epsilon). \tag{3.10}$$

In multi-user sensing, Eq. (3.9) treats "no person" exactly like any activity and forces diversity over many truly-empty slots, which can amplify imbalance-driven collapse.

To understand this failure mode formally, recall that standard information maximization (IM) [119] aims to maximize the mutual information between input $\mathbf{x}$ and prediction $\hat{\mathbf{y}}$:

$$I(\mathbf{x}; \hat{\mathbf{y}}) = H(\hat{\mathbf{y}}) - H(\hat{\mathbf{y}} \mid \mathbf{x}), \tag{3.11}$$

where $H(\hat{\mathbf{y}})$ promotes diversity via the marginal distribution and $H(\hat{\mathbf{y}} \mid \mathbf{x})$ enforces confident predictions. In practice, $H(\hat{\mathbf{y}})$ is approximated by the batch marginal entropy $H(\bar{\mathbf{p}}^{\text{std}})$ (Eqs. (3.9)–(3.10)). However, in multi-user activity classification with $M = 6$ slots and variable occupancy, the "no person" class $k = K$ is structurally dominant, i.e., $\bar{p}_K^{\text{std}} \gg \bar{p}_k^{\text{std}}$ for $k \in \{0, \ldots, K-1\}$. As a result, maximizing $H(\bar{\mathbf{p}}^{\text{std}})$ over all $K+1$ classes encourages uniformity including class K, leading to trivial collapse where predictions concentrate on the dominant class [119, 120]. This failure mode is empirically confirmed in our experiments (Section 3.5), where standard IM degrades performance under domain shift.

The core issue is that standard IM maximizes $I(\mathbf{x}; \hat{\mathbf{y}})$ over the full label space without accounting for occupancy structure. In multi-user sensing, meaningful diversity should only be enforced over occupied slots, while empty slots should not influence the

marginal distribution. This motivates our proposed formulation, which can be interpreted as maximizing the *conditional* mutual information $I(\mathbf{x}; \hat{\mathbf{y}} \mid \text{occ} = 1)$, i.e., mutual information restricted to likely-occupied slots. Our occupancy-weighted GENT loss $\mathcal{L}_{\text{gent}}^{\text{occ}}$ implements this by weighting each slot's contribution with $p_{\text{occ}}[i, m] = 1 - p_{i,m,K}$: when a slot is confidently empty ($p_{i,m,K} \to 1$), its weight approaches zero and contributes nothing to diversity; when likely occupied ($p_{i,m,K} \to 0$), it contributes fully.

This resolves collapse, since the trivial solution of predicting $k = K$ everywhere yields zero weight for all slots and thus no diversity reward. While this shares intuition with cost-sensitive reweighting [122], our approach operates at the slot level within each prediction, dynamically determined by model confidence rather than dataset-level class frequencies.

3.2.1.4 Proposed Occupancy-weighted IM

Our multi-user information maximization objective is defined as follows:

$$\mathcal{L}_{\text{IM-multi}} = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{gent}}^{\text{occ}}. \tag{3.12}$$

By focusing diversity regularization on likely-occupied slots and excluding "no person" from the marginal, this objective mitigates imbalance-driven collapse: occupied slots are encouraged to spread across activities, while truly-empty slots can confidently predict "no person" without being penalized for lacking diversity. Our key modification is to keep the same *confidence* term $\mathcal{L}_{\text{ent}}$ (Eq. (3.8)), but redefine the *diversity* term so that it focuses on *likely-occupied* slots and excludes "no person" from the marginal.

First, we compute a slot occupancy probability as the probability of predicting any activity except "no person", i.e.,

$$p_{\text{occ}}[i, m] = \sum_{k=0}^{K-1} p_{i,m,k} = 1 - p_{i,m,K}. \tag{3.13}$$

Thus, slots with high $p_{\text{occ}}[i, m]$ are likely occupied. We then weight activity probability with the slot occupancy probability, i.e.,

$$\tilde{p}_{i,m,k} = p_{i,m,k} \cdot p_{\text{occ}}[i, m], \quad k = 0, \ldots, K-1. \tag{3.14}$$

This suppresses contributions from slots confidently predicting "no person," preventing the diversity term from pushing the model to artificially diversify among empty slots.

Next, we form the *occupancy-weighted marginal* over activity classes only (excluding "no person"), i.e.,

$$\bar{p}_k^{\text{occ}} = \frac{1}{Z} \cdot \frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} \tilde{p}_{i,m,k}, \quad k = 0, \ldots, K-1, \tag{3.15}$$

where $Z = \sum_{k=0}^{K-1} \bar{p}_k^{\text{occ}}$ ensures normalization. We then define the corresponding GENT loss function term as follows:

$$\mathcal{L}_{\text{gent}}^{\text{occ}} = -H(\bar{\mathbf{p}}^{\text{occ}}) = \sum_{k=0}^{K-1} \bar{p}_k^{\text{occ}} \log(\bar{p}_k^{\text{occ}} + \epsilon). \tag{3.16}$$

3.2.2 Rotation-based Spatial Self-Supervision

To align source and target representations without labels, we add self-supervised learning (SSL) losses on the shared CSI input and a shared bottleneck embedding, rather than on individual slot outputs. Concretely, let

$$\mathbf{b}(\mathbf{x}) = B_\psi(F_\eta(\mathbf{x})) \in \mathbb{R}^{d_b},$$

denotes the bottleneck feature for sample $\mathbf{x}$. SSL regularizes this shared embedding, benefiting all slots simultaneously.

Specifically, we use a rotation prediction task (0° vs. 180°) on the frequency–time CSI grid [64]. Let $\mathrm{rot}(\mathbf{x}, r)$ be $\mathbf{x}$ rotated by angle $r \in \{0^\circ, 180^\circ\}$. The rotation head R_γ takes the concatenation of bottleneck features from the original and rotated inputs and predicts r, i.e.,

$$R_\gamma([\mathbf{b}(\mathbf{x}); \mathbf{b}(\mathrm{rot}(\mathbf{x}, r))]) \in \mathbb{R}^2.$$

Following SHOT++ [120], we (i) pre-train R_γ on target data while freezing the source feature extractor, then (ii) keep the task during adaptation but apply stop-gradient on the original branch to avoid interfering with activity classification:

$$\mathcal{L}_{\mathrm{rot}} = \mathbb{E}_{\mathbf{x},r}\Big[\mathcal{L}_{\mathrm{CE}}\big(R_\gamma([\mathrm{sg}(\mathbf{b}(\mathbf{x})); \mathbf{b}(\mathrm{rot}(\mathbf{x}, r))]), r\big)\Big]. \tag{3.17}$$

Here $\mathrm{sg}(\cdot)$ denotes stop-gradient [123], and $[\cdot;\cdot]$ denotes concatenation. Stop-gradient on the original branch prevents the rotation loss from pulling the shared embedding away from what the main slot-prediction objective needs, while the rotated branch still provides spatial regularization.

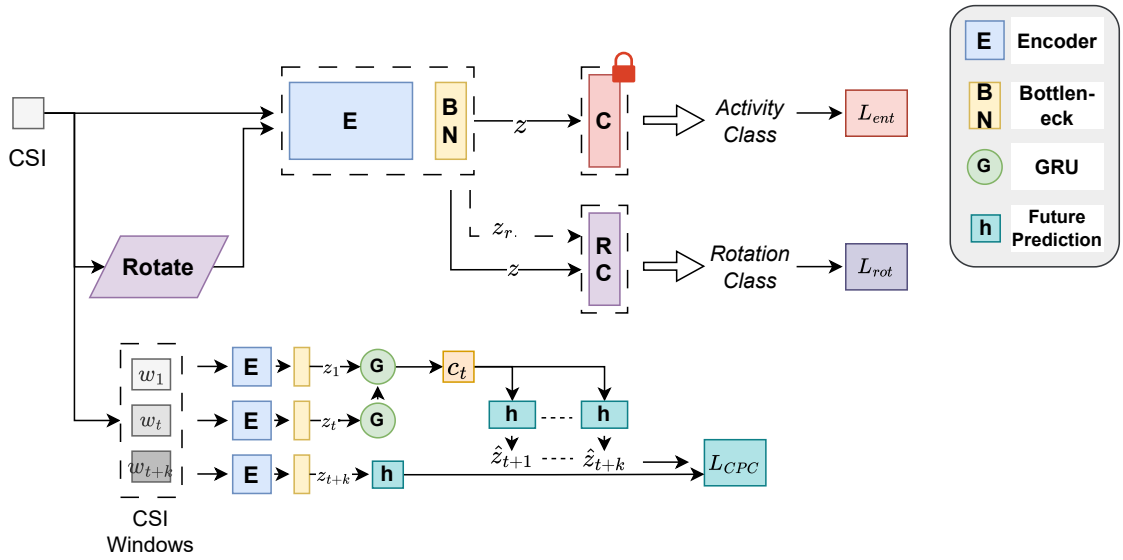


Figure 3.2: Architecture of the proposed single-user SHOT-Fi adaptation framework with spatiotemporal SSL for HAR.

3.2.3 Proposed Loss Function

The proposed multi-user adaptation objective combines occupancy-weighted information maximization and rotation-based self-supervised loss function and can be formulated as:

$$\mathcal{L}_{\text{MU-SHOT-Fi}} = \lambda_{\text{ent}}\mathcal{L}_{\text{IM-multi}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}}. \tag{3.18}$$

where λ_{ent} and λ_{rot} represent the weight of contribution of each loss term. Training proceeds in two stages: (i) rotation SSL pre-training on unlabeled target data with frozen source features, followed by (ii) joint adaptation with $\mathcal{L}_{\text{IM-multi}}$ and $\mathcal{L}_{\text{rot}}$. Algorithm 1 details the sequence of training procedure.

Algorithm 1: MU-SHOT-Fi: Multi-User Adaptation

Require: Pre-trained source model $f_{\theta_s} = C_{\phi_s} \circ B_{\psi_s} \circ F_{\eta_s}$, unlabeled target data $\mathcal{D}_t$, number of slots M, number of activity classes K

Ensure: Adapted model f_{θ_t}

Stage 1: Rotation SSL Pre-training on Target Domain
Pre-train rotation classifier R_γ with frozen F_{η_s} and B_{ψ_s} (Eq. 3.17)
Stage 2: Joint Adaptation
Initialize: $F_{\eta_t} \leftarrow F_{\eta_s}$, $B_{\psi_t} \leftarrow B_{\psi_s}$, $C_{\phi_t} \leftarrow C_{\phi_s}$
Freeze classifier: C_{ϕ_t}.requires_grad = False
for each epoch **do**
  for each batch $\{\mathbf{x}_i^t\}_{i=1}^N$ in $\mathcal{D}_t$ **do**
    Forward: $\mathbf{y}_{\text{pred}} = C_{\phi_t}(B_{\psi_t}(F_{\eta_t}(\mathbf{x}_i^t))) \in \mathbb{R}^{N \times M \times (K+1)}$
    Compute $p_{i,m,k} = \text{softmax}(\mathbf{y}_{\text{pred}}[i, m, :])$ for all b, m (Eq. 3.2)
    // Occupancy-Weighted Information Maximization
    Compute $\mathcal{L}_{\text{ent}}$ via Eq. 3.8
    Compute $p_{\text{occ}}[i, m]$ via Eq. 3.13
    Compute $\tilde{p}_{i,m,k}$ via Eq. 3.14
    Compute marginal $\bar{p}_k$ via Eq. 3.15, then normalize
    Compute $\mathcal{L}_{\text{gent}}^{\text{occ}}$ via Eq. 3.16
    $\mathcal{L}_{\text{IM-multi}} = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{gent}}^{\text{occ}}$
    // Self-Supervised Loss
    Compute $\mathcal{L}_{\text{rot}}$ with stop-gradient (Eq. 3.17)
    $\mathcal{L}_{\text{MU-SHOT-Fi}} = \lambda_{\text{ent}}\mathcal{L}_{\text{IM-multi}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}}$
    Update $\{\eta_t, \psi_t, \gamma\}$ via gradient descent
  end for
end for
return Adapted model f_{θ_t}

3.3 SU-SHOT-Fi: Single-User HAR as a Special Case

The MU-SHOT-Fi framework simplifies naturally to single-user scenarios where each CSI sample corresponds to exactly one activity from K mutually exclusive classes (see Algorithm 2 and Fig. 3.2). In this setting, the proposed adaptation strategy for SU-SHOT-Fi is explained below.

3.3.1 Standard IM with no Hungarian Matching

First, occupancy is fixed at one user per sample, eliminating variable occupancy. Thus, we use the standard SHOT-IM objective [119] denoted by $\mathcal{L}_{\text{IM}}$, which combines (i) *conditional entropy minimization* to encourage confident predictions, and (ii) *marginal entropy maximization* (GENT) to avoid degenerate collapse by promoting diverse classes across the target batch. Second, with only one prediction per sample, slot permutation invariance disappears—there is no need for Hungarian matching as predictions map directly to ground truth during source training.

3.3.2 Clustering-based Pseudo-labeling

Unlike multi-user scenarios, single-user datasets exhibit relatively balanced class distributions across activities [65]. This balanced setting enables SHOT's clustering-based pseudo-labeling [119], since the model won't converge to the dominant class which is a limitation of multi-user case. Pseudo-labeling generates supervision through clustering. At each epoch, we extract bottleneck features $\mathbf{z}_j = B_\psi(F_\eta(\mathbf{x}_j^t)) \in \mathbb{R}^{d_b}$ and predictions $\mathbf{p}_j = \sigma(C_\phi(\mathbf{z}_j)) \in \mathbb{R}^K$ for all target samples $j \in \{1, \ldots, n_t\}$. Features are normalized with a bias term $\tilde{\mathbf{z}}_j = [\mathbf{z}_j; 1]/\|\mathbf{z}_j\|_2 \in \mathbb{R}^{d_b+1}$, and class centroids are initialized as the prediction-weighted average of normalized features, i.e.,

$$\mathbf{c}_k = \frac{\sum_{j=1}^{n_t} p_{jk} \tilde{\mathbf{z}}_j}{\sum_{j=1}^{n_t} p_{jk}}, \quad k \in \{1, \ldots, K\}. \tag{3.19}$$

Each sample is then assigned to its nearest centroid using cosine distance:

$$\hat{y}_j = \arg \min_{k \in \{1,\ldots,K\}} d_{\text{cosine}}(\tilde{\mathbf{z}}_j, \mathbf{c}_k), \tag{3.20}$$

where $d_{\text{cosine}}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$. Centroids are refined once using the following assignments, i.e.,

$$\mathbf{c}_k \leftarrow \frac{\sum_{j:\hat{y}_j=k} \tilde{\mathbf{z}}_j}{|\{j : \hat{y}_j = k\}|}, \tag{3.21}$$

and pseudo-labels are then reassigned. The pseudo-labeling loss supervises the model, i.e.,

$$\mathcal{L}_{\text{PL}} = \frac{1}{n_t} \sum_{j=1}^{n_t} \mathcal{L}_{\text{CE}}(C_\phi(B_\psi(F_\eta(\mathbf{x}_j^t))), \hat{y}_j). \tag{3.22}$$

The classifier outputs $C_\phi : \mathbb{R}^{d_b} \to \mathbb{R}^K$ logits over K activities without the "no person" class.

3.3.3 Temporal SSL via CPC

We also consider Contrastive Predictive Coding (CPC) [124] to exploit temporal structure in CSI sequences. While rotation-based spatial SSL targets invariances in the frequency–time grid, temporal dynamics (e.g., periodic stride patterns during walking) provide domain-invariant cues even when absolute CSI values shift across environments and hardware. CPC splits $\mathbf{x} \in \mathbb{R}^{1\times F\times T}$ into $W = \lfloor T/w \rfloor$ windows of length w, encodes each window with encoder g_ξ, summarizes past context with GRU h_ω, and predicts future embeddings using InfoNCE loss:

$$\mathcal{L}_{\text{CPC}} = \frac{1}{K_p} \sum_{k=1}^{K_p} \left[-\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(s_{i,i}^{(k)})}{\sum_{j=1}^{N} \exp(s_{i,j}^{(k)})} \right], \tag{3.23}$$

where K_p is the prediction horizon and $s_{i,j}^{(k)} = \hat{\mathbf{z}}_{t+k}^{(i)\top} \mathbf{z}_{t+k}^{(j)} / \tau$ is the temperature-scaled cosine similarity between predicted future embedding $\hat{\mathbf{z}}_{t+k}^{(i)}$ and true future embedding $\mathbf{z}_{t+k}^{(j)}$.

3.3.4 SU-SHOT-Fi Loss Function:

The proposed SU-SHOT-Fi objective can then be given as:

$$\mathcal{L}_{\text{SU-SHOT-Fi}} = \lambda_{\text{cls}}\mathcal{L}_{\text{PL}} + \lambda_{\text{ent}}\mathcal{L}_{\text{IM}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{cpc}}\mathcal{L}_{\text{CPC}}, \tag{3.24}$$

where $\mathcal{L}_{\text{PL}}$ is the pseudo-labeling loss, $\mathcal{L}_{\text{IM}}$ is standard information maximization, and λ_{cls}, λ_{cpc} are the corresponding loss weights. Training follows the same two-stage procedure: SSL pre-training of rotation classifier and CPC model on unlabeled target data, followed by joint adaptation combining SHOT-IM with CSI-specific self-supervised losses. Algorithm 2 summarizes the complete procedure, and Fig. 3.2 illustrates the architecture.

3.4 Experimental Set-Up, Baselines, and Evaluation Metrics

3.4.1 Considered Datasets

We validate MU-SHOT-Fi on two datasets: WiMANS [51] for multi-user activity recognition under environmental and hardware variations, and Widar 3.0 [65] for single-user gesture recognition under environmental and positional shifts.

3.4.1.1 WiMANS [51]

is a comprehensive multi-user dataset captured in classroom and meeting room environments at both 2.4 GHz and 5 GHz frequencies using a 3×3 MIMO configuration. The dataset contains samples with 0 to 5 simultaneous users performing 9 distinct activities: nothing, walk, rotation, jump, wave, lie down, pick up, sit down, and stand

Algorithm 2: SU-SHOT-Fi: Single-User Adaptation

Require: Pre-trained source model $f_{\theta_s} = C_{\phi_s} \circ B_{\psi_s} \circ F_{\eta_s}$, unlabeled target data $\mathcal{D}_t$, number of activity classes K

Ensure: Adapted model f_{θ_t}

Stage 1: SSL Pre-training on Target Domain

Pre-train rotation classifier R_γ with frozen F_{η_s}, B_{ψ_s} (Eq. 3.17)

Pre-train CPC model $(g_\xi, h_\omega, \{W_k\})$ on $\mathcal{D}_t$ (Eq. 3.23)

Stage 2: Joint Adaptation with Pseudo-Labeling

Initialize: $F_{\eta_t} \leftarrow F_{\eta_s}$, $B_{\psi_t} \leftarrow B_{\psi_s}$, $C_{\phi_t} \leftarrow C_{\phi_s}$; Freeze C_{ϕ_t}

for each epoch **do**

// Generate Pseudo-Labels via K-Nearest Centroids

Extract $\mathbf{z}_j = B_{\psi_t}(F_{\eta_t}(\mathbf{x}_j^t))$, $\mathbf{p}_j = \sigma(C_{\phi_t}(\mathbf{z}_j))$ for all $j \in \{1, \ldots, n_t\}$

Normalize: $\tilde{\mathbf{z}}_j = [\mathbf{z}_j; 1] / \|[\mathbf{z}_j; 1]\|_2$

Initialize centroids: $\mathbf{c}_k = \sum_{j=1}^{n_t} [\mathbf{p}_j]_k \tilde{\mathbf{z}}_j / \sum_{j=1}^{n_t} [\mathbf{p}_j]_k$, $k \in \{1, \ldots, K\}$

Assign pseudo-labels: $\hat{y}_j = \arg\min_k d_{\text{cosine}}(\tilde{\mathbf{z}}_j, \mathbf{c}_k)$; refine centroids once

for each batch $\{\mathbf{x}_j^t\}$ **do**

Compute $\mathcal{L}_{\text{PL}}$ (Eq. 3.22), $\mathcal{L}_{\text{IM}}$ (Eq. 3.7), $\mathcal{L}_{\text{rot}}$ (Eq. 3.17), $\mathcal{L}_{\text{CPC}}$ (Eq. 3.23)

$\mathcal{L}_{\text{SU-SHOT-Fi}} = \lambda_{\text{cls}}\mathcal{L}_{\text{PL}} + \lambda_{\text{ent}}\mathcal{L}_{\text{IM}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{cpc}}\mathcal{L}_{\text{CPC}}$

Update $\{\eta_t, \psi_t, \gamma, \xi, \omega, \{W_k\}\}$ via gradient descent

end for

end for

return Adapted model f_{θ_t}

up. Each CSI measurement is collected at 1000 Hz sampling rate across 30 subcarriers over a 3-second window, yielding complex-valued tensors $\mathbf{x}_{\text{raw}} \in \mathbb{C}^{3000\times30\times3\times3}$ representing temporal samples, subcarriers, transmit antennas, and receive antennas, respectively. Following standard pre-processing [51], we extract amplitude and reshape to $\mathbf{x} \in \mathbb{R}^{1\times3000\times270}$ where the spatial dimensions are flattened (30 subcarriers $\times$ 3 transmit $\times$ 3 receive = 270 features). To test MU-SHOT-Fi, we take $M = 6$ and $K + 1 = 10$ classes (9 activities plus "no person").

The source domain consists of classroom samples at 2.4 GHz. We evaluate three domain shift scenarios:

(1) **Cross-Room** (classroom to meeting room at 2.4 GHz) evaluates performance

under environmental changes, such as room geometry and furniture layout changes;

(2) **Cross-Frequency** (classroom at 2.4 GHz to 5 GHz) evaluates performance under frequency changes.

(3) **Combined Shift** (classroom at 2.4 GHz to meeting room at 5 GHz) evaluates performance under simultaneous environmental and frequency changes.

3.4.1.2 Widar 3.0 [65]

contains gesture samples across 6 classes (push and pull, sweep, clap, slide, draw-O(H), draw-zigzag(H)) from 9 users across 5 physical locations and 5 body orientations, collected at 5.825 GHz with one transmitter and six receivers (3 antennas each). Raw CSI has dimensions $(N_t \times N_r \times N_s \times T) = (1 \times 18 \times 30 \times \text{variable})$ for transmit antenna, receive antennas (6 receivers $\times$ 3 antennas), subcarriers, and temporal samples. Following standard preprocessing [52], we standardize temporal length to 1200 samples and extract CSI phase ratios. Raw CSI $\mathbf{x}_{\text{raw}} \in \mathbb{C}^{1\times18\times30\times T}$ represents 1 transmit antenna, 18 receive antennas (6 receivers $\times$ 3 antennas per receiver), 30 subcarriers, and temporal samples. For each receiver $r \in \{1, \ldots, 6\}$, we compute the CSI ratio between its first two antennas $H_r^{\text{ratio}}(k,t) = H_{r,1}(k,t)/H_{r,2}(k,t)$ for subcarrier k and time t, which eliminates time-varying phase offsets [52]. We extract the phase $\phi_r(k,t) = \angle H_r^{\text{ratio}}(k,t)$ and stack across receivers and subcarriers, yielding $\mathbf{x} \in \mathbb{R}^{1\times180\times1200}$ where $180 = 6$ receivers $\times$ 30 subcarriers. We evaluate three domain shift scenarios:

(1) **Cross-Room** (Room 1 to Room 2) evaluates environmental shift from room geometry and materials;

(2) **Cross-Torso** (orientations 2–5 to 1) evaluates body rotation within the same

Table 3.1: WiMANS hyperparameters for MU-SHOT-Fi

Component	**Parameter**	**Value**
General	Adaptation epochs	50
	Batch size	64
	Optimizer	Adam
	Adaptation learning rate	1×10^{-4}
Multi-User Backbone	Slots (M)	6
	Classes per slot	10
	Label smoothing (ϵ)	0.2
	Entropy minimization (λ_{ent})	1.0
	Diversity maximization	Enabled
	Pseudo-labeling weight (λ_{cls})	0.0
Rotation SSL	Pre-training epochs	70
	Loss weight (λ_{rot})	0.5

room;

(3) **Cross-Face** (locations 2–5 to 1) evaluates head orientation changes affecting signal shadowing.

WiMANS serves as the primary benchmark for evaluating MU-SHOT-Fi, as it provides multi-user CSI samples with simultaneous activity annotations across diverse domain shift conditions (cross-room, cross-frequency, and combined). To the best of our knowledge, it is the only open-access dataset offering this combination [40, 125], making it the natural testbed for all multi-user claims, including occupancy-weighted information maximization, permutation-invariant set prediction, and dominant-class collapse prevention. Widar 3.0 complements this evaluation by validating the core adaptation components shared between MU-SHOT-Fi and SU-SHOT-Fi, namely rotation-based self-supervision and information maximization, in a controlled single-user setting where confounding factors such as signal entanglement and variable occupancy are absent.

Table 3.2: Widar 3.0 hyperparameters for SHOT-Fi

Component	**Parameter**	**Value**
General	Adaptation epochs	70
	Batch size	32
	Optimizer	Adam
	Adaptation learning rate	1×10^{-4}
SHOT-IM Backbone	Pseudo-labeling weight (λ_{cls})	0.1
	Entropy minimization (λ_{ent})	1.0
	Diversity maximization	Enabled
Rotation SSL	Pre-training epochs	70
	Loss weight (λ_{rot})	0.3
CPC	Pre-training epochs	70
	Pre-training learning rate	1×10^{-3}
	Loss weight (λ_{cpc})	0.3
	Window size (w)	10 timesteps
	Prediction steps (K_p)	9
	Encoder embedding dim (d_e)	256
	Context (GRU) hidden dim (d_c)	512
	Projection head dim (d_p)	256
	InfoNCE temperature (τ)	0.07
	Mask probability	0.5
	Mask ratio (per window)	0.15

3.4.2 Evaluation Metrics

We report slot-level results after Hungarian alignment (slot-wise accuracy, Activity Macro-F1), sample-level performance (exact match accuracy), and occupancy-level performance (occupancy MAE, occupancy exact match).

3.4.2.1 Multi-User HAR Evaluation Metrics.

Each test sample contains $M = 6$ user slots with M categorical predictions and M ground-truth slot labels, where a dedicated *no-person* class denotes an empty slot. Since user ordering is arbitrary, we align predicted slots with ground-truth slots using Hungarian matching [121] given by:

$$\pi^{\star} = \arg\min_{\pi \in \mathcal{S}_M} \sum_{m=1}^{M} -\log p\big(\hat{y}_{i,m} = y_{i,\pi(m)}\big)\,, \tag{3.25}$$

where $\mathcal{S}_M$ is the set of permutations over M slots and $p(\cdot)$ is the model's softmax probability.

Slot-wise Accuracy We then compute slot-wise accuracy as follows:

$$\text{SlotAcc} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbb{I}\big(\hat{y}_{i,m} = y_{i,\pi^\star(m)}\big)\,. \tag{3.26}$$

Activity Macro-F1 evaluates recognition quality over the K activity classes only (excluding $\emptyset$). Let $\mathcal{I} = \{(i,m) : y_{i,\pi^\star(m)} \neq \emptyset\}$ denotes occupied ground-truth slots:

$$\text{ActivityF1} = \frac{1}{K} \sum_{k=1}^{K} \text{F1}\Big(\{\hat{y}_{i,m}\}_{(i,m)\in\mathcal{I}}, \{y_{i,\pi^\star(m)}\}_{(i,m)\in\mathcal{I}}; k\Big).$$

Exact Match Accuracy Exact Match counts a prediction as correct only if all M slots match after Hungarian alignment, i.e.,

$$\text{ExactMatch} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\bigwedge_{m=1}^{M} \hat{y}_{i,m} = y_{i,\pi^\star(m)} \right). \tag{3.27}$$

Occupancy Metrics Ground-truth and predicted occupancy for sample i can be defined as $o_i = \sum_{m=1}^{M} \mathbb{I}(y_{i,m} \neq \emptyset)$ and $\hat{o}_i = \sum_{m=1}^{M} \mathbb{I}(\hat{y}_{i,m} \neq \emptyset)$, respectively. The occupancy metrics are then given as follows:

$$\text{OccMAE} = \frac{1}{N} \sum_{i=1}^{N} |o_i - \hat{o}_i|\,, \quad \text{OccExact} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(o_i = \hat{o}_i).$$

Remark: ExactMatch is the strictest end-to-end metric, requiring all M aligned slots to match. ActivityF1 focuses on activity recognition over occupied slots, reducing influence of the no-person class $\emptyset$, while SlotAcc (computed over all matched slot labels including $\emptyset$) can become inflated when many slots are empty.

Remark on metric interdependence: Activity F1 is computed exclusively over ground-truth occupied slots and does not penalize predictions on empty slots. Consequently, a model that over-predicts activities across all M slots can achieve relatively high Activity F1 while incurring large occupancy errors, as the metric rewards any correct activity match regardless of occupancy accuracy.

Conversely, improvements in Slot-wise Accuracy and Occupancy MAE that arise from more precise occupancy estimation may coincide with lower Activity F1, since fewer but more selective activity predictions reduce the chance of incidental matches in occupied slots.

Therefore, no single metric is sufficient for evaluating multi-user performance: Activity F1 must be interpreted jointly with occupancy metrics to distinguish genuine activity recognition from over-prediction, and Slot-wise Accuracy should be considered alongside Activity F1 to verify that gains reflect activity discrimination rather than empty-slot exploitation.

3.4.2.2 Single-User HAR Evaluation Metrics.

For single-user gesture recognition, the evaluation simplifies to standard multi-class classification. Unlike multi-user scenarios where variable occupancy and class imbalance create fundamental evaluation challenges, single-user metrics follow established classification conventions. Each sample corresponds to exactly one gesture from $K = 6$ classes, and we report: **Classification Accuracy** The fraction of correctly classified samples, i.e., Accuracy $= \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i)$, where $\hat{y}_i$ is the predicted class and y_i is the ground-truth class.

Per-Class F1-Score To evaluate performance across gesture types beyond overall accuracy, we compute F1-scores for each class k using standard binary classification metrics, i.e., $\mathrm{F1}_k = \frac{2 \cdot \mathrm{Precision}_k \cdot \mathrm{Recall}_k}{\mathrm{Precision}_k + \mathrm{Recall}_k}$. We report both per-class F1-scores and their macro-average. Unlike accuracy, macro-F1 weights each gesture equally and highlights gesture-specific precision/recall failures, revealing which gestures are most affected by domain shift and whether adaptation improves performance uniformly across classes.

3.4.3 Considered Baselines

We compare MU-SHOT-Fi and SU-SHOT-Fi against:

(1) **Source-only** model trained on labeled source data and tested on target data with no adaptation, to quantify the domain-shift gap;

(2) **SHOT-IM [119]**, a standard source-free UDA baseline based on information maximization, with pseudo-labeling enabled only in the single-user case;

(3) **SHOT++ [120]**, an improved SHOT-IM variant with additional SSL target-side regularization, with pseudo-labeling enabled only in the single-user case.

We select these baselines because, to the best of our knowledge, there are no prior SFUDA methods tailored to multi-user Wi-Fi sensing; thus, *Source-only* quantifies the domain-shift gap and SHOT-IM/SHOT++ [119, 120] serves as the closest and strongest existing SFUDA references for assessing how standard single-user adaptation objectives behave when extended to permutation-invariant, variable-occupancy outputs.

For a fair comparison in the multi-user scenario, all methods use the same set-prediction formulation and Hungarian matching during source training to handle

permutation invariance. During target adaptation, pseudo-labeling is disabled for SHOT-IM/SHOT++ in the multi-user setting since slot-level pseudo-labels are ill-defined without correspondence and become unstable under variable occupancy and dominant $\emptyset$ slots. For SHOT++ we also disable the MixMatch stage since it is a generic semi-supervised add-on that could be applied to all methods [126].

3.4.4 Model Hyperparameter Settings

3.4.4.1 MU-SHOT-Fi.

The feature extractor consists of three 2D convolutional blocks with kernels (27,27), (15,15), (7,7); strides (7,7), (3,3), (1,1); and channels 32, 64, 128. Each block uses batch normalization, LeakyReLU activation, and dropout (0.2). Global average pooling produces 128-dimensional features compressed through a bottleneck to 128 dimensions. The classifier outputs 6×10 logits for $M = 6$ slots with $K + 1 = 10$ classes. Source model trains for 50 epochs with Hungarian matching, label smoothing ($\epsilon = 0.2$), and learning rate 1×10^{-3}. Adaptation runs for 50 epochs with batch size 64, learning rate 1×10^{-4}, and loss weights $\lambda_{\text{ent}} = 1.0$ and $\lambda_{\text{rot}} = 0.5$. Complete hyperparameters are given in Table 3.1.

3.4.4.2 SU-SHOT-Fi.

We use ResNet-18 as feature extractor, modified for input shape $(B, 1, 180, 1200)$. The bottleneck reduces features to dimension 512 and the classifier outputs logits for 6 classes. Source models train for 30 epochs with label smoothing ($\epsilon = 0.1$), SGD (learning rate 0.1, momentum 0.9, weight decay 5×10^{-4}), and batch size 32. CPC uses window size $w = 10$ (120 windows per 1200-timestep sample),

Table 3.3: WiMANS: SFUDA results under three domain shifts. Values are mean±std across runs. Higher is better (↑) except where noted (↓). Best within each scenario is **bolded** and second-best is underlined.

Scenario	Method	Exact Match (%) ↑	Activity F1 (%)† ↑	Slot-wise Acc (%) ↑	Occupancy MAE ↓	Occupancy Exact Match (%) ↑
Cross-Room	Source Only	2.12±0.43	15.89±0.47	55.09±2.47	**1.52±0.24**	20.45±4.77
	SHOT-IM	2.82±2.94	**24.97±1.01**	37.12±7.09	2.07±0.47	17.99±7.68
	SHOT++	7.05±2.17	16.88±1.41	55.32±1.96	1.63±0.26	23.28±4.86
	MU-SHOT-Fi (Ours)	**7.76±1.51**	17.08±1.15	55.29±1.98	1.60±0.25	**24.61±3.72**
	MU-SHOT-Fi + CPC	6.34±2.41	16.16±0.54	**55.43±1.37**	1.63±0.25	22.75±4.98
Cross-Frequency	Source Only	0.00±0.00	25.69±0.97	21.16±4.71	2.92±0.33	7.40±1.14
	SHOT-IM	0.35±0.49	**26.56±2.86**	32.39±2.91	2.61±0.32	9.34±2.17
	SHOT++	0.71±0.25	19.61±1.59	**47.08±1.57**	1.89±0.16	**15.16±3.52**
	MU-SHOT-Fi (Ours)	0.71±0.25	18.82±1.50	46.97±1.35	**1.89±0.15**	14.63±3.52
	MU-SHOT-Fi + CPC	**1.41±0.24**	19.55±2.13	**47.33±1.91**	1.91±0.14	13.40±2.87
Combined Shift	Source Only	0.00±0.00	**24.99±2.53**	19.61±4.08	3.00±0.35	5.99±4.75
	SHOT-IM	0.00±0.00	23.62±2.75	28.63±4.21	2.80±0.38	9.17±4.11
	SHOT++	**0.18±0.25**	19.23±1.94	41.06±2.76	2.16±0.16	10.58±2.62
	MU-SHOT-Fi (Ours)	**0.18±0.25**	18.47±2.91	**41.97±2.39**	**2.12±0.12**	**10.93±1.79**
	MU-SHOT-Fi + CPC	**0.18±0.25**	18.45±2.41	41.58±2.68	2.15±0.14	5.99±4.75
Average	Source Only	0.71±0.14	22.19±1.32	31.95±3.75	2.48±0.31	11.28±3.55
	SHOT-IM	1.06±1.14	**25.05±2.21**	32.71±4.74	2.49±0.39	12.17±4.65
	SHOT++	2.65±0.89	18.57±1.65	47.82±2.10	1.89±0.19	16.34±3.67
	MU-SHOT-Fi (Ours)	**2.88±0.67**	18.12±1.85	48.08±1.91	**1.87±0.17**	**16.72±3.01**
	MU-SHOT-Fi + CPC	2.64±0.97	18.05±1.69	**48.11±1.99**	1.90±0.18	14.04±4.20
Random Predictor		≈0.00	≈11.11	≈10.00	≈5.00	≈1.54

Note: Average row shows mean performance across the three domain shift scenarios.

predicts $K = 9$ future windows, with encoder dimension 256, GRU hidden dimension 512, and InfoNCE temperature $\tau = 0.07$. Adaptation runs for 70 epochs with Adam optimizer, learning rate 1×10^{-4}, batch size 32, and loss weights $\lambda_{\text{cls}} = 0.1$, $\lambda_{\text{ent}} = 1.0$, $\lambda_{\text{rot}} = 0.3$, $\lambda_{\text{cpc}} = 0.3$. Results average over 3 runs with different random seeds. Complete hyperparameters are given in Table 3.2.

3.4.5 Computational Overhead

Table 3.4 summarizes the parameter counts, GFLOPs, and per-batch wall-clock times for all methods, measured on an NVIDIA H100 80 GB GPU with batch size 64 and input shape ($64 \times 3000 \times 270$) using 10 warm-up and 50 timed iterations. Since SFUDA assumes access only to a pretrained source model and unlabeled target data, we report adaptation and inference costs only.

SHOT-IM relies solely on entropy minimization during adaptation, contributing 107.71 GFLOPs per batch. SHOT++ and MU-SHOT-Fi additionally perform rotation SSL pre-training (215.43 GFLOPs) before the joint adaptation loop, resulting in 323.14 GFLOPs per batch during adaptation. The SSL pre-training stage (0.009 s per batch) is run only once prior to adaptation and does not contribute to per-epoch adaptation time. Despite this additional pre-training cost, the wall-clock time of MU-SHOT-Fi during adaptation (0.234 s per batch) remains comparable to SHOT++ (0.232 s per batch), as both share an identical forward pass structure.

After deployment, all methods use only the main backbone components (F_η, B_ψ, C_ϕ) for inference, incurring 107.71 GFLOPs per batch and a wall-clock time of 0.003 s per batch (0.058 ms per sample). This decoupling enables a practical deployment strategy: adaptation is performed once on a server with sufficient resources, after which only the lightweight adapted model is deployed at the edge. The total parameter count of MU-SHOT-Fi is 910,976 during adaptation (886,210 for F_η, 16,512 for B_ψ, 7,740 for C_ϕ, and 514 for the rotation head R_γ), reducing to 910,462 at inference once R_γ is discarded.

Table 3.4: Computational overhead per batch (batch size = 64, input $64 \times 3000 \times 270$) on an NVIDIA H100 80 GB GPU (10 warm-up / 50 timed iterations). R_γ denotes the rotation head used only during adaptation.

Method	Stage	Parameters	GFLOPs	Time/Batch (s)
SHOT-IM	Adaptation (entropy min.)	910,462	107.71	0.1035
SHOT++	SSL pre-training of R_γ (once)	903,236	215.43	0.0087
	Joint adaptation	910,976	323.14	0.2313
MU-SHOT-Fi	SSL pre-training of R_γ (once)	903,236	215.43	0.0087
	Joint adaptation	910,976	323.14	0.2337
All methods	Inference ($F_\eta + B_\psi + C_\phi$)	910,462	107.71	0.0037

3.5 Numerical Results and Discussions

3.5.1 MU-SHOT-Fi Results and Analysis

Table 3.3 presents comprehensive performance metrics for MU-SHOT-Fi across three domain shift scenarios on the WiMANS dataset. Our experimental results reveal a consistent pattern: adaptation gains scale inversely with source-only baseline performance. This relationship reflects a fundamental principle in domain adaptation: when the initial domain gap is large (weak source-only performance), there exists greater headroom for improvement through adaptation. Specifically, scenarios exhibiting strong domain mismatch (Cross-Frequency and Combined shifts) demonstrate near-complete source model failure (0.00% Exact Match, approximately 20% Slot-wise Accuracy), yet achieve substantial recovery through adaptation. In contrast, the Cross-Room scenario begins from a relatively stronger baseline (55.09% Slot-wise Accuracy), with limited improvements due to the reduced domain mismatch.

All results are reported as mean ± std across 3 independent runs with different

random seeds. The non-overlapping standard deviation intervals between MU-SHOT-Fi and source-only baselines confirm the reliability of the reported improvements. For example, under Combined Shift, MU-SHOT-Fi achieves $41.97\% \pm 2.39$ Slot-wise Accuracy compared to $19.61\% \pm 4.08$ for source-only.

We note that Exact Match is an inherently strict metric due to the combinatorial output space: with $M = 6$ slots each over $K + 1 = 10$ classes, the joint label space contains $10^6 = 1{,}000{,}000$ configurations. A uniformly random predictor achieves Exact Match of approximately $(1/10)^6 \approx 0.0001\%$. In comparison, MU-SHOT-Fi attains 2.88% average Exact Match, representing an improvement of approximately 28,800× over random chance. To the best of our knowledge, prior works on WiMANS and related multi-user sensing benchmarks do not report Exact Match even under full supervision [50, 51, 127], reflecting a broader consensus that slot-level accuracy and occupancy estimation provide more stable measures of multi-user performance. We include Exact Match as an additional stringent evaluation perspective and report a random predictor baseline in Table 3.3 for reference.

3.5.1.1 Cross-Room Domain Shift.

Source-only models attain 55.09% slot-wise Accuracy but only 2.12% Exact Match, indicating that while individual slots are often correct, full multi-user configurations are rarely recovered. MU-SHOT-Fi increases Exact Match to 7.76% while maintaining SlotAcc (55.29%) and achieving best occupancy performance (OccMAE 1.60 vs. 1.52 source-only; OccExact 24.61%).

3.5.1.2 Cross-Frequency Domain Shift.

Frequency-dependent propagation effects induce severe distribution shift. Source-only models completely fail at sample level (0.00% Exact Match, 21.16% Slot-wise Accuracy). Both MU-SHOT-Fi and SHOT++ recover to 0.71% Exact Match and 47% Slot-wise Accuracy with identical OccMAE (1.89), representing a 26-percentage-point improvement over source-only baseline. This recovery demonstrates that occupancy-weighted information maximization combined with rotation SSL effectively realigns features under frequency-induced domain shift.

3.5.1.3 Combined Environment and Frequency Shift.

The combined room-and-frequency shift induces the most severe domain mismatch, yielding near-zero Exact Match for all methods and weak source-only generalization (19.61% Slot-wise Accuracy). MU-SHOT-Fi achieves best Slot-wise Accuracy (41.97%), lowest Occupancy MAE (2.12 vs. 3.00 source-only), and highest Occupancy Exact Match (10.93%). While MU-SHOT-Fi ties SHOT++ in Exact Match (0.18%), its consistent gains across slot correctness and occupancy metrics indicate that occupancy-aware information maximization yields more stable adaptation under compound domain shifts.

Notably, source-only models in cross-frequency and combined-shift settings attain relatively high ActivityF1 (25.69%, 24.99%) despite 0% Exact Match. This discrepancy arises because ActivityF1 is computed only over occupied ground-truth slots and does not penalize errors in occupancy counting, extra predicted users, or inconsistent slot assignment after matching. Exact Match and SlotAcc instead reflect the

full end-task requirement—correct counting and coherent multi-user slot configuration—so improvements in occupancy consistency substantially increase these metrics without proportional gains in ActivityF1.

3.5.2 Ablation Studies

3.5.2.1 Integrating CPC within MU-SHOT-Fi

Table IV depicts that adding CPC to MU-SHOT-Fi provides no benefit and often hurts performance. In Cross-Room, Exact Match drops from 7.76% to 6.34%. In Combined shift, Slot-wise Acc decreases from 41.97% to 41.58%. Only Cross-Frequency shows marginal improvement (Exact Match: 0.71% to 1.41%), but this is offset by degradation elsewhere. The degradation of CPC in multi-user scenarios stems from a fundamental conflict: CPC enforces sample-level temporal consistency by predicting future windows from past context, but multi-user prediction requires permutation-invariant slot outputs. Hungarian matching reorders slots independently per sample, breaking the temporal coherence CPC attempts to learn. This mismatch causes CPC's gradients to conflict with the slot-prediction objective, degrading rather than improving adaptation. We thus exclude CPC from MU-SHOT-Fi, using only rotation SSL for spatial regularization combined with occupancy-weighted information maximization. This design achieves the best average performance across metrics (Table 3.3, bottom rows).

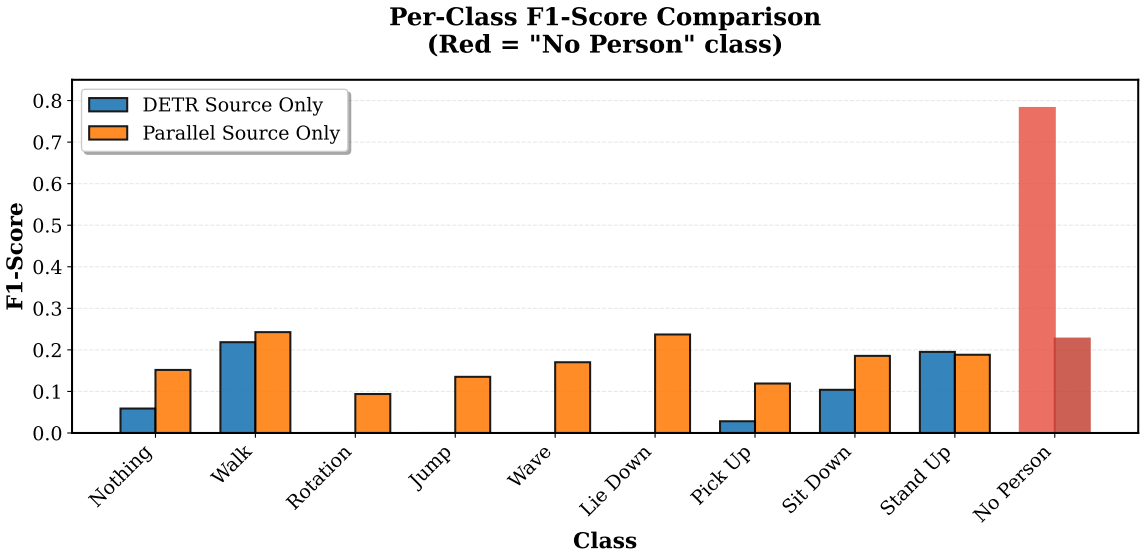


Figure 3.3: Per-class F1-scores under Cross-Frequency shift (source-only). DETR achieves 77.1% F1 on "no person" (red) but only 14.5% average on activities, demonstrating majority-class exploitation. Parallel heads show more balanced performance (23.1% vs 16.3%).

3.5.2.2 Set-Prediction Head Ablation: Pooled Parallel vs. Query-Based Decoder

In Table 3.5, we compare two architectures for multi-user slot prediction: (i) a pooled parallel classifier that averages CNN features globally then predicts each slot independently, and (ii) a query-based transformer decoder that preserves spatial structure and uses attention to predict slots [62]. Both use Hungarian matching during source training and occupancy-weighted GENT with rotation SSL during adaptation. Table 3.5 shows the query-based decoder achieves higher source-only Slot-wise Accuracy (56.23% vs. 31.95%), but this comes from predicting "no person" rather than learning activities. That is, ActivityF1 is only 8.51% compared to 22.19% for the pooled classifier. Fig. 3.3 confirms this under cross-frequency: the decoder attains 77.1% F1 on "no person" but 14.5% on activities, while the pooled classifier is balanced

Table 3.5: WiMANS set-prediction head ablation: pooled parallel slot classifier vs. query-based transformer decoder (DETR-style). Both variants use the same SFUDA pipeline (Hungarian matching in source training; occupancy-weighted GENT and rotation-based self-supervised learning during target adaptation). Values are mean±std across runs. Best and second-best within each scenario are **bolded** and underlined, respectively.

Scenario	Architecture	Slot-wise Acc (%) ↑	Activity F1 (%) ↑	Exact Match (%) ↑	Occ. MAE ↓	Δ Slot-wise Acc (%)
Cross-Room	Source-only (Pooled Parallel)	55.09 ± 2.47	15.89 ± 0.47	2.12 ± 0.43	1.52 ± 0.24	-
	MU-SHOT-Fi (Pooled Parallel)	55.29 ± 1.98	**17.08 ± 1.15**	7.76 ± 1.51	1.60 ± 0.25	**+0.20**
	Source-only (Query Decoder)	**58.55 ± 0.25**	12.45 ± 0.50	**8.64 ± 1.31**	1.53 ± 0.07	-
	MU-SHOT-Fi (Query Decoder)	58.00 ± 1.06	16.07 ± 0.36	7.59 ± 0.99	**1.48 ± 0.15**	-0.55
Cross-Frequency	Source-only (Pooled Parallel)	21.16 ± 4.71	**25.69 ± 0.97**	0.00 ± 0.00	2.92 ± 0.33	-
	MU-SHOT-Fi (Pooled Parallel)	46.97 ± 1.35	18.82 ± 1.50	0.71 ± 0.25	1.89 ± 0.15	**+25.81**
	Source-only (Query Decoder)	**55.87 ± 4.26**	8.34 ± 1.52	**3.35 ± 2.81**	1.74 ± 0.25	-
	MU-SHOT-Fi (Query Decoder)	54.02 ± 2.38	19.26 ± 3.71	1.41 ± 0.25	**1.65 ± 0.24**	-1.85
Combined	Source-only (Pooled Parallel)	19.61 ± 4.08	**24.99 ± 2.53**	0.00 ± 0.00	3.00 ± 0.35	-
	MU-SHOT-Fi (Pooled Parallel)	41.97 ± 2.39	18.47 ± 2.91	0.18 ± 0.25	2.12 ± 0.12	**+22.36**
	Source-only (Query Decoder)	**54.26 ± 2.49**	4.74 ± 0.88	0.35 ± 0.25	1.94 ± 0.99	-
	MU-SHOT-Fi (Query Decoder)	53.35 ± 1.75	13.48 ± 2.36	**1.41 ± 0.25**	**1.66 ± 0.07**	-0.91
Average	Source-only (Pooled Parallel)	31.95 ± 3.75	**22.19 ± 1.32**	0.71 ± 0.14	2.48 ± 0.31	-
	MU-SHOT-Fi (Pooled Parallel)	48.08 ± 1.91	18.12 ± 1.85	2.88 ± 0.67	1.87 ± 0.17	**+16.13**
	Source-only (Query Decoder)	**56.23 ± 2.33**	8.51 ± 0.97	**4.11 ± 1.46**	1.74 ± 0.44	-
	MU-SHOT-Fi (Query Decoder)	55.12 ± 1.73	16.27 ± 2.14	3.47 ± 0.50	**1.60 ± 0.15**	-1.11

Note: Δ Slot-wise Acc. (%) = (Adapted) − (Source-only), computed using the mean Slot-wise Accuracy for each scenario. Adapted variants use occupancy-weighted GENT, Hungarian matching, and rotation-based self-supervised learning. The pooled parallel classifier uses global average pooling followed by per-slot linear classification. The query-based decoder preserves spatial tokens and predicts slots using a 6-layer transformer decoder with learned queries.

(23.1% vs. 16.3%). During adaptation, the pooled classifier improves Slot-wise Accuracy by +16.13%, while the decoder degrades (−1.11%). This shows the decoder's majority-class bias harms adaptation. We adopt the pooled parallel architecture for lower complexity and better adaptation.

Table 3.6: Hyperparameter sensitivity under Cross-Room shift ($\lambda_{\text{ent}} = 1.0$ fixed). Values are mean ± std across 3 runs.

λ_{rot}	Exact Match (%) ↑	Occ. Exact Match (%) ↑
0.0 (SHOT-IM)	2.82 ± 2.94	17.99 ± 7.68
0.1	7.41 ± 2.41	22.93 ± 4.62
0.5 (default)	7.41 ± 2.41	22.93 ± 4.62
1.0	7.41 ± 2.41	22.93 ± 4.62

3.5.2.3 Hyperparameter Sensitivity

We analyze the sensitivity of MU-SHOT-Fi to the self-supervised loss weight λ_{rot} under Cross-Room shift, with $\lambda_{\text{ent}} = 1.0$ fixed. Table 3.6 reports Exact Match and Occupancy Exact Match across $\lambda_{\text{rot}} \in \{0.0, 0.1, 0.5, 1.0\}$.

We note that performance remains stable across all nonzero configurations, with identical Exact Match (7.41%) and Occupancy Exact Match (22.93%) for $\lambda_{\text{rot}} \in \{0.1, 0.5, 1.0\}$. This stability indicates that adaptation is primarily constrained by the source representation quality and task difficulty, rather than the precise loss weighting — as long as both $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{rot}}$ contribute meaningful gradients toward the frozen classifier hypothesis, the backbone converges to the same solution, consistent with findings in SHOT++ [120]. In contrast, setting $\lambda_{\text{rot}} = 0$ (equivalent to SHOT-IM without rotation SSL) causes Exact Match to drop to 2.82% and Occupancy Exact Match to 17.99%, with substantially higher variance across runs, confirming that rotation SSL is necessary for stable adaptation but that its precise weight does not affect the converged solution once active.

Table 3.7: Widar 3.0: Source-free domain adaptation results under three domain shifts. Values are mean ± std across runs. Higher is better (↑). Best within each scenario is **bolded** and second-best is underlined.

Scenario	Method	Accuracy (%) ↑	F1-Macro (%) ↑	Gain (%)
Cross-Room	Source Only	69.09 ± 0.57	68.37 ± 0.58	-
	SHOT-IM	78.19 ± 1.13	77.08 ± 0.89	+9.10
	+ SSL (SHOT++)	79.16 ± 1.88	77.97 ± 1.54	+10.07
	+ CPC (SU-SHOT-Fi)	**79.91 ± 1.72**	**78.67 ± 1.21**	**+10.82**
Cross-Torso	Source Only	87.73 ± 2.28	88.15 ± 1.83	-
	SHOT-IM	89.07 ± 1.75	89.61 ± 1.42	+1.34
	+ SSL (SHOT++)	**89.69 ± 1.74**	**90.25 ± 1.41**	+1.96
	+ CPC (SU-SHOT-Fi)	**89.69 ± 2.00**	**90.25 ± 1.58**	+1.96
Cross-Face	Source Only	83.73 ± 2.82	84.41 ± 2.48	-
	SHOT-IM	87.56 ± 1.08	88.27 ± 0.90	+3.83
	+ SSL (SHOT++)	**87.64 ± 1.11**	**88.39 ± 0.91**	+3.91
	+ CPC (SU-SHOT-Fi)	**87.64 ± 1.32**	88.26 ± 1.01	+3.91
Average	Source Only	80.18 ± 1.89	80.31 ± 1.63	-
	SHOT-IM	84.94 ± 1.32	84.99 ± 1.07	+4.76
	+ SSL (SHOT++)	85.50 ± 1.58	85.54 ± 1.29	+5.32
	+ CPC (SU-SHOT-Fi)	**85.75 ± 1.68**	**85.73 ± 1.27**	**+5.57**

Note: Average row shows mean performance across the three domain shift scenarios. Per-class F1-scores are detailed in Table 3.8.

3.5.2.4 SU-SHOT-Fi Results and Analysis

Table 3.7 reports classification accuracy across three domain shift scenarios. SU-SHOT-Fi achieves best or tied-best performance in all settings, with gains scaling inversely with source-only baseline strength. Source baselines reflect varying domain shift severity: Cross-Room (69.09%), Cross-Face (83.73%), Cross-Torso (87.73%). Adaptation gains correspondingly decrease from +10.82% (Cross-Room) to +3.91% (Cross-Face) to +1.96% (Cross-Torso). This inverse scaling aligns with domain adaptation theory [128]: when source and target distributions exhibit low divergence, source features already generalize well, constraining adaptation headroom.

SU-SHOT-Fi's advantage over SHOT++ is largest in Cross-Room (+0.75%), where CPC-based temporal modeling exploits domain-invariant periodic patterns (e.g., stride rhythms) that remain stable despite environmental layout changes. In Cross-Torso and Cross-Face, where source baselines exceed 83%, all adaptation methods (SHOT, SHOT++, SU-SHOT-Fi) converge to similar performance as the adaptation ceiling is approached. Averaged across scenarios, SU-SHOT-Fi improves over SHOT-IM by 0.81% and over source-only by 5.57%.

Table 3.8 reports per-class F1-scores. SU-SHOT-Fi achieves best average performance on 5 of 6 gestures. Performance heterogeneity reveals underlying CSI signature characteristics: *Slide* (74.11% F1) produces low Doppler content and gradual amplitude variations that blend with background noise under domain shift, while *Draw-Zigzag(H)* (92.11% F1) generates pronounced multipath variations from rapid directional reversals, creating robust discriminative signatures.

SU-SHOT-Fi yields largest gains on temporal-rich gestures: *Clap* improves by 7.78% (81.00% $\rightarrow$ 88.78%) and *Sweep* by 4.78%. This pattern validates CPC's temporal modeling: clapping produces periodic amplitude bursts with consistent inter-burst intervals that CPC captures through window-level prediction, while sweeping creates continuous Doppler shifts spanning multiple temporal windows. In contrast, gestures with static phases interspersed with motion—*Push&Pull* (+2.66%) and *Draw-O(H)* (+5.34%)—benefit less from CPC, as temporal coherence is disrupted by motion-pause transitions where rotation SSL's spatial regularization dominates.

Table 3.8: Widar 3.0: Per-class F1-scores (%) across three domain shifts. Results show average performance over 3 runs. Best within each scenario is **bolded** and second-best is underlined.

Gesture	Method	Cross-Room	Cross-Torso	Cross-Face	Average
Clap	Source Only	72.67	83.00	87.33	81.00
	SHOT-IM	84.00	86.67	**91.00**	87.22
	+ SSL (SHOT++)	86.33	87.33	90.00	87.89
	+ CPC (SU-SHOT-Fi)	**87.67**	**87.67**	**91.00**	**88.78**
Draw-O(H)	Source Only	55.67	88.00	91.33	78.33
	SHOT-IM	67.67	88.67	**92.33**	82.89
	+ SSL (SHOT++)	68.67	89.67	91.67	83.34
	+ CPC (SU-SHOT-Fi)	**69.67**	**90.33**	91.00	**83.67**
Draw-Zigzag(H)	Source Only	71.67	97.33	93.00	87.33
	SHOT-IM	82.33	**98.33**	**95.00**	91.89
	+ SSL (SHOT++)	82.00	**98.33**	**95.00**	91.78
	+ CPC (SU-SHOT-Fi)	**83.67**	**98.33**	94.33	**92.11**
Push&Pull	Source Only	88.67	90.67	89.33	89.56
	SHOT-IM	90.33	91.67	93.00	91.67
	+ SSL (SHOT++)	**91.33**	**92.67**	**94.00**	**92.67**
	+ CPC (SU-SHOT-Fi)	**91.33**	92.00	93.33	92.22
Slide	Source Only	40.67	83.33	74.00	66.00
	SHOT-IM	57.33	82.67	79.67	73.22
	+ SSL (SHOT++)	58.00	**83.67**	80.00	73.89
	+ CPC (SU-SHOT-Fi)	**58.33**	83.33	**80.67**	**74.11**
Sweep	Source Only	77.33	86.00	71.67	78.33
	SHOT-IM	76.00	89.00	79.33	81.44
	+ SSL (SHOT++)	80.67	**90.00**	**79.67**	**83.44**
	+ CPC (SU-SHOT-Fi)	**81.00**	89.67	78.67	83.11
Macro Avg	Source Only	67.78	88.06	84.44	80.09
	SHOT-IM	76.28	89.50	**88.39**	84.72
	+ SSL (SHOT++)	77.83	**90.28**	**88.39**	85.50
	+ CPC (SU-SHOT-Fi)	**78.61**	90.22	88.17	**85.67**

Note: Macro Avg computed across the six gesture classes. Average column shows mean F1-scores across the three domain shift scenarios.

3.6 Conclusion

This thesis proposed *MU-SHOT-Fi*, the first Source-Free Unsupervised Domain Adaptation (SFUDA) framework tailored for multi-user Wi-Fi sensing. Multi-user scenarios introduce fundamental computational bottlenecks, including variable occupancy, signal entanglement from concurrent activities, and severe class imbalance. We demonstrated that standard adaptation mechanisms, such as pseudo-labeling,

fail in these settings by converging toward the dominant "no person" class, thereby ignoring actual activities. To circumvent this collapse, we introduced an occupancy-weighted information maximization objective that dynamically down-weights empty slots during diversity regularization, enabling robust adaptation across variable user counts without requiring explicit occupancy estimation. Furthermore, we integrated binary rotation SSL to provide essential spatial regularization on the frequency-time structure of CSI. Extensive evaluation on the WiMANS dataset confirms that the proposed framework delivers consistent performance gains that scale inversely with domain shift severity, establishing a principled methodology for privacy-preserving, multi-user wireless sensing in dynamic environments.

Chapter 4

Contrastive Predictive Coding with Compression for Enhanced Channel State Feedback in Wireless Networks

4.1 AI-Driven CSI Feedback: A Review

This section reviews existing AI-driven CSI feedback methods that support 3GPP objectives. 3GPP categorizes AI-driven CSI compression and reconstruction into three distinct types [129], as summarized in **Table-4.1**.

• *Type-1* involves joint training of the full model at a single side, either the BS or UE, followed with a split deployment at UE and BS. This approach is simple and resource-efficient, but lacks post-deployment adaptability, making it less effective in dynamic environments.

• *Type-2* enables collaborative training between the BS and UE where the same model instance is split between UE and BS. This method enables end-to-end optimization through the exchange of intermediate features, gradients, or activations. While this improves adaptability to real-time channel variations, it introduces significant

communication overhead, latency, and system complexity.

• *Type-3* refers to separate training of different models at each side. This setup, which may also include federated or distributed learning, offers modularity and improved generalization, but poses challenges in synchronization, cross-device compatibility, and computational efficiency at the UE.

A graphical illustration of the model training types studied in 3GPP Rel-18 is given in Fig. 4.1.

Current research works are predominantly based on Type-1 training due to its implementation simplicity and deployment practicality. However, a growing number of studies have begun exploring Type-3 training recently, motivated by its potential for improved generalization in heterogeneous wireless environments. Notable works leveraging Type-1 training include [56–58,130–132], while emerging studies investigating Type-3 include [133–135]. The following subsections review representative works in both categories.

4.1.1 Type-1: Joint Training at One Side

Early CSI compression efforts, such as [130], demonstrated that an autoencoder trained on uplink CSI at the BS could generalize across frequencies. The encoder can then be offloaded to the UE for CSI feedback, while the decoder remains at the BS. Building on those principles, [131] introduced a two-part training objective to improve reconstruction without requiring access to clean training data. The first leverages Stein's Unbiased Risk Estimator (SURE) to optimize reconstruction from noisy CSI, while the second adds a compression regularization term (derived via a

Table 4.1: Summary of Related Works on CSI Feedback Using AI/ML Techniques.

Paper	Type	Train. Mode	Architecture	Prediction	Generalization	Datasets	Eval. Metrics	Loss Function
[53]	*Type-1* (BS)	Offline	Conv Autoencoder + Quantizer	✗	✗	Simulated using COST 2100 model in indoor and outdoor scenarios	NMSE	MSE + Quantization regularizer
[54]	*Type-1* (BS)	Offline	Full-Conv Autoencoder	✗	✗	Simulated using COST 2100 model in indoor and outdoor scenarios	NMSE, Cosine Correlation	MSE
[130]	*Type-1* (BS)	Offline	Conv Autoencoder	✗	✓	Simulated using MATLAB in an urban microcell scenario	NMSE, Cosine Similarity, Per-User Rate	MSE
[131]	*Type-1* (BS)	Offline	Conv Variational Autoencoder	✗	✗	CSiNet Indoor Dataset	NMSE	(MSE or SURE) + MI upper-bound regularizer
[132]	*Type-1* (BS)	Offline	Conv Autoencoder	✗	✗	Simulated using COST 2100 model in indoor and outdoor scenarios	NMSE	MSE
[56]	*Type-1* (BS)	Offline	Conv Encoder + Transformer Decoder	✗	✗	Simulated using Quadriga v2.8.1 in 3GPP-38.901 UMi and UMa	NMSE	MSE + Quantization Loss (codebook + encoder terms)
[57]	*Type-1* (BS)	Offline		✗	✗	Simulated using COST 2100 model in indoor and outdoor scenarios	NMSE	
[58]	*Type-1* (UE)	Online	Conv Autoencoder	✗	✗	Simulated using CDL model in 3GPP TR 38.901	NMSE	MSE + Regularizer
[135]	*Type-3*	Offline	Conv Autoencoder	✗	✗	Simulated using 3GPP TR 38.901 UMa scenario	NMSE, SGCS, SE, Overhead	SGCS + NMSE
[134]	*Type-3*	Online	Conv Autoencoder	✗	✓	Simulated using COST 2100 and CDL from 3GPP TR 38.901 R15	NMSE, Overhead	MSE + Regularizer
Ours	***Type-1 (BS)***	**Offline**	**Conv Autoencoder + CPC**	✓	✓	**3GPP-compliant datasets (Nokia, Oppo, CATT) [136]**	**SGCS, Complexity, Inference Time, Overhead**	**InfoNCE + (1-SGCS)**

NMSE: Normalized Mean Square Error, CDL: Clustered Delay Line, SGCS: Squared Generalized Cosine Similarity, SE: Spectral Efficiency, UMi: Urban Micro, UMa: Urban Macro, FDD: Frequency Division Duplex, MSE: Mean Square Error.

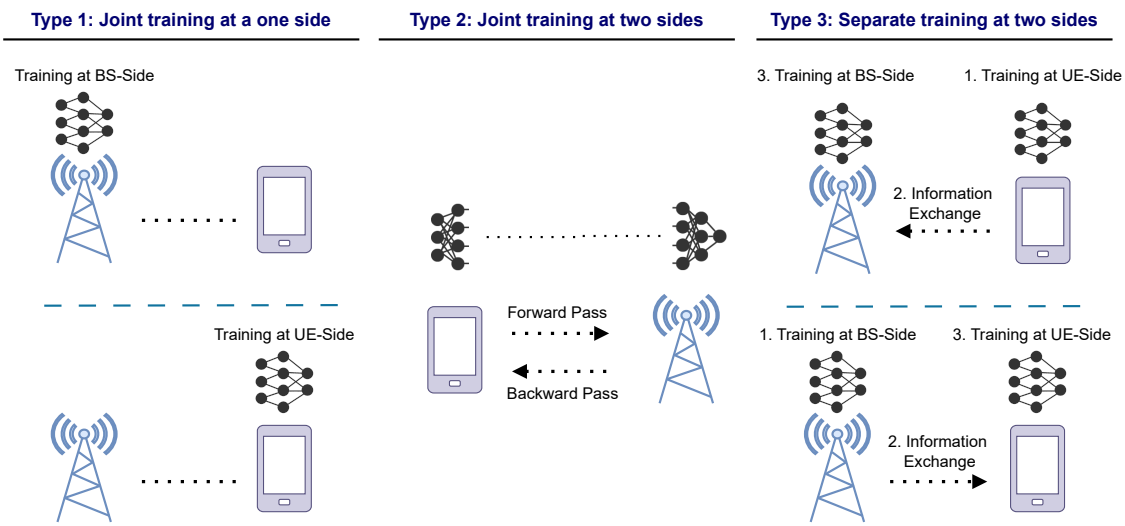


Figure 4.1: Overview of the model training types studied in 3GPP- Release 18: (1) Joint training of the encoder and decoder at a single side (BS or UE), (2) Joint training across two sides with intermediate activation and gradient exchange, and (3) Independent training at each side with shared datasets. The forward and backward pass flow is illustrated for each configuration.

variational upper bound on mutual information) to encourage a compact latent representation. Together, these terms balance compression efficiency and reconstruction fidelity. The authors in [132] applied deep learning to jointly address channel estimation and feedback in Frequency Division Duplex (FDD) massive MIMO systems. Their primary model, CEFnet, consists of four main stages: CSI channel estimation using pilots, a feature extraction network to process the estimated channel, an encoder to perform dimensionality reduction, and a quantization module to enable efficient transmission. On the BS side, an additional refinement stage is introduced to enhance the decompressed CSI using CNN. To address the computational limitations of the UE, the authors also proposed an alternative model, CEFnet-B, which skips channel estimation at the UE. Instead, raw pilot signals are directly compressed and quantized, and after decompression at the BS, CSI is estimated and subsequently

refined. This design shifts the computational burden entirely to the BS.

Similarly, in [56], the authors considered more computations at the BS side. The architecture employs a lightweight encoder for CSI compression followed by the quantization layer at UE. In contrast, the decoder at the BS is more complex, as after dequantization a transformer-based network equipped with multi-head attention and positional encoding has been applied to learn interactions across distant antenna and subcarrier elements to effectively reconstruct the CSI. End-to-end offline training was performed at the BS, after which the encoder and quantization layers were transferred to the UE.

[53] proposed CQNet, which inserts a learnable quantizer between encoder and decoder to jointly optimize compression and quantization. While this work represents an important step in learned CSI compression, it operates entirely on static CSI snapshots and does not model temporal dependencies or mitigate channel aging [23–25]. In [54], the authors proposed DeepCMC, a fully convolutional autoencoder with entropy coding that approaches the rate-distortion bound for CSI feedback. Similarly, this work focuses exclusively on compression fidelity, with no mechanism for temporal prediction or robustness to CSI aging [23–25].

In [57], the authors proposed LWNet, which further reduced UE computational demands by processing CSI samples in parallel using convolutional filters with different kernel sizes to extract features at multiple granularities before merging. An Efficient Channel Attention (ECA) module enabled global pooling without dimensionality reduction, while the decoder used Omni-Dimensional Dynamic Convolutions (ODConv) to optimize computation. This design emphasized a lightweight UE encoder while offloading complexity to the BS. Lastly, [58] tackled key limitations of the

offline training paradigm, which is predominantly used in Type-1 settings, by incorporating an online learning mechanism directly at the UE. Their approach enabled local fine-tuning of both the encoder and decoder modules before transmitting the updated model back to the BS. However, while using online learning improves adaptability, deploying complex decoder architectures—such as those in [56] may not be feasible, as it is limited by the UE's limited computational resources. Moreover, the method did not explicitly account for the quality of incoming data; thus fine-tuning on low-quality or noisy CSI samples could inadvertently degrade reconstruction performance.

4.1.2 Type-3: Separate Training at Two Sides

As mentioned earlier, there are only a handful of recent research works that fall under this category. Type-3 decouples encoder and decoder training between the UE and BS, primarily to enhance privacy and reduce communication overhead. [135] proposed a strategy where the UE trains the encoder offline and transmits latent representations to the BS, and the decoder is trained separately—minimizing data exchange and privacy risks. Another way to implement Type-3 is via Federated Learning (FL), which enables multiple UEs to contribute to a global model deployed on a BS by sending their local model weights or gradients. As shown in [133, 134], distributed training across UEs helps preserve privacy and reduces reliance on the BS, as it aggregates local updates rather than performing centralized training. These approaches also incorporate compressed sensing, knowledge distillation, and local fine-tuning to personalize models with minimal overhead.

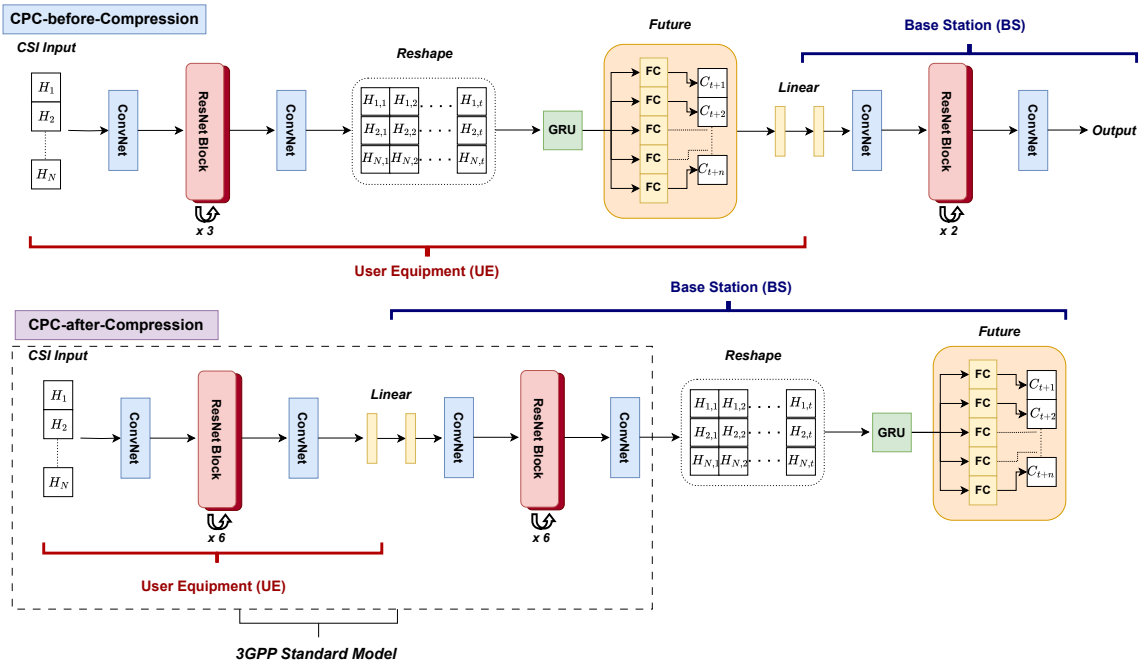


Figure 4.2: The CPC-after-Compression model (below) builds on the 3GPP standard backbone by adding CPC to compressed representations, while CPC-before-Compression (top) applies CPC directly on raw CSI features.

4.1.3 Key Takeaways and Summary

Despite recent advances in AI-driven CSI feedback methods, existing CSI compression studies [56–58, 130–135] do not address the problem of CSI prediction jointly. Moreover, the existing CSI prediction works have neither investigated the benefits of representation learning, nor explored the use of open-access 3GPP-compliant datasets for validation. However, a recent study in [66] considered prediction directly on raw CSI vectors using a Bi-LSTM, either at the UE before compression or at the BS after decompression, with quantization applied only on the latent space. While jointly trains compression and prediction using only cosine similarity, our CPC-before-Compression uses a combined 1-SGCS and InfoNCE loss to explicitly encourage temporal predictive

structure in the latent space.

4.2 Proposed Age-Aware CPC-based CSI Feedback

In this section, we propose a 3GPP-compliant AI solution that explicitly targets channel aging [23–25] and supports robust generalization across deployment scenarios. Specifically, we incorporate CSI prediction through CPC [124] into the 3GPP standard compression model. This integrated framework, referred to as *CPC with Compression*, leverages temporal dependencies in the CSI sequence to improve both prediction robustness and compression effectiveness. By forecasting future CSI representations, the model compensates for delays between CSI acquisition and usage, which is critical in highly dynamic environments. On the other hand, generalization is achieved through a comprehensive evaluation across diverse 3GPP-compliant datasets, highlighting the model's adaptability beyond the training environment. Furthermore, we incorporate structured pruning and aggressive quantization to reduce model size, enabling efficient deployment on resource-constrained UEs without compromising performance. For our training paradigm, we build upon 3GPP Type-1 training framework due to its simplicity, low deployment complexity, and compatibility with telecommunication systems.

4.2.1 AI/ML Models Considered in 3GPP

The compression model considered in 3GPP from [137] employs a ResNet-based architecture for end-to-end training of both encoder and decoder components. The encoder begins with a convolutional layer that maps input CSI $\mathbf{H}_t$ to 64 feature maps, followed by six ResNet blocks—each consisting of two convolutional layers with skip

connections. By concatenating the input with the block output, the architecture enhances gradient flow and mitigates vanishing gradient issues. The output from the final ResNet block is projected into a 32-dimensional latent vector $\mathbf{z}_t$ via a quantized linear layer, yielding a compact yet informative representation that reduces CSI feedback overhead. On the decoder side, $\mathbf{z}_t$ is dequantized and progressively reconstructed back to the original CSI $\tilde{\mathbf{H}}_t$, mirroring the encoder in structure. To quantify reconstruction quality, we adopt the Squared Generalized Cosine Similarity (SGCS) metric, defined as:

$$\text{SGCS} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{1+\frac{\sum_d |\mathbf{H}_i - \tilde{\mathbf{H}}_i|^2}{\sum_d |\mathbf{H}_i|^2 + \epsilon}}, \tag{4.1}$$

where $\mathbf{H}_i$ and $\tilde{\mathbf{H}}_i$ denote the ground-truth and reconstructed CSI respectively, B is the batch size, and $\epsilon = 1 \times 10^{-10}$ is added to avoid division by zero. Higher SGCS values indicate stronger geometric alignment between the original and reconstructed CSI. The model is then trained by minimizing:

$$\mathcal{L}_{\text{SGCS}} = 1 - \text{SGCS}, \tag{4.2}$$

such that minimizing $\mathcal{L}_{\text{SGCS}}$ is equivalent to maximizing the SGCS metric.

The overall design intentionally avoids complex modules such as attention mechanisms or advanced preprocessing, prioritizing computational efficiency. This leads to reduced inference latency and makes the model well-suited for deployment in resource-constrained environments.

Leveraging on models considered in 3GPP, we develop two design variants that address channel aging [23–25]. The variants are described in the following.

Algorithm 3: CPC-before-Compression

Require: CSI sequence $\mathbf{H}_{t-L+1:t}$ at UE

Ensure: Compressed feedback $\mathbf{z}_t$; reconstructed predicted representations $\{\hat{\mathbf{C}}_{t+k}\}_{k=1}^{T}$ at BS

UE Processing:

$\mathbf{F}_{t-L+1:t} \leftarrow \text{Encoder}_\theta(\mathbf{H}_{t-L+1:t})$

$\mathbf{F}_{\text{flat}} \leftarrow \text{ReshapeToSequence}(\mathbf{F}_{t-L+1:t})$

$\mathbf{C}_t \leftarrow \text{GRU}(\mathbf{F}_{\text{flat}})$ (temporal modeling before compression)

for $k = 1$ **to** T **do**

  $\mathbf{C}_{t+k} \leftarrow \text{FC}_k(\mathbf{C}_t)$

end for

$\mathbf{z}_{t+k} \leftarrow \text{Quantize}(\text{LinearCompress}(\mathbf{C}_{t+k}))$ (for each $k = 1, \ldots, T$)

Transmit $\{\mathbf{z}_{t+k}\}_{k=1}^{T}$ via uplink

BS Processing:

$\mathbf{C}'_{t+k} \leftarrow \text{Dequantize}(\mathbf{z}_{t+k})$ (for each $k = 1, \ldots, T$)

$\hat{\mathbf{C}}_{t+k} \leftarrow \text{LinearExpand}(\mathbf{C}'_{t+k})$ (for each $k = 1, \ldots, T$)

$\mathcal{L}_{\text{SGCS}} \leftarrow$ Using (4.2) $(\{\mathbf{C}_{t+k}\}, \{\hat{\mathbf{C}}_{t+k}\})$

$\mathcal{L}_{\text{InfoNCE}} \leftarrow$ Using (4.3) $(\mathbf{C}_t, \{\mathbf{C}_{t+k}\})$

$\mathcal{L}_{\text{total}} \leftarrow$ Using (4.5)

return $\mathbf{z}_t, \{\hat{\mathbf{C}}_{t+k}\}_{k=1}^{T}$

4.2.2 CPC-before-Compression

Applying CPC before compression allows temporal features to be encoded prior to dimensionality reduction. This design facilitates easier reconstruction (as shown in Fig. 4.2 and Algorithm 3) because the representation is already mapped using an *autoregressive GRU-based model* [138] as a predictive layer to capture temporal dependencies and predict future CSI representations. The architecture comprises an encoder, a predictive module, a compression layer, and an optional decoder since the representation is restored from the first layer. However, since most computations occur on the encoder side, this design requires more resources, leaving less capacity for optimization on resource-limited UE devices.

The encoder takes as input a sequence of L consecutive CSI matrices $\mathbf{H}_{t-L+1:t}$, each mapped to 64 feature maps via an initial convolutional layer, followed by three ResNet blocks and a final 1×1 convolution, producing the feature sequence $\mathbf{F}_{t-L+1:t}$. The features are then reshaped into a temporal sequence $\mathbf{F}_{\text{flat}}$ and passed to a GRU module, which summarizes the temporal context into a 128-dimensional context vector $\mathbf{C}_t$. A set of T dedicated fully-connected layers then projects $\mathbf{C}_t$ into future predicted representations $\{\mathbf{C}_{t+k}\}_{k=1}^{T}$, each of 128 dimensions. The predicted representations are subsequently compressed into a 32-dimensional latent vector $\mathbf{z}_t$ via a quantized linear layer and transmitted to the BS. At the BS, $\mathbf{z}_t$ is dequantized and expanded back to 128 dimensions via a linear layer, followed by two residual blocks, yielding the reconstructed predicted representations $\{\hat{\mathbf{C}}_{t+k}\}_{k=1}^{T}$.

To train the model, we introduce the InfoNCE loss as given below [124]. InfoNCE loss encourages $\mathbf{C}_t$ to be predictive of future representations in the latent space:

$$\mathcal{L}_{\text{InfoNCE}} = \frac{1}{T}\sum_{k=1}^{T} \ell_k, \tag{4.3}$$

$$\ell_k = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\frac{\mathbf{C}_{t+k}^{(i)} \cdot \mathbf{C}_t^{(i)}}{\tau}\right)}{\sum_{j=1}^{B} \exp\left(\frac{\mathbf{C}_{t+k}^{(i)} \cdot \mathbf{C}_t^{(j)}}{\tau}\right)}. \tag{4.4}$$

where T is the number of future time steps, τ is the temperature parameter, and B is the batch size. The SGCS loss is computed between the original predicted representations $\{\mathbf{C}_{t+k}\}$ and their reconstructions $\{\hat{\mathbf{C}}_{t+k}\}$. The total training loss is then given by:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{SGCS}} + (1-\alpha)\mathcal{L}_{\text{InfoNCE}}, \tag{4.5}$$

Algorithm 4: CPC-after-Compression

Require: CSI matrix $\mathbf{H}_t$ at UE

Ensure: Compressed feedback $\mathbf{z}_t$; reconstructed CSI $\widetilde{\mathbf{H}}_t$ at BS; future representations $\{\mathbf{C}_{t+k}\}_{k=1}^{T}$

UE Processing:
$\mathbf{F}_t \leftarrow \text{Encoder}_\theta(\mathbf{H}_t)$
$\mathbf{z}_t \leftarrow \text{Quantize}(\text{LinearCompress}(\mathbf{F}_t))$
Transmit $\mathbf{z}_t$ via uplink
BS Processing:
$\mathbf{F}'_t \leftarrow \text{Dequantize}(\mathbf{z}_t)$
$\widetilde{\mathbf{H}}_t \leftarrow \text{Decoder}_\phi(\mathbf{F}'_t)$
$\widetilde{\mathbf{H}}_{\text{flat}} \leftarrow \text{ReshapeToSequence}(\widetilde{\mathbf{H}}_{t-L+1:t})$
$\mathbf{C}_t \leftarrow \text{GRU}(\widetilde{\mathbf{H}}_{\text{flat}})$ (CPC after reconstruction)
for $k = 1$ **to** T **do**
    $\mathbf{C}_{t+k} \leftarrow \text{FC}_k(\mathbf{C}_t)$
end for
$\mathcal{L}_{\text{SGCS}} \leftarrow$ Using (4.2) $(\mathbf{H}_t, \widetilde{\mathbf{H}}_t)$
$\mathcal{L}_{\text{InfoNCE}} \leftarrow$ Using (4.3) $(\mathbf{C}_t, \{\mathbf{C}_{t+k}\})$
$\mathcal{L}_{\text{total}} \leftarrow$ Using (4.5)
return $\mathbf{z}_t, \widetilde{\mathbf{H}}_t, \{\mathbf{C}_{t+k}\}_{k=1}^{T}$

where $\alpha \in [0, 1]$ controls the trade-off between reconstruction quality and temporal prediction accuracy. In our experiments, $\alpha = 0.5$ was selected empirically, as equal weighting was found to balance reconstruction fidelity and contrastive learning stability across all datasets.

4.2.3 CPC-after-Compression

Applying CPC after compression, as shown in Fig. 4.2 and Algorithm 4, is more suitable for resource-limited UE devices. Since the predictive layer contributes most to the model's complexity, performing compression first reduces the computational burden on the UE side. In this method, CPC is integrated entirely into the decoder at the BS, after the final convolutional layer responsible for reconstruction. The

encoder is identical to the 3GPP baseline, mapping each input CSI matrix $\mathbf{H}_t$ to a 32-dimensional latent vector $\mathbf{z}_t$ via a convolutional feature extractor followed by six ResNet blocks and a quantized linear layer. At the BS, $\mathbf{z}_t$ is dequantized and decoded through six ResNet blocks back to the reconstructed CSI $\tilde{\mathbf{H}}_t$. The reconstructed frames $\tilde{\mathbf{H}}_{t-L+1:t}$ are then reshaped into a temporal sequence and passed to a GRU module, which generates a 128-dimensional context vector $\mathbf{C}_t$. A set of T fully-connected layers then projects $\mathbf{C}_t$ into future predicted representations $\{\mathbf{C}_{t+k}\}_{k=1}^{T}$. The model is trained using $\mathcal{L}_{\text{total}}$ from (4.5), where $\mathcal{L}_{\text{SGCS}}$ from (4.2) is computed between $\mathbf{H}_t$ and $\tilde{\mathbf{H}}_t$, and $\mathcal{L}_{\text{InfoNCE}}$ from (4.3) is computed between $\mathbf{C}_t$ and $\{\mathbf{C}_{t+k}\}_{k=1}^{T}$.

4.2.4 Discussions and Insights

By shifting the computational load to the BS, *CPC-after-Compression* prioritizes efficiency over early feature preservation. However, this trade-off may impact prediction accuracy compared to the *CPC-before-Compression* method, as some information is lost before temporal dependencies are fully captured. Additionally, applying *CPC-after-Compression* aligns better with split learning (SL) frameworks. *Both CPC-before and CPC-after-Compression* help reduce computational overhead on the UE by offloading more processing tasks to the BS. However, *CPC-after-Compression* becomes more viable when the UE has strict resource constraints.

4.3 Experimental Setup & Results

This section discusses the experimental set-up, datasets, and validates the performance of the proposed model over a variety of datasets. We follow the agreed-upon model hyperparameters from 3GPP drafts, summarized in Table 4.3.

Table 4.2: Model parameters, storage, inference time, and computational cost (batch size = 256, 2-bit quantization).

Metric	**3GPP**	**CPC-before**	**CPC-after**
Encoder Params	872,448 (213.00 KB)	10,587,168 (2,574.13 KB)	872,448 (213.00 KB)
Decoder Params	1,358,786 (331.74 KB)	103,296 (25.22 KB)	1,810,754 (430.00 KB)
Encoder Time (ms)	2.4533	2.0984	2.4457
Decoder Time (ms)	2.5065	0.6799	2.9415
Encoder GFLOPs	0.59	10.47	0.59
Decoder GFLOPs	0.84	0.026	1.21

The evaluation is performed on proprietary datasets provided by Nokia, Oppo, and CATT, specifically `N0KIR4_113dsei.npy`, `OPPOR4_113dsei00.npy`, and `CAT0R4_113dsei01.npy` [136]. To ensure a broader evaluation, we construct a mixed dataset by combining 100k samples from each of these datasets. The datasets follow a four-dimensional structure spanning samples, real/imaginary components, sub-bands (13), and antenna ports (32), and vary in the number of samples contributed by different companies.

Table 4.2 summarizes the complexity of each model variant.

• The *3GPP baseline* maintains a compact encoder (872K parameters, 2.45 ms per batch) with moderate decoder cost.

• *CPC-before-Compression* incurs a significantly larger encoder footprint (10.6M parameters, $\sim$17.8$\times$ more GFLOPs per batch: 10.47 vs. 0.59 GFLOPs) due to the GRU operating on intermediate feature maps of dimension $64 \times 13 \times 32 = 26{,}624$. Despite this, encoder inference remains comparable (2.10 ms per batch) as both models fall into a memory-bandwidth-bound regime where wall-clock time is insensitive to arithmetic complexity [139]. Its decoder is substantially lighter (103K parameters,

0.026 GFLOPs per batch) as temporal modeling is offloaded to the encoder side.

• *CPC-after-Compression* preserves the same encoder as the baseline (0.59 GFLOPs per batch, 2.45 ms per batch) while adding moderate decoder overhead (1.81M parameters, 1.21 GFLOPs per batch) to accommodate the BS-side GRU.

All methods maintain a 64-bit communication overhead per timestep, as the quantized linear layer outputs 32 latent dimensions with 2-bit quantization.

In our setup, we follow the Type-1 joint training scheme described in [140]. For evaluation, we use the SGCS metric instead of the commonly used NMSE, as SGCS offers a more robust measure of subspace preservation and geometric alignment. Table 4.4 presents the SGCS and InfoNCE loss values for each variant.

Table 4.3: Hyperparameters and Their Values

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Optimizer	Adam	Learning Rate	10^{-4}
Number of Epochs	150	Batch Size	256
Validation Patience	25	Data Split	80:10:10
Number of Seeds	5	Time Window (L)	10
Hidden Size	128	Future Time Steps (T)	5
Temperature (τ)	0.1	Contribution parameter (α)	0.5
Uniform Quantization	2 bits		

4.3.1 Overall Performance Evaluation Results

4.3.1.1 Performance From Considered Models in 3GPP

We begin with the models considered in 3GPP, which serves as a baseline using SGCS only. The Nokia-trained model achieves an SGCS of 0.729 on its own test set and 0.735 on Oppo, while the Oppo-trained model attains 0.733 on its own data and 0.726 on Nokia. These results suggest that Nokia's and Oppo's models share

Table 4.4: SGCS (↑) and InfoNCE (↓) across datasets.

Training	Nokia	Oppo	CATT	Mixed
3GPP Standard (no-CPC) (SGCS only)				
NOKIA	0.729	0.735	0.563	0.676
OPPO	0.726	0.733	0.560	0.673
CATT	0.637	0.644	0.674	0.656
Mixed	0.725	0.732	0.614	0.691
CPC-before-Compression				
NOKIA	0.726 (0.646)	0.725 (0.639)	0.738 (0.668)	0.720 (0.664)
OPPO	0.869 (0.150)	0.869 (0.147)	0.866 (0.338)	0.867 (0.158)
CATT	0.912 (0.666)	0.912 (0.651)	0.925 (0.279)	0.917 (0.313)
Mixed	0.886 (0.186)	0.886 (0.182)	0.887 (0.260)	0.885 (0.183)
CPC-after-Compression				
NOKIA	0.730 (0.008)	0.735 (0.008)	0.566 (0.074)	0.677 (0.009)
OPPO	0.733 (0.008)	0.739 (0.008)	0.568 (0.068)	0.680 (0.009)
CATT	0.648 (0.014)	0.656 (0.014)	0.674 (0.035)	0.664 (0.010)
Mixed	0.720 (0.005)	0.728 (0.005)	0.627 (0.037)	0.692 (0.005)

consistent characteristics. On the other hand, the CATT-trained model, despite achieving 0.674 on its own test set, shows noticeably lower generalization to other datasets (SGCS between 0.637 and 0.644), likely due to both its smaller size (100k samples) and possible domain-specific variability. We note that training on a mixed dataset improved performance across the board, particularly for Nokia (0.725), Oppo (0.732), and Mixed (0.691) test sets, although generalization to CATT remained limited (0.614).

4.3.1.2 CPC-before-Compression Performance

Table 4.4 shows that CPC-before-Compression consistently outperforms the 3GPP baseline across all training and test configurations. SGCS scores range from 0.720 to

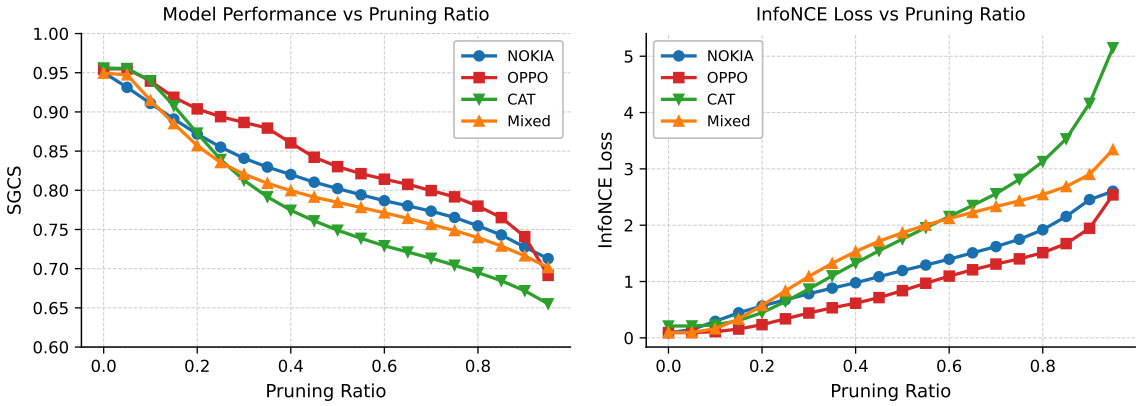
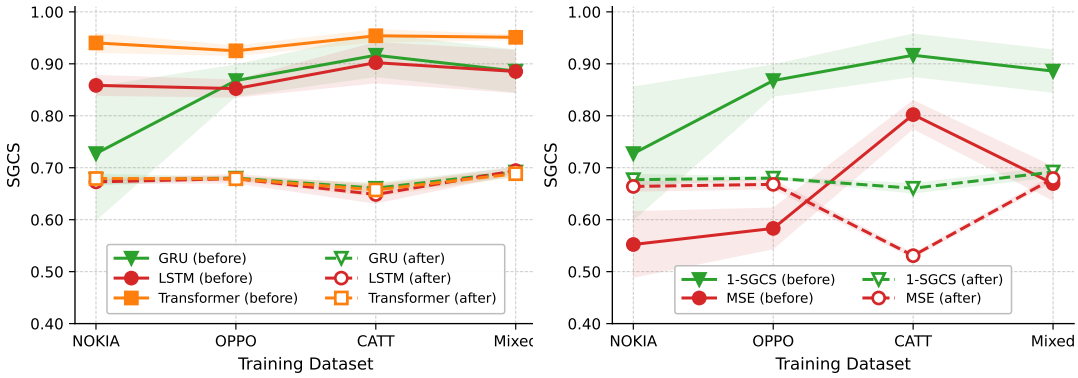


Figure 4.3: Impact of structured pruning on CPC-before-Compression (GRU, Hidden=128). The left panel reports SGCS (↑) and the right panel reports InfoNCE loss (↓) as a function of pruning ratio, across all four training datasets. Results are averaged over 5 runs.



(a) Effect of temporal backbone architecture.

(b) Effect of reconstruction loss (1-SGCS vs. MSE).

Figure 4.4: Ablation studies measuring SGCS averaged over all test datasets. Solid and dashed lines denote CPC-before-Compression and CPC-after-Compression, respectively. Shaded bands indicate standard deviation over 5 runs.

0.925, with CATT and OPPO-trained models achieving the strongest reconstruction fidelity. The NOKIA-trained model exhibits lower SGCS and higher InfoNCE loss (e.g., 0.726 / 0.646 on Nokia test), consistent with the training instability discussed in the ablation studies. Training on the mixed dataset yields balanced generalization

across all test domains (0.885–0.887), confirming robustness to domain shift. InfoNCE loss varies widely across dataset pairs (0.150–0.666), reflecting differences in temporal predictability across channel environments. To reduce model complexity, we apply post-training structured pruning to the GRU layer in CPC-before-Compression. As shown in Fig. 4.3, increasing the pruning ratio leads to a steady decline in SGCS and a sharp rise in InfoNCE loss, with performance remaining acceptable up to moderate pruning ratios before degrading significantly at higher sparsity levels.

4.3.1.3 CPC-after-Compression

achieves SGCS scores comparable to the 3GPP baseline while providing future CSI representations at the BS. Scores range from 0.566 to 0.739, with CATT test performance remaining the weakest across all training datasets, consistent with its domain-specific characteristics noted earlier. Notably, InfoNCE loss remains near zero across all configurations (0.005–0.074), suggesting that performing temporal modeling on reconstructed CSI reduces the diversity of latent representations as the information loss happens during compression and decompression. Thus, the contrastive learning loses its effectiveness. As a result, compression bottleneck is the dominant constraint in this variant.

4.3.2 Ablation Studies

4.3.2.1 Effect of Temporal Backbone Architecture

Fig. 4.4a compares GRU, LSTM, and Transformer backbones across both model variants. For CPC-after-Compression, all three backbones converge to nearly identical

Table 4.5: Effect of prediction horizon T on SGCS (↑) and InfoNCE loss (↓) averaged over all test datasets and 5 runs.

		SGCS (↑)				**InfoNCE Loss** (↓)			
Variant	**Train**	T=2	T=5	T=10	T=20	T=2	T=5	T=10	T=20
CPC-before -Comp	NOKIA	0.704	0.727	0.743	0.778	0.566	0.654	0.938	1.631
	OPPO	0.834	0.868	0.801	0.891	0.293	0.198	0.421	0.839
	CATT	0.915	0.916	0.943	0.839	0.514	0.477	0.662	1.568
	Mixed	0.873	0.886	0.817	0.842	0.182	0.203	0.975	1.959
CPC-after -Comp	NOKIA	0.677	0.677	0.676	0.675	0.023	0.025	0.019	0.017
	OPPO	0.680	0.680	0.679	0.675	0.028	0.024	0.022	0.021
	CATT	0.660	0.661	0.664	0.655	0.017	0.018	0.018	0.018
	Mixed	0.694	0.692	0.698	0.693	0.015	0.013	0.016	0.015

performance, confirming that the compression bottleneck is the dominant limiting factor rather than the temporal architecture. For CPC-before-Compression, the Transformer achieves marginally higher SGCS, though the gain over GRU and LSTM is negligible for most training datasets. The exception is the NOKIA training dataset, where GRU exhibits higher variance and lower mean SGCS compared to LSTM and Transformer, indicating sensitivity to backbone choice under domain-specific data characteristics. Overall, GRU provides a favorable trade-off between complexity and performance.

4.3.2.2 Effect of Prediction Horizon

Table 4.5 evaluates the effect of prediction horizon T on SGCS and InfoNCE loss. For *CPC-after-Compression*, SGCS remains nearly constant across all horizons (e.g., NOKIA: 0.677, 0.677, 0.676, 0.675), demonstrating that the model sustains reconstruction quality regardless of how far ahead it predicts. For *CPC-before-Compression*, most training datasets exhibit stable or slightly varying SGCS, with the exception of NOKIA, which improves with longer horizons. This is attributed to NOKIA's

Table 4.6: Effect of compressed representation size on SGCS (↑) / InfoNCE (↓) averaged over all test datasets and 5 runs. **Bold** denotes the best SGCS and best InfoNCE among all models for each compression size and training dataset.

Model	Train	Size=32	Size=64	Size=128
3GPP Baseline	NOKIA	0.676	0.704	0.725
	OPPO	0.673	0.701	0.730
	CATT	0.653	0.712	0.758
	Mixed	0.691	0.720	0.779
CPC-before-Comp	NOKIA	**0.727** / **0.654**	**0.734** / 0.756	**0.749** / 0.617
	OPPO	**0.868** / **0.198**	**0.833** / 0.319	**0.794** / 0.355
	CATT	**0.916** / **0.477**	**0.937** / 0.620	**0.945** / 0.622
	Mixed	**0.886** / 0.203	**0.860** / 0.422	0.770 / **0.154**
CPC-after-Comp	NOKIA	0.677 / **0.009**	0.702 / 0.017	0.743 / 0.021
	OPPO	0.680 / **0.008**	0.708 / 0.020	0.736 / 0.021
	CATT	0.661 / **0.014**	0.724 / 0.021	0.771 / 0.021
	Mixed	0.692 / **0.005**	0.740 / 0.015	**0.779** / 0.017

stronger temporal autocorrelation structure, where additional prediction steps provide a richer contrastive training signal, enabling the GRU to learn more robust temporal representations. Notably, InfoNCE loss increases with T for *CPC-before-Compression* (e.g., NOKIA: 0.566 at $T = 2$ to 1.631 at $T = 20$), reflecting a more challenging contrastive objective as the prediction horizon extends, while remaining near-zero for *CPC-after-Compression* across all horizons.

4.3.2.3 Effect of Reconstruction loss

Fig. 4.4b compares the 1-SGCS and MSE reconstruction losses across both model variants. The 1-SGCS loss consistently outperforms MSE, as it directly optimizes the evaluation metric. The performance gap is particularly pronounced in CPC-before-Compression, where MSE training leads to significant degradation for Mixed, NOKIA, and OPPO training datasets. These results confirm that aligning the training objective with the evaluation metric is critical for effective CSI compression.

Table 4.7: Effect of GRU hidden dimension on model complexity and SGCS/InfoNCE performance for CPC-before-Compression. Results averaged over all test datasets and 5 runs. **Bold** denotes the best SGCS (↑) and best InfoNCE (↓) per training dataset.

	Hidden=32	**Hidden=64**	**Hidden=128**
Encoder Params	663,072	2,648,608	10,587,168
Encoder GFLOPs (per batch)	0.65	2.62	10.47
Enc. Time (ms/batch)	1.978	2.0337	2.0984
Dec. Time (ms/batch)	0.669	0.6801	0.6799
SGCS ↑ / *InfoNCE* ↓			
NOKIA	0.809 / 0.818	**0.823** / 1.423	0.727 / **0.654**
OPPO	0.826 / 0.893	0.799 / 1.009	**0.868** / **0.198**
CATT	0.783 / 2.240	0.903 / 0.764	**0.916** / **0.477**
Mixed	0.785 / 1.251	0.876 / 0.327	**0.886** / **0.203**

4.3.2.4 Effect of Bottleneck Size

Table 4.6 reports SGCS and InfoNCE across three compressed representation sizes. The 3GPP baseline and CPC-after-Compression improve consistently with larger sizes in SGCS, while their InfoNCE remains near zero across all sizes, confirming that compression is the dominant bottleneck rather than the temporal objective. CPC-before-Compression outperforms both variants at every size, though it exhibits instability at Size=128 for certain training datasets (e.g., 0.770 for Mixed), suggesting Size=32 offers a better complexity-performance trade-off for the contrastive pre-compression approach.

4.3.2.5 Effect of Temporal Hidden Dimension

Table 4.7 examines the effect of the GRU hidden dimension on model complexity and performance. Larger hidden sizes improve both SGCS and InfoNCE across most

Table 4.8: Effect of decoder capacity on SGCS (↑) / InfoNCE (↓) for CPC-after-Compression variants. Results averaged over 5 runs.

Model	**Train**	**Nokia**	**Oppo**	**CATT**	**Mixed**
Decoder: 1,810,754 params	*Enc. 2.4457 ms / Dec. 2.9415 ms per batch*			*Dec. GFLOPs: 1.21*	
CPC-after-Comp	NOKIA	0.730 / 0.008	0.735 / 0.008	0.566 / 0.074	0.677 / 0.009
	OPPO	0.733 / 0.008	0.739 / 0.008	0.568 / 0.068	0.680 / 0.009
	CATT	0.648 / 0.014	0.656 / 0.014	0.674 / 0.035	0.664 / 0.010
	Mixed	0.720 / 0.005	0.728 / 0.005	0.627 / 0.037	0.692 / 0.005
Decoder: 4,057,026 params	*Enc. 2.4648 ms / Dec. 3.6716 ms per batch*			*Dec. GFLOPs: 3.45*	
CPC-after-Comp-Ver2	NOKIA	0.734 / 0.036	0.739 / 0.036	0.568 / 0.056	0.680 / 0.034
	OPPO	0.734 / 0.024	0.739 / 0.024	0.569 / 0.059	0.680 / 0.024
	CATT	0.629 / 0.012	0.635 / 0.013	0.670 / 0.018	0.648 / 0.010
	Mixed	0.729 / 0.015	0.737 / 0.014	0.622 / 0.044	0.696 / 0.013

datasets, as wider representations enable richer temporal context modeling. Notably, Hidden=128 achieves the best SGCS for OPPO, CATT, and Mixed, while Hidden=64 provides competitive results at a significantly reduced encoder footprint (2,649M vs. 10,587M parameters). Despite the substantial increase in encoder GFLOPs (0.65, 2.62, and 10.47 GFLOPs per batch for Hidden=32, 64, and 128 respectively), per-batch inference latency remains nearly constant (1.978–2.098 ms), as fixed GPU kernel launch overhead dominates short-running kernels [139], further compounded by on-the-fly GRU weight quantization during each forward pass. Overall, Hidden=64 offers a favorable trade-off, achieving over 75% parameter reduction relative to Hidden=128 with only marginal SGCS degradation.

4.3.2.6 Effect of Decoder Capacity

Table 4.8 compares CPC-after-Compression with an enhanced variant (Ver2) that introduces additional feature extraction layers before the CPC module at the BS.

Since the BS has no strict resource constraints, a larger decoder is practically feasible. Ver2 increases decoder parameters from 1.81M to 4.06M and Dec. GFLOPs from 1.21 to 3.45 GFLOPs per batch, while decoder inference time increases marginally from 2.94 to 3.67 ms per batch. Despite this additional capacity, SGCS improvements are marginal (e.g., NOKIA-trained: 0.730 to 0.734 on Nokia test), and InfoNCE loss increases slightly across most configurations. This suggests that the performance ceiling of CPC-after-Compression is not limited by decoder capacity, but rather by the information loss introduced during the compression-decompression cycle prior to temporal modeling.

4.4 Conclusion

This work builds upon the CSI compression model considered in 3GPP and develops two AI-enhanced architectures that integrate contrastive predictive coding to address the problem of channel aging [23–25] in 3GPP-compliant systems. Through evaluations on 3GPP-compliant datasets from Nokia, Oppo, and CATT, we showed that applying CPC before compression leads to consistently higher SGCS scores, often exceeding 0.90, and 32× lower decoder GFLOPs compared to the 3GPP baseline.

Ablation studies confirmed that the 1-SGCS loss, GRU backbone at hidden size 64, and compression level of 64 bits offer the best complexity-performance trade-offs. Notably, both variants maintain stable reconstruction quality across prediction horizons of 2 to 20 steps, enabling the BS to obtain future CSI estimates at no additional reconstruction cost. Additionally, pruning experiments revealed a favorable trade-off between model size and predictive performance [141]. Future directions include exploring lightweight designs with compressed intermediate representations,

and adaptive fine-tuning across channel variations.

Chapter 5

Conclusions and Future Directions

5.1 Concluding Remarks

This thesis addressed two interconnected challenges in wireless sensing and communication-aware AI deployment.

First, it investigated domain adaptation for Wi-Fi sensing under both single-user and multi-user settings. While prior work has shown promising results in controlled single-user scenarios, practical deployment requires robustness to domain shifts and, more importantly, the ability to handle the complexity of multi-user environments. In this context, we introduced *MU-SHOT-Fi*, a framework designed for unsupervised domain adaptation in multi-user Wi-Fi sensing. The proposed method addressed three core challenges that are largely absent in single-user settings: variable occupancy, severe class imbalance, and signal entanglement across users. To mitigate these issues, MU-SHOT-Fi incorporated an occupancy-weighted information maximization loss that reduces collapse toward dominant classes by downweighting empty slots during diversity regularization. This enables adaptation across varying user counts without requiring explicit occupancy estimation. In parallel, we introduced *SU-SHOT-Fi* for

the simpler single-user scenario, which served both as a baseline and as a comparative setting to highlight the additional difficulties introduced by multi-user sensing. Our results further showed that CPC is beneficial in single-user adaptation, but becomes less effective in multi-user settings due to the conflict between sample-level temporal consistency and permutation-invariant slot prediction.

Second, this thesis examined AI-enhanced deployment under 3GPP-inspired communication constraints, with particular attention to channel aging and model partitioning. Through the study of two AI-based architectures, we showed that deployment performance is shaped not only by model accuracy, but also by how computation is distributed between the UE and BS. The results revealed a practical trade-off between model complexity and deployment feasibility, highlighting two distinct design choices: pushing a larger portion of computation to the UE, or retaining more of the model at the BS depending on latency, resource, and channel constraints.

Taken together, these contributions show that robust wireless intelligence depends not only on improving prediction accuracy in isolated settings, but also on designing methods that remain reliable under domain shifts, multi-user interference, and realistic deployment limitations. Despite these contributions, several important challenges remain open that are listed in the following.

5.2 Limitations in Existing Datasets

A central limitation in current Wi-Fi sensing research lies in the datasets used to develop and evaluate models. Existing CSI-based datasets for human activity recognition have enabled important progress, but they remain limited in ways that directly affect the goals of this thesis [11,51,79–81]. In particular, many datasets are collected

from a small number of participants in homogeneous and highly controlled environments [16, 85, 87, 142]. As a result, they fail to capture the diversity in behavior, body characteristics, environments, and hardware conditions needed for models to generalize reliably across domains.

This limitation is especially critical for domain adaptation. Since Wi-Fi sensing signals are highly sensitive to changes in environment, demographics, device placement, and hardware configuration, large domain gaps can easily destabilize adaptation and reduce discriminative performance. In realistic settings, models may become overconfident, collapse toward dominant classes, or fail to separate activities under strong distribution shifts. These issues are even more pronounced in multi-user scenarios, where dataset limitations are compounded by occupancy variation and signal overlap. A further issue is that most available datasets are biased toward controlled indoor conditions [15, 84, 86]. Such datasets do not adequately reflect the variability of practical deployment environments, where noise, layout changes, and dynamic reflectors can substantially alter CSI measurements [89, 90]. Consequently, progress in Wi-Fi sensing is increasingly constrained not only by model design, but also by the lack of diverse, realistic, and standardized datasets that support robust evaluation under real-world conditions.

5.3 Open Challenges in Domain Adaptation

The results of this thesis also highlight several open challenges in domain adaptation for Wi-Fi sensing. Although the proposed framework advances adaptation in multi-user settings, this problem remains underexplored. Most existing domain adaptation

methods are developed for single-user scenarios and do not directly address the structural difficulties introduced by multiple simultaneously active users.

One persistent challenge is the dependence on data preprocessing. Many prior approaches in Wi-Fi sensing rely heavily on carefully engineered preprocessing pipelines to reduce CSI noise and stabilize the input representation [44, 52, 118]. While effective in controlled experiments, this dependence can create a mismatch between training and deployment, especially when the same preprocessing assumptions cannot be guaranteed in practice. Future adaptation methods should therefore aim to be more resilient to raw-signal variability and less dependent on tightly controlled preprocessing steps.

Another challenge is the unresolved tension between occupancy estimation and activity recognition in multi-user adaptation. In supervised multi-user methods, strategies such as permutation-invariant learning or query-based prediction can exploit label information during training. In unsupervised adaptation, however, target-domain labels are unavailable. Although MU-SHOT-Fi addresses this difficulty through occupancy-weighted regularization and uncertainty-aware adaptation, the trade-off remains only partially resolved. Future work should focus on learning representations that better separate user occupancy from user activity while maintaining stability under label-free adaptation.

5.4 Future Directions

A promising direction for future research is multi-modal learning. Combining Wi-Fi sensing with complementary modalities can improve robustness and decision-making by leveraging different sources of information. Prior work has shown that integrating

Wi-Fi sensing with visual data can enhance human activity recognition [143], while combining CSI with inertial measurements can improve robustness beyond unimodal systems [144]. These results suggest that multi-modal fusion may help address some of the ambiguity and instability that remain challenging in wireless sensing alone.

However, this direction must be approached carefully. Multi-modal methods often introduce heavier architectures, higher latency, and greater energy consumption, which can conflict with the practical deployment constraints emphasized in this thesis [145]. In addition, many fusion methods assume synchronized modalities, balanced sample availability, and strong information overlap, assumptions that may not hold in real deployments.

Most importantly, certain modality choices may undermine the original motivation for Wi-Fi sensing. For example, integrating CSI with vision can compromise privacy and increase computational cost, weakening two of the key advantages of Wi-Fi-based sensing [146]. For this reason, future work should prioritize privacy-preserving and lightweight modality combinations, together with efficient fusion strategies that remain compatible with edge and communication-constrained deployment.

Overall, this thesis demonstrated that advancing wireless intelligence requires jointly addressing learning robustness and deployment realism. On the sensing side, Chapter 3 showed that domain adaptation for Wi-Fi sensing becomes fundamentally harder in multi-user settings due to variable occupancy, class imbalance, and signal entanglement, motivating new adaptation objectives beyond those used in single-user scenarios. On the communication side, Chapter 4 showed that channel aging cannot be effectively addressed when CSI compression and CSI prediction are treated as

separate problems. Instead, it demonstrated that integrating temporal representation learning within a 3GPP-compliant compression pipeline provides a unified and age-aware CSI feedback framework, while also revealing practical trade-offs between predictive performance, feedback overhead, and UE/BS computational complexity. Taken together, these findings suggest that future wireless AI systems should be developed as end-to-end adaptive systems that jointly account for sensing uncertainty, temporal variation, and deployment constraints.

Bibliography

[1] N. Hasanzadeh, R. Djogo, H. Salehinejad, and S. Valaee, "Hand movement velocity estimation from WiFi channel state information," in *2023 IEEE 9th Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2023, pp. 296–300.

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

[3] F. Adib and D. Katabi, "See through walls with wifi!" in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 75–86.

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

[5] 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 Conf. (RadarConf21)*. IEEE, 2021, pp. 1–5.

[6] 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 Intl. Conf. on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[7] 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 Jrnl.*, vol. 21, no. 1, pp. 548–559, 2020.

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

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

[10] 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 Jrnl.*, vol. 9, no. 24, pp. 24 643–24 654, 2022.

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

[12] S. Tan and J. Yang, "WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. of the 17th ACM Intl. 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

[13] 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 Jrnl.*, vol. 8, no. 7, pp. 5619–5631, 2021.

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

[15] 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, mar 2018. [Online]. Available: https://doi.org/10.1145/3191755

[16] 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 Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8671–8688, 2022.

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

[18] 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," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–26, 2019.

[19] 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 Intl. Conf. on Control & Automation (ICCA)*, 2022, pp. 510–515.

[20] J. Liu, Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng, "Tracking vital signs during sleep leveraging off-the-shelf WiFi," in *Proc. of the 16th ACM Intl. 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

[21] X. Wang, C. Yang, and S. Mao, "PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices," in *2017 IEEE 37th Intl. Conf. on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 1230–1239.

[22] I. Shirakami and T. Sato, "Heart rate variability extraction using commodity Wi-Fi devices via time domain signal processing," in *2021 IEEE EMBS Intl. Conf. on Biomedical and Health Informatics (BHI)*. IEEE, 2021, pp. 1–4.

[23] M. Moradian, A. Dadlani, A. Khonsari, and H. Tabassum, "Age-aware dynamic frame slotted aloha for machine-type communications," *IEEE Transactions on Communications*, vol. 72, no. 5, pp. 2639–2654, 2024.

[24] M. S. Kumar, A. Dadlani, and H. Tabassum, "Age of information in unreliable tandem queues," *IEEE Communications Letters*, 2025.

[25] A. A. Al-Habob, H. Tabassum, and O. Waqar, "Non-orthogonal age-optimal information dissemination in vehicular networks: A meta multi-objective reinforcement learning approach," *IEEE Transactions on Mobile Computing*,

vol. 23, no. 10, pp. 9789–9803, 2024.

[26] S. Banerji and R. S. Chowdhury, "On ieee 802.11: wireless lan technology," *arXiv preprint arXiv:1307.2661*, 2013.

[27] 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," *IEEE Commun. Surveys & Tutorials*, 2024.

[28] IEEE Standards Association, "The evolution of Wi-Fi technology and standards," https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/, 2023, accessed: 2024-08-23.

[29] L. Galati-Giordano, G. Geraci, M. Carrascosa, and B. Bellalta, "What will Wi-Fi 8 be? a primer on IEEE 802.11 bn ultra high reliability," *IEEE Commun. Magazine*, vol. 62, no. 8, pp. 126–132, 2024.

[30] 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 Wi-Fi sensing," *IEEE Communications Standards Magazine*, vol. 8, no. 1, pp. 58–64, 2024.

[31] N. Keshtiarast, P. K. Bishoyi, I. M. Lumbantobing, and M. Petrova, "When next-gen sensing meets legacy wi-fi: Performance analyses of ieee 802.11 bf and ieee 802.11 ax coexistence," *arXiv preprint arXiv:2503.04637*, 2025.

[32] X. Liu, T. Chen, Y. Dong, Z. Mao, M. Gan, X. Yang, and J. Lu, "Wi-Fi 8: Embracing the millimeter-wave era," *IEEE Communications Magazine*, 2024.

[33] M. Hatami, Q. Qu, Y. Chen, H. Kholidy, E. Blasch, and E. Ardiles-Cruz, "A survey of the real-time metaverse: Challenges and opportunities," *Future Internet*, vol. 16, no. 10, p. 379, 2024.

[34] B. Barahimi, H. Tabassum, M. Omer, and O. Waqar, "Context-aware predictive coding: A representation learning framework for wifi sensing," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 6119–6134, 2024.

[35] 3GPP, "Technical Report (TR) 38.843: Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface," 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network (TSG RAN), 3GPP Technical Report TR 38.843, Sep. 2025, release 19, Version 19.0.0.

[36] X. Mootoo, H. Tabassum, and L. Chiaraviglio, "Emforecaster: A deep learning framework for time series forecasting in wireless networks with distribution-free uncertainty quantification," *IEEE Transactions on Network Science and Engineering*, 2025.

[37] 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, jan 2011. [Online]. Available: https://doi.org/10.1145/1925861.1925870

[38] Z. Yan, Y. Huang, J. Pei, H. Tabassum, and L. Chiaraviglio, "Emfusion: Conditional diffusion framework for trustworthy frequency selective emf forecasting in wireless networks," *arXiv preprint arXiv:2512.15067*, 2025.

[39] N. Hasanzadeh and S. Valaee, "Enhancing generalization in human activity recognition through improved wi-fi channel state information phase processing and antenna pair selection," in *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2024, pp. 1–6.

[40] A. Y. Radwan, M. Yildirim, N. Hasanzadeh, H. Tabassum, and S. Valaee, "A tutorial-cum-survey on self-supervised learning for wi-fi sensing: Trends, challenges, and outlook," *IEEE Commun. Surveys & Tut.*, 2025.

[41] Z. Wei, W. Chen, S. Ning, W. Lin, N. Li, B. Lian, X. Sun, and J. Zhao, "A survey on wifi-based human identification: Scenarios, challenges, and current solutions," *ACM Trans. on Sensor Networks*, vol. 21, no. 1, pp. 1–32, 2025.

[42] M. Cominelli, F. Gringoli, and F. Restuccia, "Exposing the csi: A systematic investigation of csi-based wi-fi sensing capabilities and limitations," in *2023 IEEE International Conf. on Pervasive Computing and Commun. (PerCom)*. IEEE, 2023, pp. 81–90.

[43] J. Strohmayer and M. Kampel, "Data augmentation techniques for cross-domain wifi csi-based human activity recognition," in *IFIP International Conf. on Artificial Intelligence Applications and Innovations*. Springer, 2024, pp. 42–56.

[44] C. Shi, J. Liu, N. Borodinov, B. Leao, and Y. Chen, "Towards environment-independent behavior-based user authentication using wifi," in *2020 IEEE 17th International Conf. on Mobile Ad Hoc and Sensor Sys. (MASS)*. IEEE, 2020, pp. 666–674.

[45] 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.

[46] Y. Liang, W. Wu, H. Li, F. Han, Z. Liu, P. Xu, X. Lian, and X. Chen, "Wiai-id: Wi-fi-based domain adaptation for appearance-independent passive person identification," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 1012–1027, 2023.

[47] J. Jiao, X. Wang, and C. Han, "Robust indoor localization in dynamic environments: A multi-source unsupervised domain adaptation framework," *arXiv preprint arXiv:2502.07246*, 2025.

[48] Y. Fang, P.-T. Yap, W. Lin, H. Zhu, and M. Liu, "Source-free unsupervised domain adaptation: A survey," *Neural Networks*, vol. 174, p. 106230, 2024.

[49] J. Li, Z. Yu, Z. Du, L. Zhu, and H. T. Shen, "A comprehensive survey on source-free domain adaptation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5743–5762, 2024.

[50] S. Tan, L. Zhang, Z. Wang, and J. Yang, "Multitrack: Multi-user tracking and activity recognition using commodity wifi," in *Proceedings of the 2019 CHI Conf. on Human Factors in Computing Sys.*, 2019, pp. 1–12.

[51] 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," in *European Conf. on Computer Vision*. Springer, 2024, pp. 72–91.

[52] H. Yan, X. Zhang, J. Huang, Y. Feng, M. Li, A. Wang, W. Ou, H. Wang, and Z. Liu, "Wi-sfdagr: Wifi-based cross-domain gesture recognition via source-free domain adaptation," *IEEE Internet of Things Journal*, 2025.

[53] Z. Liu, L. Zhang, and Z. Ding, "An efficient deep learning framework for low rate massive mimo csi reporting," *IEEE Transactions on Commun.*, vol. 68, no. 8, pp. 4761–4772, 2020.

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

[55] Y. Guo, W. Chen, F. Sun, J. Cheng, M. Matthaiou, and B. Ai, "Deep learning for CSI feedback: One-sided model and joint multi-module learning perspectives," *arXiv preprint arXiv:2405.05522*, 2024.

[56] H. Shao, H. Zhang, W. Zhang, and X. Zhang, "Quantized trainable compressed sensing for MIMO CSI feedback," *IEEE Trans. on Vehicular Technology*, 2024.

[57] Y. An, S. Lu, H. Cai, and Z. Ji, "A deep learning-based approach to lightweight CSI feedback," *Physical Commun.*, vol. 68, p. 102538, 2025.

[58] X. Zhang, J. Wang, Z. Lu, and H. Zhang, "Continuous online learning-based CSI feedback in massive MIMO sys." *IEEE Commun. Letters*, 2024.

[59] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5g wireless communications: A deep learning approach," *IEEE Trans. on network science and engineering*, vol. 7, no. 1, pp. 227–236, 2018.

[60] T. Zhou, X. Liu, Z. Xiang, H. Zhang, B. Ai, L. Liu, and X. Jing, "Transformer network based channel prediction for csi feedback enhancement in ai-native air interface," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 9, pp. 11 154–11 167, 2024.

[61] J. Chengyong, G. Jiajia, L. Xiangyi, J. Shi, and Z. Jun, "Ai for csi prediction in 5g-advanced and beyond," *China Commun.*, vol. 22, no. 11, pp. 1–16, 2025.

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conf. on computer vision*. Springer, 2020, pp. 213–229.

[63] A. Mohammadi and H. Tabassum, "Amar: Lightweight attention-based multi-user activity recognition from wi-fi csi," 2026. [Online]. Available: https://arxiv.org/abs/2605.20649

[64] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[65] 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 Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8671–8688, 2022.

[66] S. Kadambar, A. T. Abebe, A. Kumar, A. K. R. Chavva, and H.-J. Ji, "Deep learning based joint csi compression and prediction for beyond-5g sys." in *GLOBECOM 2023-2023 IEEE Global Commun. Conf.* IEEE, 2023, pp. 4792–4797.

[67] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[68] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[69] J. Yang, X. Chen, D. Wang, H. Zou, C. X. Lu, S. Sun, and L. Xie, "Deep learning and its applications to WiFi human sensing: A benchmark and a tutorial," no. arXiv:2207.07859, Jul 2022, arXiv:2207.07859 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2207.07859

[70] X. Wang, C. Yang, and S. Mao, "Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity wifi," *ACM Trans. on Intelligent Sys. and Technology (TIST)*, vol. 9, no. 1, pp. 1–27, 2017.

[71] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *Intl. Jrnl. of computer science*, vol. 1, no. 2, pp. 111–117, 2006.

[72] N. Hasanzadeh and S. Valaee, "Enhancing generalization in human activity recognition through improved Wi-Fi channel state information phase processing and antenna pair selection," in *2024 IEEE 34th Intl. Workshop on Machine Learning for Signal Processing (MLSP)*, 2024, pp. 1–6.

[73] V. S. Abhayawardhana and I. J. Wassell, "Common phase error correction for OFDM in wireless communication," in *Proc. of the 3rd Intl. Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP'02)*, November 2002.

[74] M. F. Wagdy and M. S. P. Lucas, "A phase-measurement offset compensation technique suitable for automation," *IEEE Trans. on Instrumentation and Measurement*, vol. IM-36, no. 3, pp. 721–724, Sept. 1987.

[75] D. Wu, Y. Zeng, F. Zhang, and D. Zhang, "Wifi csi-based device-free sensing: from fresnel zone model to csi-ratio model," *CCF Trans. on Pervasive Computing and Interaction*, vol. 4, no. 1, pp. 88–102, 2022.

[76] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. of the 21st Annual Intl. Conf. on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 65–76. [Online]. Available: https://doi.org/10.1145/2789168.2790093

[77] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[78] A. Dempster, D. F. Schmidt, and G. I. Webb, "Minirocket: A very fast (almost) deterministic transform for time series classification," in *Proc. of the 27th ACM SIGKDD Conf. on knowledge discovery & data mining*, 2021, pp. 248–257.

[79] 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.

[80] S. Liu, Y. Zhao, and B. Chen, "WiCount: A deep learning approach for crowd counting using WiFi signals," in *2017 IEEE Intl. Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE Intl. Conf. on Ubiquitous Computing and Commun. (ISPA/IUCC)*, 2017, pp. 967–974.

[81] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity Wi-Fi devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, jan 2018. [Online]. Available: https://doi.org/10.1145/3161183

[82] Z. Hao, Y. Duan, X. Dang, Y. Liu, and D. Zhang, "Wi-sl: Contactless fine-grained gesture recognition uses channel state information," *Sensors*, vol. 20, no. 14, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/14/4025

[83] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "CrossSense: Towards cross-site and large-scale WiFi sensing," in *Proc. of the 24th Annual Intl. Conf. on Mobile Computing and Networking*, ser. MobiCom '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 305–320. [Online]. Available: https://doi.org/10.1145/3241539.3241570

[84] 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.

[85] I. Galdino, J. C. H. Soto, E. Caballero, V. Ferreira, T. C. Ramos, C. Albuquerque, and D. C. Muchaluat-Saade, "ehealth CSI: A Wi-Fi CSI dataset of human activities," *IEEE Access*, vol. 11, pp. 71 003–71 012, 2023.

[86] F. Meneghello, N. D. Fabbro, D. Garlisi, I. Tinnirello, and M. Rossi, "A CSI dataset for Wireless human sensing on 80 mhz Wi-Fi channels," *IEEE Commun. Magazine*, vol. 61, no. 9, pp. 146–152, 2023.

[87] A. Zhuravchak, O. Kapshii, and E. Pournaras, "Human activity recognition based on Wi-Fi CSI data -a deep neural network approach," *Procedia Computer Science*, vol. 198, pp. 59–66, 2022, 12th Intl. Conf. on Emerging Ubiquitous Systems and Pervasive Networks / 11th Intl. Conf. on Current and Future Trends of Information and Communication Technologies in Healthcare. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921024509

[88] 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, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/19/8860

[89] 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 Intl. Conf. on Computer Communication and Networks (ICCCN)*, 2018, pp. 1–8.

[90] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "Zigfi: Harnessing channel state information for cross-technology communication," *IEEE/ACM Trans. on Networking*, vol. 28, no. 1, pp. 301–311, 2020.

[91] S. Fang, T. Islam, S. Munir, and S. Nirjon, "EyeFi: Fast human identification through vision and WiFi-based trajectory matching," in *2020 16th Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2020, pp. 59–68.

[92] F. Abuhoureyah, S. K. Swee, and W. Y. Chiew, "Multi-user human activity recognition through adaptive location-independent WiFi signal characteristics," *IEEE Access*, 2024.

[93] F. tu Zahra, Y. S. Bostanci, and M. Soyturk, "LSTM-based jamming detection and forecasting model using transport and application layer parameters in Wi-Fi based IoT systems," *IEEE Access*, 2024.

[94] X. Fu, C. Wang, and S. Li, "Wi-SensiNet: Through-wall human activity recognition based on WiFi sensing," in *2024 5th Intl. Conf. on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. IEEE, 2024, pp. 238–241.

[95] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[96] L.-H. Shen, A.-H. Hsiao, F.-Y. Chu, and K.-T. Feng, "Time-selective RNN for device-free multiroom human presence detection using WiFi CSI," *IEEE Trans. on Instrumentation and Measurement*, vol. 73, pp. 1–17, 2024.

[97] A. Kumar, S. Singh, V. Rawal, S. Garg, A. Agrawal, and S. Yadav, "CNN-based device-free health monitoring and prediction system using WiFi signals," *Intl. Jrnl. of Information Technology*, vol. 14, no. 7, pp. 3725–3737, 2022.

[98] A. Nadia, S. Lyazid, K. Okba, and C. Abdelghani, "A CNN-MLP deep model for sensor-based human activity recognition," in *2023 15th Intl. Conf. on Innovations in Information Technology (IIT)*. IEEE, 2023, pp. 121–126.

[99] Z. He, X. Zhang, Y. Wang, Y. Lin, G. Gui, and H. Gacanin, "A robust CSI-based Wi-Fi passive sensing method using attention mechanism deep learning," *IEEE Internet of Things Jrnl.*, vol. 10, no. 19, pp. 17 490–17 499, 2023.

[100] B. Barahimi, H. Singh, H. Tabassum, O. Waqar, and M. Omer, "Rscnet: Dynamic csi compression for cloud-based WiFi sensing," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 4179–4184.

[101] Q. Gao, Y. Hao, and Y. Liu, "Autosen: improving automatic WiFi human sensing through cross-modal autoencoder," in *ICASSP 2024-2024 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 8235–8239.

[102] J. Bian, "SwinFi: a CSI compression method based on swin transformer for Wi-Fi sensing," *arXiv preprint arXiv:2405.03957*, 2024.

[103] J. Yang, C. Yan *et al.*, "Sensefi: A library and benchmark on deep-learning-empowered wifi human sensing," *Patterns*, vol. 4, no. 2, 2023.

[104] Z. Koh, Y. Zhou, B. P. L. Lau, C. Yuen, B. Tuncer, and K. H. Chong, "Multiple-perspective clustering of passive Wi-Fi sensing trajectory data," *IEEE Trans. on Big Data*, vol. 8, no. 5, pp. 1312–1325, 2020.

[105] K. Xu, J. Wang, H. Zhu, and D. Zheng, "Self-supervised learning for WiFi CSI-based human activity recognition: A systematic study," 2023. [Online]. Available: https://arxiv.org/abs/2308.02412

[106] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[107] S. Liu, Y. Zhao, and B. Chen, "WiCount: A deep learning approach for crowd counting using WiFi signals," in *2017 IEEE Intl. Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE Intl. Conf. on Ubiquitous Computing and Commun. (ISPA/IUCC)*. IEEE, 2017, pp. 967–974.

[108] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[109] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[110] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Jrnl.*, vol. 6, no. 6, pp. 10 639–10 651, 2019.

[111] J. Ding and Y. Wang, "A WiFi-based smart home fall detection system using recurrent neural network," *IEEE Trans. on Consumer Electronics*, vol. 66, no. 4, pp. 308–317, 2020.

[112] S. Tan, Y. Ren, J. Yang, and Y. Chen, "Commodity WiFi sensing in ten years: Status, challenges, and opportunities," *IEEE Internet of Things Jrnl.*, vol. 9, no. 18, pp. 17 832–17 843, 2022.

[113] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[114] R. Zandi, K. Behzad, E. Motamedi, H. Salehinejad, and M. Siami, "Robofisense: Attention-based robotic arm activity recognition with WiFi sensing," *IEEE Jrnl. of Selected Topics in Signal Processing*, 2024.

[115] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI open*, vol. 3, pp. 111–132, 2022.

[116] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "xLSTM: Extended long short-term memory," *arXiv preprint arXiv:2405.04517*, 2024.

[117] W. Tang and T. M. Khoshgoftaar, "Noise identification with the k-means algorithm," in *16th IEEE Intl. Conf. on Tools with Artificial Intelligence*. IEEE, 2004, pp. 373–378.

[118] B.-B. Zhang, D. Zhang, Y. Li, Y. Hu, and Y. Chen, "Unsupervised domain adaptation for rf-based gesture recognition," *IEEE Internet of Things Journal*, vol. 10, no. 23, pp. 21 026–21 038, 2023.

[119] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conf. on machine learning*. PMLR, 2020, pp. 6028–6039.

[120] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8602–8617, 2021.

[121] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[122] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in neural information processing systems*, vol. 32, 2019.

[123] 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 Sys.*, vol. 33, pp. 21 271–21 284, 2020.

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

[125] F. Wang, T. Zhang, W. Xi, H. Ding, G. Wang, D. Zhang, Y. Cui, F. Liu, J. Han, J. Xu *et al.*, "A survey on wi-fi sensing generalizability: Taxonomy, techniques, datasets, and future research prospects," *IEEE Communications Surveys & Tutorials*, 2026.

[126] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing Sys.*, vol. 32, 2019.

[127] H. Rizk, A. Elmogy, M. Rihan, and H. Yamaguchi, "Multisensex: A sustainable solution for multi-human activity recognition and localization in smart environments," *AI*, vol. 6, no. 1, p. 6, 2025.

[128] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[129] 3GPP TSG RAN WG1, "Discussions on AI-CSI," 3GPP, Xiamen, China, Tech. Rep. R1-2308873, 2023, 3GPP TSG RAN WG1 #114-bis Meeting.

[130] V. Rizzello and W. Utschick, "Learning the CSI denoising and feedback without supervision," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Commun. (SPAWC)*. IEEE, 2021, pp. 16–20.

[131] T.-H. Huang, A. Malhotra, and S. Hamidi-Rad, "A deep learning method for joint compression and unsupervised denoising of CSI feedback," in *ICC 2023-IEEE International Conf. on Commun.* IEEE, 2023, pp. 4150–4156.

[132] J. Guo, T. Chen, S. Jin, G. Y. Li, X. Wang, and X. Hou, "Deep learning for joint channel estimation and feedback in massive mimo sys." *Digital Commun. and Networks*, vol. 10, no. 1, pp. 83–93, 2024.

[133] C. Sun, T. Cui, W. Zhang, Y. Bai, S. Wang, and H. Li, "On the combination of ai and wireless technologies: 3GPP standardization progress," in *2024 IEEE/CIC International Conf. on Commun. in China (ICCC Workshops)*. IEEE, 2024, pp. 523–528.

[134] C. Tan, D. Cai, F. Fang, Z. Ding, and P. Fan, "Federated unfolding learning for CSI feedback in distributed edge networks," *IEEE Trans. on Commun.*, 2024.

[135] A. Saini, J. H. Kim, A. A. Tehrani, Y. Xing, and W. Gerstacker, "Network-first separate training with raw dataset sharing: A training approach for AI/ML-driven CSI feedback," in *2024 IEEE International Conf. on Commun. Workshops (ICC Workshops)*. IEEE, 2024, pp. 1950–1955.

[136] 3GPP RAN4 Working Group, "NR AIML Air Interface CSI Compression Datasets (R4_113)," https://www.3gpp.org/ftp/tsg_ran/WG4_Radio/Data_sharing/NR_AIML_air/CSI_compression/Datasets/R4_113, 2025, accessed: 2025-08-06.

[137] 3GPP TSG-RAN WG4 Meeting #112-bis, "WF on requirements for AI/ML air interface," Qualcomm, Hefei, Anhui, China, Tech. Rep. R4-2417212, Oct 14th–Oct 18th 2024.

[138] 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.

[139] NVIDIA. (2025) Cuda c programming guide: Cuda graphs. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#cuda-graphs

[140] 3GPP TSG-RAN WG4 Meeting #113, "On AI/ML Based CSI Compression," Nokia, Orlando, US, Tech. Rep. R4-2419178, Nov 18th–Nov 22nd 2024, agenda item: 7.17.2.1, Document for: Discussion.

[141] A. Y. Radwan, M. Shehab, and M.-S. Alouini, "Tinyml nlp scheme for semantic wireless sentiment classification with privacy preservation," in *2025 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2025, pp. 133–138.

[142] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, no. 1, pp. 286–293, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16103

[143] 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 *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[144] W. Guo, S. Yamagishi, and L. Jing, "Human activity recognition via wi-fi and inertial sensors with machine learning," *IEEE Access*, vol. 12, pp. 18 821–18 836, 2024.

[145] Y. Zong, O. M. Aodha, and T. Hospedales, "Self-supervised multimodal learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.

[146] F. Zhou, G. Zhu, X. Li, H. Li, and Q. Shi, "Towards pervasive sensing: A multimodal approach via CSI and RGB image modalities fusion," in *Proceedings of the 3rd ACM MobiCom Workshop on Integrated Sensing and Communications Systems*, 2023, pp. 25–30.