Slim-Sense: A Resource Efficient WiFi Sensing Framework towards Integrated Sensing and Communication

VIJAY KUMAR SINGH, Indraprastha Institute of Information Technology Delhi, India
ARYAN WALECHA, Indraprastha Institute of Information Technology Delhi, India
ASHUTOSH GERA, Indraprastha Institute of Information Technology Delhi, India
RISHABH JAY, Indraprastha Institute of Information Technology Delhi, India
ARANI BHATTACHARYA, Indraprastha Institute of Information Technology Delhi, India
MUKULIKA MAITY, Indraprastha Institute of Information Technology Delhi, India

With the growing use cases of CSI-based WiFi sensing, future WiFi networks are moving towards integrating sensing and communication (ISAC) by sharing the same frequency resources between data communication and WiFi sensing. However, it is known that WiFi sensing is detrimental to WiFi communication due to its expensive use of frequency resources for collecting CSI samples, limiting its effectiveness in ISAC. To address this challenge, we propose Slim-Sense, a novel approach to resource saving while maximizing the sensing accuracy. We first demonstrate that it is possible to perform accurate WiFi sensing without using the entire bandwidth. In fact, we can obtain close to maximum accuracy while utilizing only 24.42% of the bandwidth and 25% of the antennas. Obtaining such accuracy at low bandwidth requires the selection of the antennas and bandwidth that are most relevant for sensing activities. One of Slim-Sense's highlights is using a novel approach consisting of a Sparse Group Regularizer (SGR) and Hierarchical Reinforcement learning (HRL) technique to select the minimum optimal bandwidth resources for sensing while maximizing sensing accuracy. Considering the stochastic nature of the sensing environment and the difference in requirements of different sensing applications, Slim-Sense provides an environment and application-specific bandwidth resources for sensing. We evaluate Slim-Sense with four different WiFi CSI datasets, each varying in sensing environment and application, including one we collected in 46 different environmental settings. The experimental evaluation shows that Slim-Sense saves up to 92.9% resources while incurring $< 5\%$ reduction in sensing accuracy compared to using entire spectrum resources. We show that Slim-Sense is generalized to different environments and sensing models. Compared to the state-of-art solution, Slim-Sense outperforms and achieves a maximum improvement of 28.75% in resource-saving and 42.18% in sensing accuracy.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; *Ubiquitous computing*; Mobile computing; Human Activity Recognition.

ACM Reference Format:
Vijay Kumar Singh, Aryan Walecha, Ashutosh Gera, Rishabh Jay, Arani Bhattacharya, and Mukulika Maity. 2025. Slim-Sense: A Resource Efficient WiFi Sensing Framework towards Integrated Sensing and Communication. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 1, Article 225 (March 2025), 33 pages. https://doi.org/10.1145/3712271

Authors' Contact Information: Vijay Kumar Singh, vijay@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India; Aryan Walecha, aryan21455@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India; Ashutosh Gera, ashutosh21026@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India; Rishabh Jay, rishabh22401@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India; Arani Bhattacharya, arani@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India; Mukulika Maity, mukulika@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, New Delhi, India.

ACM 2474-9567/2025/3-ART225
https://doi.org/10.1145/3712271

1 Introduction

In recent years, WiFi sensing using Channel State Information (CSI) is increasingly being used to develop device-free, non-intrusive, and pervasive Human Activity Recognition (HAR) systems. In WiFi sensing, RF signals are harnessed to capture and recognize different activities of a person in the indoor environment. With the growing use cases of WiFi sensing, future WiFi networks aim to support ISAC (Integrated Sensing and Communication) by sharing the same frequency spectrum resources between data communication and sensing. Enabling ISAC is one of the main objectives of the upcoming WiFi standard IEEE 802.11bf [1, 6, 29].

With the increasing importance of the co-existence of sensing and communication applications, a few prior works such as WiImg [37] have pointed out that the huge number of packets exchanged (200-2000 packets/sec) for sensing hampers the performance of communication. They further report that exchanging only about 200 packets/sec for sensing causes a 40% reduction in communication throughput. On the other hand, multiple works [12, 25, 27] have reported that reducing the sensing packet exchange frequency hampers sensing application performance.

Hence, to enable ISAC, a recent prior work *SenCom* [15] tries to utilize the in-band WiFi communication traffic by passively sniffing the communication packets to measure the CSI data. The paper proposes methods to transform the CSI data collected from communication frames to be used for WiFi sensing. On the same lines, *MUSE-Fi* [17] utilizes CSI from uplink/downlink traffic and beamforming reports for activity recognition in a multi-person environment. In the absence of communication traffic, these solutions switch to an active mode of CSI data collection, utilizing the entire bandwidth. *BeamSense* [35] tries to utilize beamforming reports measures using NDP packets made originally for the communication purpose for sensing. However, beamforming reports do not provide fine-grained information of the environment required for accurate sensing [39]. Moreover, collecting beamforming reports at high rates requires the entire bandwidth, which hampers communication. On other hand, WiImg [42] utilizes low-rate CSI samples to sense human activities. It generates artificial high-rate CSI samples from the data collected at a lower sampling rate by leveraging lightweight GAN models.

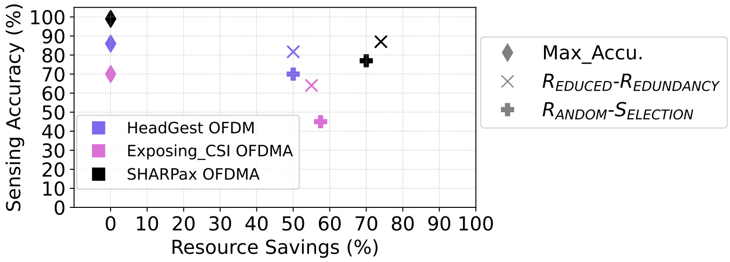


Fig. 1. Sensing Accuracy and Resource Saving for Exposing_CSI [8], SHARPax [28] and *HeadGest* datasets, with full bandwidth having maximum possible accuracy Max_Accu, Random-Selection, and Reduced-Redundancy.

While such efforts are much needed for the true co-existence of WiFi sensing and networking, in this paper, we take an orthogonal yet novel approach to enable smooth co-existence. We aim to minimize the utilization of spectrum resources (i.e., channel bandwidth and antennas) and computation needed (training time) for sensing purposes without impacting the accuracy of sensing applications. All prior works assume that it is important to use the complete spectrum of resources for performing sensing. Recent WiFi communication standards (WiFi 6 and WiFi 7) both use OFDMA (Orthogonal Frequency Division Multiple Access) and OFDM as channel access mechanisms. In case of OFDMA, the entire channel is broken into orthogonal sets of sub-carriers or RUs (Resource Units). Similarly, there could be multiple orthogonal sub-channels in the case of OFDM. In this paper, we first ask the question, *"Can we perform WiFi sensing by utilizing only a part of the bandwidth, i.e., by making it the slimmest possible?"* To that end, we performed a motivation experiment where we utilized two publicly available WiFi sensing datasets and collected one dataset of our own (*HeadGest*). We first sense using the entire bandwidth, followed by choosing the resource units randomly out of the entire channel, and finally, only select the resource units that are uncorrelated (additional details on experiments are in Section 5.3). Fig. 1 shows the accuracy of HAR versus the resource savings. We observe that we indeed can detect human activity utilizing only about $30-50\%$ of resources while incurring a small reduction of $4-14\%$ in sensing accuracy (for correlated selection). This forms the motivation for our work. On the other hand, random or correlated selection causes a significant reduction in sensing accuracy. Furthermore, to minimize the computation cost, we choose only relevant sub-carriers within an RU/sub-channel as was used by prior work [31]. Thus, we propose Slim-Sense to perform WiFi sensing in a way that maximizes sensing accuracy while minimizing resource usage and computation cost for sensing.

However, designing a generic system for diverse HAR applications using WiFi sensing comes with a number of challenges. A key aspect of HAR applications is that they have a variety of resource requirements. For example, more resources would be required to detect fine-grained activities such as finger movement or breath rate monitoring compared to coarse-grained activities such as walking or running detection. Thus, we need to decide *how much* resources would be needed for optimal sensing performance. The environment itself, i.e., the placement of the TX-RX pair, the presence of other obstacles, etc, affects the performance of HAR. Further, given the WiFi channel's stochastic and frequency-selective nature, it is not sufficient to specify only *how many* resources should be selected. It is also important to figure out *which* resources should be used. Thus, we need a systemic way to figure out *how much* and *which* resources should be allocated for sensing purposes to achieve optimal performance.

To solve this problem of selection of relevant resources, we provide a novel solution, Slim-Sense, that obtains the optimal trade-off between sensing accuracy and resource saving. We consider the available channel bandwidth and antennas as the entire spectrum resource. The selected resources must align with the sensing application and ensure maximum sensing accuracy. To achieve this, we integrate resource selection for sensing in the training procedure of the sensing model. We use a Sparse Group Regulariser (SGR), which deselects frequency resources to minimize the sensing loss during the training process. We integrate SGR with the sensing model. The selection (for sensing) and deselection (for communication) of resources, aimed at optimizing the trade-off between sensing accuracy and resource savings, are tuned by the SGR hyperparameters. Furthermore, resource selection is significantly influenced by the different antenna configurations at the sensing devices, especially when the number of antennas exceeds one. Thus, we model the antenna selection as a hyperparameter tuning task. In sensing scenarios, hyperparameter tuning is influenced by environmental dynamics. Since the environment is only partially observable, we formulate this problem as a POMDP (Partially Observable Markov Decision Process) problem. We solve the POMDP problem by designing a Hierarchical Reinforcement Learning (HRL) model to find out the optimal hyperparameters to obtain the best minimum resources (sub-channels/RUs and Antennas) for sensing without impacting its accuracy. The RL model has two agents, and they learn to make the best decision by sharing their knowledge with each other. The agents can not observe the true environmental state, instead they observe the environment with some uncertainty as it is only partially observable. Each agent maintains a belief state after interacting with the environment and shares their belief states with each other. One agent chooses the best possible antenna resources, and another one chooses only the minimum possible bandwidth resources controlled by the SGR hyperparameter for selected antennas.

We then evaluate Slim-Sense utilizing 4 different datasets: HeadGest, Exposing_CSI [8], SHARPax [28] and SimWiSense [14]. Each of the datasets varies in terms of various activities they capture, from course-grained movements such as walking and running to fine-grained hand movement and so on; the experimental setups vary in terms of the environment, noise, number of participants, interference levels, and so on. Slim-Sense adapts to these varying setups and provides up to 92.9% resource savings with only a minimal reduction in sensing accuracy (less than 5%). Due to the usage of Doppler vectors (computed from CSI samples) as the features, Slim-Sense generalizes to different environments. We demonstrate that the SHARP model, when using Doppler vectors as input features, achieves 44.57%, 45.54%, and 42.97% more sensing accuracy compared to a SHARP model with amplitude, phase, and the combination of amplitude and phase, respectively. Further, we show that Slim-Sense's framework is generic enough to be able to incorporate a different sensing model as well. We demonstrate the functionality of Slim-Sense's generalization framework through the usage of SimWiSense's few-shot embedding learning (FREL) model. Using SimWiSense's sensing model and datasets, Slim-Sense provides 84.56% sensing accuracy in a new and unseen environment with 50% resource saving. We compare Slim-Sense with a number of baselines, including a recent prior work WiImg, and show that Slim-Sense outperforms all the baselines and provides the best trade-off between sensing accuracy and resource savings. We believe Slim-Sense enables a smooth coexistence of sensing and communication by providing an optimal trade-off between sensing accuracy and resource-saving.

Key Contributions: We now summarize our contributions as follows:

(a) *Identifying the Scope of Slimming:* We first show the feasibility of WiFi sensing using only a part of the available spectrum resources with only a minimal impact on sensing accuracy. Specifically, we show that utilizing only 30 – 50% of the bandwidth used for WiFi communication is sufficient for sensing, with only a small degradation of 4 – 14% in sensing accuracy. Such an observation can help realize the aim of integrating sensing and communication by sharing the same spectrum of resources.

(b) *Ensuring Environment-Aware and Application-Specific Resources Selection:* We design Slim-Sense that applies a *Sparse Group Regularizer (SGR)* to select appropriate resources to be fed onto the state-of-the-art sensing model SHARP [28]. We design a *Heirarchical Reinforcement Learning (HRL)* model that helps choose the minimum “useful" spectrum resources. HRL model, during training, interacts with the environment through the SHARP model. During interaction (learning the environment), the SHARP model with SGR takes available input resources as features and selects the most optimal subset of features for sensing. The SHARP, with selected resources, obtains high sensing accuracy in new and unseen environments. The HRL model computes rewards from obtained sensing accuracy by penalizing it with the number of selected resources. Based on the obtained reward, the HRL model tunes the hyperparameters. On convergence, the HRL model selects minimal resources while providing maximum sensing accuracy in new and unseen environments. The saved resources are used for communication. Note that Slim-Sense takes Doppler vectors as input and provides an optimal subset of the Doppler vectors and, in turn, spectrum resources. The Doppler vectors offer an environment-independent representation of activities (details in Sections 2.1 and 3.1).

(c) *Extensive Evaluation of Slim-Sense:* We evaluate Slim-Sense on three publicly available datasets and our own HeadGest dataset. The datasets vary in participants, activities, locations, and environmental setups. We compare Slim-Sense with baseline approaches and WiImg [42]. In the case of our diverse HeadGest dataset, Slim-Sense achieves 50% of resource-saving with only a 4.3% reduction in sensing accuracy compared to Max_Accu (one where we utilize all the spectrum resources). In the case of diverse activities and IEEE 802.11ax, Exposing_CSI dataset, Slim-Sense achieves resource saving over 92% with only 3.5% reduction in sensing accuracy compared to Max_Accu. In less challenging SHARPax dataset, Slim-Sense shows a negligible reduction in sensing accuracy with up to 91% resource saving compared to Max_Accu. Compared with the existing solution WiImg, overall, Slim-Sense achieves up to 42.18 more sensing accuracy with up to 28.25% more resource savings. We further demonstrate Slim-Sense's generalizability with existing solutions by integrating it with an existing solution, SimWiSense, achieving 50% resource savings with only a 6.8% reduction in sensing accuracy compared to Max_Accu. In summary, through an extensive set of evaluations, we show that Slim-Sense achieves a good tradeoff between resource savings and sensing accuracy across varied environments and other sensing models. We have released the implementation of Slim-Sense and preprocessed dataset.[1].

2 Background

2.1 Channel State Information (CSI)

Channel State Information (CSI) is the property of the wireless channel that is measured by most WiFi devices, originally proposed for communication to implement advanced communication techniques such as beamforming, MIMO, spatial multiplexing, and channel bonding. These techniques require knowledge of the current channel conditions directly related to the properties of the WiFi channel. The CSI provides a very fine-grained knowledge of the current channel conditions. It involves detailed knowledge of how the signal interacts with its physical environment at a given time. Signal properties, such as strength, interference, and multipath effects, depend on channel conditions and change over time due to environmental changes. Any change in environment, such as human movement, alters the signal's multi-path propagation, which can be captured by the CSI. Hence, WiFi sensing systems primarily utilize CSI to develop a non-intrusive and passive human activity recognition system.

The new WiFi standard 802.11n/ac/ax and beyond has multiple-input multiple-output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) or Orthogonal Frequency Division Multiple Access (OFDMA) at the physical layer. The WiFi system with OFDM or OFDMA and MIMO computes CSI data between each pair of transmitter and receiver antennas for each subcarrier. In OFDMA/OFDM, the wide-band channel is divided into K orthogonal subcarriers or tones. The user information is transmitted over these K subcarriers in parallel. At the OFDMA/OFDM receiver, the channel parameters are continuously estimated for each subcarrier for each received packet $n \in \{1 \dots N\}$. The estimated channel parameters are collected as CSI, a large complex matrix that is environment-dependent and describes the channel frequency response of each subcarrier along every receiving antenna for each packet. In OFDM/OFDMA, we are given the R number of sub-channels/RUs. Considering multi-path propagation, P copies of the transmitted signal are collected at the receiver. Hence, CSI $H_k^{r,a,n}$ estimated for each packet n, each subcarrier $k \in \{1 \dots K\}$ of each sub-channel/RU $r \in \{1 \dots R\}$ and each antenna $a \in \mathbf{A}$(set of antennas) can be represented as:

$$H_k^{r,a,n} = \sum_{p=1}^{P} \eta_{k,p}^{r,n} e^{-j2\pi(f_c+k/T)\tau_p^n}, \tag{1}$$

where f_c is the central frequency, $T = 1/\Delta f$ (Δf is sub-carrier spacing) is OFDM/OFDMA symbol duration, $\eta_{k,p}^{r,n}$ is the attenuation factor and τ_p^n is the path delay associated with path $p \in \{1 \dots P\}$. These symbols are all summarized in Table 1. Since CSI data is noisy in nature and environment-dependent, it is often not directly used for sensing. Therefore, the WiFi sensing system extracts robust and environment-independent features from the raw CSI data. Slim-Sense primarily focuses on the Doppler phase shift vector [28], which is a robust and environment-independent feature.

2.2 Doppler Phase Shift Vector

The multipath components of a signal, as defined in Eq. (1), are obtained via scattering from static objects, such as walls and furniture, as well as from dynamic objects, such as moving humans. Hence, each multipath component has a different path length, delay, and attenuation. Different multipath signal components arrive at different times due to different path lengths, causing variations in path delay. The variations in the path delay introduce phase shift variations in the received signal. The phase shift $\phi_k^n = -2\pi(f_c + k/T)\tau_p^n$ is directly proportional to the main frequency f_c and path delay τ_p^n. The symbol duration T and subcarrier spacing Δf are constants. The main

[1] https://github.com/vijay13feb/SLIM-SENSE.git

Table 1. List of symbols and variables.

Symbol	Definition
$\mathbf{A}$	Set of antennas available in the AP
$a_i \in \mathbf{A}$	A single antenna
$r \in \{1 \dots R\}$	r is sub-channels/RUs index, and R is the total number of sub-channels/RUs
$k \in \{1 \dots K\}$	k is subcarrier index and K is total number of subcarriers
$n \in \{1 \dots N\}$	n is the packet index and N is the total number of packets
$H_k^{r,a,n}$	CSI values of each k of each r of each a of each packet n
$p \in \{1 \dots P\}$	p path/copy of a signal and P is total number of paths/copies of the signal
$\eta_{k,p}^{r,n}$	Attenuation associated with path p for k^{th} subcarrier of r^{th} RU and n^{th} packet
τ_p^n	Path delay associated with path p and packet n
f_c	Central frequency
Δf	Sub-carrier spacing
$T = 1/\Delta f$	Symbol duration
$\phi_k^{r,a,n}$	Undesired phase offset for k^{th} subcarrier of r^{th} RU of a^{th} antenna and n^{th} packet
$\mathcal{F}$	Short-time Fourier Transform
$i \in \{1 \dots O\}$	i is observation window index and O is total number of observation windows
$H_k^{r,a,n}$	CSI estimated for each packet n, each subcarrier k of each r and each antenna a
$H^{r,a}(i)$	CSI data matrix for r of a related to observation window i
$D_i^{r,a}$	The i^{th} Doppler vector of each sub-channel/RU r and each antenna a
W	Total number of packets in each observation window i
$v \in \{1 \dots V\}$	v is Doppler velocity index and V is the length of the Doppler vector
$d_{v,i}^{r,a}$	Doppler velocity in the Doppler vector
N_D^i	Horizontally stacked Doppler vectors of all r and then vertically stacked by antenna-wise of i^{th} observation window
f	Recognition system function
$C_1, \dots, C_m$	set of activities
$\mathbf{P}$ and $\hat{\mathbf{P}}$	Predicted probability distribution and ground truth probability distribution
$\hat{R}$ and $\hat{A}$	Relevant subset of sub-channels/RUs and antennas, respectively
$\lambda_1 \in \Lambda_1$	Lambda values of set Λ_1
$\lambda_g \in \Lambda_g$	Lambda values of set Λ_g
S, s, and $\hat{s}$	State space, current state, and next state
α	Action space
t	Transition probability
ρ	Reward function
Ω, ω	Observation space, observation probability
γ	Discount factor
Θ_1 and Θ_2	penalty weightage
G_1, G_2	Learning Agent
b^1, b^2	Belief state of G_1 and G_2, respectively
β	HRL learning rate

frequency f_c is different for each k. Thus, the phase shift of the same path delay varies across different subcarriers. The main frequency f_c of each subcarrier k remains fixed throughout the transmission. Hence, variations in phase shift only depend on the path delay τ_p^n. The path delay τ_p^n can be expressed as:

$$\tau_p^n = \frac{l_p^n + \Delta_p^n}{c}, \tag{2}$$

where c is the speed of light, l_p^n is the length of path p related to the initial position of the physical object, and Δ_p^n is the change in path length caused by movement of the physical object during the transmission period nT_c. The multipath components consist of static paths scattered from static physical objects and dynamic paths scattered from dynamic physical objects. The length of static paths remains the same throughout the transmission period nT_c, as the position of the static objects is fixed. Thus, $\Delta_p^n = 0$ and path delay $\tau_p^n = \frac{l_p^n}{c}$ of each static path of the multipath remains the same throughout the transmission duration. Therefore, the phase shift $\phi_k^n = -2\pi(f_c + k/T)\tau_p^n$ remains constant (f_c, T and τ_p^n remain constants) for static paths for each subcarrier. However, the length of dynamic paths changes over time since the position of dynamic objects such as humans changes. For example, when a dynamic object such as a human performs a specific activity in the indoor environment, the activity-related movements of a human induce variation in the path length over time as each body part acts as a scatterer moving at a specific velocity. The path length related to the moving scatterer changes during nT_c. Hence, the change in path length Δ_p^n for dynamic paths (whose l_p^n changes during nT_C) is defined as:

$$\Delta_p^n = -\int_0^{nT_c} v_p(x) \cos\theta_p(x)\, dx, \tag{3}$$

where $v_p(x)$ is the velocity of the scatterer related to dynamic path p, $\cos\theta_p(x)$ results from the combination of angles of the motion of the scatterer and angle of arrival/departure of the signal of p. The change in path length Δ_p^n varies with changes in velocity $v_p(x)$ and angle $\cos\theta_p(x)$ over transmission duration nT_c. Thus, according to Equation (2), τ_p^n changes as Δ_p^n varies. As a result, variations in τ_p^n lead to complex variations in phase shift $\phi_k^n = -2\pi(f_c + k/T)\tau_p^n$. The Doppler phase shift vector reveals the complex variations in phase shift caused by moving scatterer points.

The Doppler vector is computed from the $W \leq N$ subsequent estimated CSI samples through a short-time Fourier transform, i.e., during the channel observation window i. These W CSI samples are estimated at the WiFi monitor for W subsequent packets, collected with a sampling rate of T_c. The value of W is selected in such a way that the velocities and angles remain constant during $i \in \{1 \ldots O\}$, where O is the total number of observation windows. From Equation (3), this gives us:

$$\Delta_p^n = -v_p \cos\theta_p nT_c, \tag{4}$$

Thus, Doppler phase shift $v_p \cos\theta_p$ is estimated from Equations (1), (2) and (4). Let $H^{r,a}(i)$ denote the $K \times W$ dimensional matrix of r and a for i^{th} observation window, which reveal the human movements. It is defined as:

$$H^{r,a}(i) = \begin{bmatrix} H_1^{r,a,1}, \ldots, H_1^{r,a,W} \\ \vdots \\ H_K^{r,a,1}, \ldots, H_K^{r,a,W} \end{bmatrix} \tag{5}$$

where each value in $H^{r,a}(i)$ matrix, represent estimated CSI according to Equation (1) within the current observation window i. The short time Fourier transform $\mathcal{F}$ is applied on CSI matrix $H^{r,a}(i)$ to estimate the $V \times 1$ dimensional Doppler vector D_i as:

$$D_i^{r,a} = [d_{1,i}^{r,a}, \ldots, d_{V,i}^{r,a}]^T, \tag{6}$$

where each element in the $D_i^{r,a}$ is estimated as:

$$d_{v,i}^{r,a} = \sum_{k=1}^{K} |\mathcal{F}\{H^{r,a}(i)\}|^2, \tag{7}$$

where $v \in \{1 \dots V\}$ is Doppler velocity index. The expression $\mathcal{F}\{H^{r,a}(i)\}$ represents the Fourier transform applied column-wise (along subcarrier index k) to estimate the Doppler vector. The absolute values of squared Fourier transform coefficients are summed over the subcarrier axis k. The phase shift depends only on path variations Δ_p^n caused by human movements. The non-zero entries v in the Doppler vector reveal the presence of a moving scatterer with velocity $v_p(x)$ and angle $\cos\theta_p(x)$ as:

$$v_p \cos\theta_p = \frac{v_c}{f_c T_c V}, \tag{8}$$

where $v_p \cos\theta_p$ represents activity-related movements and reveals the dynamic components Δ_p^n (Equation (4)). Hence, the quantity in Equation (8) serves as a reliable indicator of dynamic components and is considered an effective feature for the activity recognition model. Human activities cause both small- and large-scale variations in the Doppler phase shift vectors due to differences in velocity and angle associated with each activity. The activity recognition model must detect and extract patterns at different scales to recognize different activities accurately.

2.3 SHARP Model

A state-of-the-art model used for activity recognition, such as walking, jumping, etc, is SHARP [28]. SHARP uses Doppler phase shift vectors as input to capture multi-scale variations of participant's activities. It consists of a sequence of max-pool and convolutional layers of different-sized kernels to identify the most important features of the Doppler vectors. Next, the most important features of the Doppler vectors map are converted to a single dimension by a flattened layer. Before passing the output features to *Dense* layer 20% of features are dropped to avoid overfitting. Finally, features are passed to *Dense* layer to assign the probabilities to each activity. This model has been validated to be highly accurate in settings such as homes, offices, and halls for 5-12 activities.

2.4 Related Works

We divide the related work into two categories: (1) WiFi sensing applications (2) ISAC (Integrated Sensing and Communication).

WiFi sensing applications: WiFi signals have been used by a number of prior works for tasks such as fall detection [5, 18, 21], motion detection [13, 24, 41], identification of breathing rate [2, 33], user and gesture recognition [3, 23, 32]. Works on fall and motion detection pose the problem as one of binary classification. User recognition and gesture recognition utilize multi-label classification, whereas identification of breathing rate utilizes different forms of regression. A number of recent works have also looked at additional challenges of WiFi sensing. For example, Muse-Fi [17] tackles the challenges of sensing in the context of multiple people by separately using downlink and uplink CSI and beamforming reports. RF-Net [9] utilizes extra features and a new neural network for meta-learning to generalize easily to different environments. OneFi [36], and WiLearner [10] adapt to unseen gestures using a self-attention mechanism and autoencoder, respectively. The work [28] proposes a convolutional neural network that generalizes to multiple types of activity recognition. These prior works show efficient activity recognition across different locations and participants. However, they utilize the entire available bandwidth for sensing, which is detrimental to WiFi communication, and utilizing the entire bandwidth for sensing may leave no or insufficient spectrum resources for effective communication.

ISAC (Integrated Sensing and Communication) A number of works also aim to integrate communication with the sensing capabilities of WiFi as shown in Table 2. For example, BeamSense [35] leverages existing WiFi

Table 2. Comparison of Slim-Sense with existing works.

Parameter	ISAC	Saving Channel Bandwidth	Saving Antennas	Adaptive Sampling Rate	Generalizability
BeamSense [35]	✓	✗	✗	✗	✓
SenCom [15]	✓	✗	✗	✓	✓
WiImg [42]	✓	✗	✗	✓	✗
Slim-Sense	✓	✓	✓	✗	✓

network devices to obtain *compressed beamforming reports (CBR)* by sniffing ongoing WiFi traffic and utilizing CBR reports (compressed version of CSI) already supported by WiFi devices to recognize human activities. Most of the 802.11 standard devices support *channel sounding protocol* to exchange CBR reports between transceivers. The two most closely related works are SenCom and WiImg.

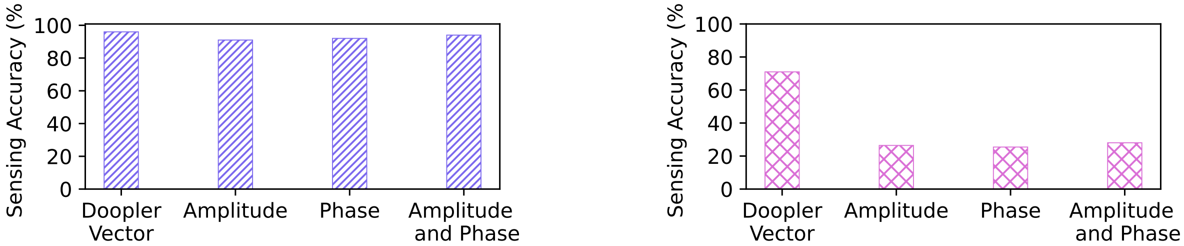
A recent work SenCom [15] overcomes several challenges of WiFi-based passive sensing, such as insufficient CSI collection due to intermittent and unevenly distributed traffic from AP to clients and impaired CSI data due to different modes (diversity or multiplexing) of communication. The key idea of SenCom is to enable WiFi sensing through CSI samples collected passively from MIMO downlink communication traffic. SenCom provides solutions for intermittent downlink traffic, which is unevenly distributed. If sufficient MIMO downlink traffic is not available, it moves to collect the CSI packets in active mode. In active mode, it utilizes all of the resources. WiImg [42] shows that the high frequency of sensing packets impacts the performance of communication. They point out that typical sensing applications have a communication rate of 200 − 2000 packets/sec. With such high sensing frequency exchanges, the authors claim that *WiFi sensing actually can not co-exist with WiFi communication*. The key idea of the other closely related work WiImg is to convert low-rate CSI samples into images and then apply a Generative Adversarial Network (GAN) for CSI images inpainting to generate images of high-rate CSI samples. It then utilizes high-rate CSI samples (synthetic samples) for activity recognition. In this way, WiImg minimizes the impact of sensing on WiFi communication by reducing the need for high packet rates. To achieve this, WiImg converts low-rate time-domain CSI samples into CSI amplitudes. Then, the low-rate CSI amplitudes are converted into images, simulating the structure of color images with three channels: red, green, and black. It uses 3 antennas to represent the channels, subcarriers to represent the image width, and sampling rate (for example, 100 packets/second) to represent the image length. Next, WiImg distributes the image length over the range of 0-250 (to the high sampling rate of 250 packets/second). The proposed GAN model then inpaints distributed images based on the original image. The authors show that with just 25 packets/sec, the sensing accuracy is improved by 31% compared to other interpolation works that try to interpolate the missing CSI values. Thus, both SenCom and WiImg's approach and resources saved differ significantly from that of Slim-Sense. Another work related to Slim-Sense is BeamSense[35], which utilizes beamforming reports as raw input data for sensing, unlike CSI-based WiFi sensing. However, unlike Slim-Sense, it is not possible to send packets for communication in parallel while sensing.

Our evaluation compares the performance of Slim-Sense with WiImg, as both of them save spectrum resources. We do not compare the performance of Slim-Sense with BeamSense, as BeamSense does not currently conserve spectrum resources for communication. Since there are no public datasets that capture CSI in passive mode, and the dataset of SenCom is not available, it is also not possible to compare Slim-Sense's performance with SenCom. However, SenCom's strategy of passive-mode sensing is orthogonal to that of Slim-Sense, and it is possible to integrate both the strategies to further save spectrum resources.

3 Motivation and Problem Formulation

3.1 Motivation

Future WiFi networks aim to enable ISAC (Integrated Sensing and Communication), where the sensing and communication nodes will co-exist. However, existing WiFi sensing solutions harm communication performance due to their excessive usage of spectrum resources. Therefore, we ask this question *"Is it always necessary to use all the spectrum resources while performing WiFi sensing?"* To answer this question, we perform a motivation experiment using the two existing datasets (Exposing_CSI and SHARPax) and our own dataset *HeadGest* (dataset details are in Section 5.1). First, we randomly choose a set of RUs/sub-channels from all the RUs/sub-channels. We extract the Doppler vectors from the CSI samples of the randomly selected RUs/sub-channels and train the appropriate deep-learning model used for that dataset. Fig. 1 shows a plot of sensing accuracy and resource savings for three different datasets. We observe that a random selection of resources named Random-Selection (details in Section 5.3) saves 50%, 57.5%, and 32.5% resources for *HeadGest*, Exposing_CSI and SHARPax datasets respectively. However, it incurs a 16.4%, 20.28%, and 22.22% reduction in sensing accuracy compared to the original *HeadGest*, Exposing_CSI, and SHARPax models, respectively. Next, we design a better algorithm where, instead of randomly choosing the RUs/sub-channels, we compute cross-correlation between a pair of RUs/sub-channels. We chose only uncorrelated RUs based on the correlation and performed the same experiment. We name this method as Reduced-Redundancy as it aims to reduce the redundancy of the resources (details in Section 5.3). Fig. 1 shows that Reduced-Redundancy provides better accuracy (improvement of 10.84%, 10.14%, and 8.08%) and more savings (improvement of 0%, 16.5%, 38.8%) compared to Random-Selection. This shows the promise of performing WiFi sensing without utilizing full spectrum resources. However, the accuracy obtained even with Reduced-Redundancy is lower compared to original *HeadGest*, Exposing_CSI, and SHARPax models utilizing the entire bandwidth. Thus, we aim to design a sensing framework to enable a smooth integration of WiFi sensing with communication that can (1) minimize the utilization of spectrum resources for sensing, (2) minimize the computation needed for sensing, and finally (3) maximize sensing accuracy. Moreover, the performance of the sensing framework depends on the input features computed from the CSI data. In this motivation experiment, we utilize the Doppler vectors as input features. The Doppler phase shift vector reveals the phase shift in the signal caused by human movements independent of the surrounding environment's configuration.



(a) Simple Scenario: Training and Testing in the same environment.

(b) Challenging Scenario: Training in the simple scenario and Testing in the challenging scenario.

Fig. 2. Sensing accuracy with different input features for Exposing_CSI dataset.

Doppler Vector as Input Feature: The raw CSI, as defined in Equation (1), is noisy in nature and affected by surrounding static objects, walls, and environmental settings. Thus, it is usually not used directly for activity recognition, as prior works show that it does not give high accuracy of sensing [7]. Early sensing solutions computed statistical features such as amplitude and phase variation from the collected raw CSI data. The amplitude for the CSI sample of packet n is the magnitude of the total complex sum of Equation (1) as:

$$A_k^{r,a,n} = \left|H_k^{r,a,n}\right|, \tag{9}$$

Moreover, the phase for the CSI sample of the packet n is the angle of the total complex sum of Equation (1) as:

$$P_k^{r,a,n} = \arg\left(H_k^{r,a,n}\right), \tag{10}$$

CSI is affected by the multi-path propagation of the indoor environment and hence also accounts for the reflections from static objects. The reflected signals, each with a different delay and phase shift, are superposed differently at each subcarrier, and the delay induces a subcarrier-specific phase shift. Such behavior is environment-specific. Hence, amplitude and phase, from Equation (9) and (10), can change due to various environmental settings, including static obstacles and multipath effects. They do not solely depend on human movements. Moreover, phase variation ($P_k^{r,a,n} - P_k^{r,a,n+1}$) depends on τ_p^n which is environment dependent. Therefore, amplitude and phase variation does not support the development of robust, environment-independent human activity recognition (HAR) systems. On the other hand, the Doppler vectors reveal the complex variations in the phase shift caused by only human activities. As per Equation (2), variations in phase shift depend on path length. The path length of signals reflected from humans performing activities changes over time as per Equation (3). However, the path length of the signals reflected from the static object remains the same throughout the transmission. The initial phase shift may change based on the object's characteristics (such as shape, position, and material); the variation in phase shift remains the same over time. Thus, the Doppler vector provides effective input features for environment-independent human activity recognition. We conducted an experimental study, as shown in Fig. 2, to demonstrate the robustness of Doppler vectors by training and testing the SHARP model using Exposing_CSI CSI data in two scenarios:

(1) Simple Scenario: The training and testing of the SHARP model are conducted in the Lab environment (details in Table 4), which remains the same during the experiment. The SHARP model with Doppler vectors, amplitude, phase, or the combination of both achieves similar sensing accuracy over 90%, as shown in Fig. 2a.
(2) Challenging Scenarios: The SHARP model with specific input features is trained in one environment (lab) and tested in a new or unseen environment (Office). The SHARP model, with the Doppler vectors as input features, achieves 44.57%, 45.54%, and 42.97% more sensing accuracy compared to the SHARP model with amplitude, phase, and combination of both, as shown in Fig. 2b. This indicates that using the Doppler vector as input features is more effective for environment-independent sensing than amplitude, phase, and combination of both.

3.2 Problem Formulation

We now formally describe the problem. We are given $|A|$ number of antennas and R number of sub-channels/RUs. The number of sub-channels/RUs is based on channel access techniques such as OFDM or OFDMA and available maximum bandwidth in 802.11n/ac/ax. Our aim is to choose the minimal set of resources while maximizing the sensing accuracy. We model the choice of resource selection as choosing the minimum set of input features computed from the raw CSI samples. We compute Doppler vectors, which provide effective input features for environment-independent sensing. Note that these Doppler vectors represent RU/sub-channels and antennas. Further, these Doppler vectors work as input features for the activity recognition model. *Hence, we compute the Doppler vector $D_i^{r,a}$ (details in Section 2.1) from CSI data collected at each a for each r separately.* We horizontally stack $D_i^{r,a}$ for all r of each antenna a. The sequence of $D_i^{r,a}$ ($r \in \{1 \ldots R\}$) represents the entire bandwidth. Next, to select the most optimal sub-channel/RUs across antennas, we vertically stack them antennas-wise to get matrix N_D^i as:

$$N_D^i = \begin{bmatrix} D_i^{1,1} & D_i^{2,1} & \dots D_i^{R,1} \\ D_i^{1,2} & D_i^{2,2} & \dots D_i^{R,2} \\ & \vdots & \\ D_i^{1,A} & D_i^{2,A} & \dots D_i^{R,A} \end{bmatrix}, \tag{11}$$

where $i \in \{1 \dots O\}$ is the observation window index and O is the total number of observation windows. The dimension of the matrix N_D^i is $|A| \times (|R \times V|)$. Each row in N_D^i represents the entire available bandwidth of each antenna a, and each column in N_D^i represents the Doppler vectors of specific r across all antennas. *We prepare the Doppler vector dataset ($[N_D^1, \dots, N_D^O]$) labeled with activities by stacking consecutive Doppler vector matrix N_D^i.* The user may perform one of several activities $C_1, \dots, C_m$. Let the probabilities of each activity given by the recognition system be denoted as $\mathbf{P} = [P(C_1), \dots, P(C_m)]$. We represent the recognition system by a function $f : [N_D^1, \dots, N_D^O] \rightarrow \mathbf{P}$. The recognition system f, trained with N_D^i, when $A = 1$ is independently applied to the Doppler vectors computed from each antenna at runtime. This makes the recognition system f generalized across devices with different numbers of antennas.

Let $P(\mathbf{A})$ be the power set of antennas. Let the most relevant subset of antennas, denoted by $\hat{\mathbf{A}} \in P(\mathbf{A})$, and the most relevant subset of sub-channels/RUs, denoted by $\hat{\mathbf{R}}$, be selected to sense the activities. The overall objective of Slim-Sense is to obtain a probability distribution $\mathbf{P}$ that is as close as possible to the ground truth $\hat{\mathbf{P}}$ while penalizing the selection of more antennas and more RUs. This is achieved by selecting the most relevant $\hat{\mathbf{R}}$ and $\hat{\mathbf{A}}$. We first quantify the error using the categorical cross-entropy between the distributions $\mathbf{P}$ and $\hat{\mathbf{P}}$:

$$Q_1 = \sum_{j=1}^{m} \hat{\mathbf{P}}(C_j) \log(\mathbf{P}(C_j)) \tag{12}$$

We now seek to quantify the amount of data used to compute Q_1. Let $X_v^{ra} = 1$ denote that $d_{v,i}^{r,a}$ (as per Equation (7)) is selected, and 0 otherwise. Then, the number of Doppler velocities used for the decision is quantified as:

$$Q_2 = \sum_{a \in \mathbf{A}} \sum_{r=1}^{R} \sum_{i=1}^{O} \sum_{v=1}^{V} X_v^{ra} |d_{v,i}^{r,a}| \tag{13}$$

Then, we quantify the number of RUs used across the antennas. Let y_{ra} be 1 if r^{th} RU of a^{th} antenna is selected and 0 otherwise. This gives us:

$$Q_3 = \sum_{a \in \hat{\mathbf{A}}} \sum_{i=1}^{O} \sum_{r=1}^{R} y_{ra} |D_i^{r,a}| \tag{14}$$

Hence, the overall objective Q is defined as:

$$\mathbf{Q} = \arg \min_{\substack{\hat{\mathbf{A}} \in P(\mathbf{A}) \\ \lambda_1 \in \Lambda_1 \\ \lambda_g \in \Lambda_g}} \left\{ \mathbf{Q}_1 + \lambda_1 \mathbf{Q}_2 + \lambda_g \mathbf{Q}_3 \right\}, \tag{15}$$

where $\lambda_1 \in \Lambda_1$ and $\lambda_g \in \Lambda_g$, as shown in Table 3, are the scalar hyperparameters that control the trade-off between minimizing loss function and deselecting irrelevant $d_{v,i}^{r,a}$ and $D_i^{r,a}$. *The optimal value of λ_1 and λ_g provides most relevant set of Doppler vectors denoted as $\hat{R}$.*

Table 3. Set of possible value of hyperparameters.

Hyperparameters	**Possible Values**
Λ_1	$\{0.0, \ldots, 1.0\}$
Λ_g	$\{0.0, \ldots, 1.0\}$
$\mathbf{P}(A)$ if $A = 4$	$\{\{0\}, \{1\}, \{2\}\{3\}, \{0,1\}, \{0,2\}, \{0,3\}, \{1,2\}, \{1,3\},$ $\{2,3\}, \{0,1,2\}, \{0,1,3\}, \{0,2,3\}, \{1,2,3\}, \{0,1,2,3\}\}$

4 Design and Solution Approach of Slim-Sense

We now discuss the design and solution approach of Slim-Sense as shown in Fig. 3. The input to Slim-Sense is Doppler vectors and ground truth labels of activities. The output will be a selected set of relevant resources, i.e., antennas and RUs. Note that all the notations are listed in Table 1. Next, we present the components of Slim-Sense. Finally, we describe the working of Slim-Sense.

4.1 Design of Slim-Sense

Equation (15) defines our joint optimization problem of minimizing loss function (Q_1), complexity of the neural network f (Q_2), and utilization of bandwidth resources (Q_3). The objective Q_2 provides sparsity across $d_{v,i}^{r,a}$ within a $D_i^{r,a}$, and where decision variable X_v^{ra} select $d_{v,i}^{r,a}$ that contribute to recognition activities. The objective Q_3 provides sparsity across $D_i^{r,a}$ within the entire bandwidth represented by N_D^i, and where decision variable y_{ra} select set of $D_i^{r,a}$ that contribute to recognition activities. We aim to control the decision variables to tune the degree of the sparsity applied at the $d_{v,i}^{r,a}$ and $D_i^{r,a}$ level. This ensures we obtain the minimum set of optimal Doppler vectors with optimal Doppler velocities for sensing with minimal impact on sensing accuracy. To achieve this, we integrate the tuning process within the training of f, which ensures feature selection aligned with the learning objectives. To implement this approach, we apply $L1$ regularizer to apply sparsity across $d_{v,i}^{r,a}$ and employ group regularizer to apply sparsity across $D_i^{r,a}$. $L1$ and group regularizer jointly form the Sparse Group Regularizer (SGR). f with SGR takes N_D^i (with a given combination of antennas) as input and provides $\hat{R}$ while optimizing Q. To obtain the optimal combination of antennas $\hat{A}$, f with SGR takes N_D^i as input with different antenna combinations $P(A)$. With each combination of antennas, f with SGR provides corresponding $\hat{R}$ while optimizing Q. The optimization of Q includes minimizing resource usage and loss function and is controlled by the hyperparameters are $\hat{\mathbf{A}}$, λ_1 and λ_g. Hence, it is essential to tune these hyperparameters to achieve optimal performance.

One straightforward solution is to perform an exhaustive search of these hyperparameters. The exhaustive search involves selecting hyperparameters, training the function f with each combination of hyperparameters, and observing the performance of f in one instance. The search process is sequential, selecting one set of hyperparameters at an instance. Consequently, it makes a series of iterations where hyperparameters are selected sequentially over time. Moreover, selecting hyperparameters from the set of hyperparameters is performed through random or grid search, and the selection of hyperparameters in each instance is independent of previous instances. *Thus, hyperparameter tuning is a sequential process that follows the Morkovain property, which resembles a Markov Decision Process (MDP).* Note that f is unknown, and we will only be able to observe it partially. The environment of different sizes and clutters, consisting of multiple reflecting surfaces, as well as placement of the monitoring devices, the access points, and the number of antennas at the receiving devices, affect the CSI [34], which reflects in the Doppler vector. Hence, f can sense only a portion of the environmental changes, as many of these changes cannot be directly observed. Considering the stochastic and partially observable environment, we model the problem of hyperparameter tuning as an instance of the Partially Observable Markov Decision Process (POMDP), a generalized form of MDP. We formally define the POMDP as a tuple $\boldsymbol{T} = (S, \alpha, t, \rho, \Omega, \omega, \gamma)$, consisting of state space (S), action space (α), transition probability (t), reward function (ρ) observations space (Ω), observation probability (ω) and discount factor (γ). The state space (S) consists of a set of states in which each state is represented as $s = ([N_D^1, \ldots, N_D^O], (\mathbf{P}(C_1), \ldots, \mathbf{P}(C_m)))$. The action space is a set of all possible values of hyperparameters defined as $\alpha = (P(\mathbf{A}), \Lambda_1, \Lambda_g)$. The observation space Ω is a set presenting the performance (defined in Equation (15)) of f with selected hyperparameters. The observation probability ω defines the uncertainty associated with observations. The discount factor γ is retained as a parameter and t is the transition probability. The reward function ρ defines the accuracy achieved by the function f to recognize the activities. Let $\hat{C}_i$ be the activity label recognized by the function f, and C_i be the ground truth activity labels. Thus, the reward function ρ is defined as:

$$\rho = \frac{1}{O}\sum_{i=1}^{O}\mathbf{1}(\hat{C}_i = C_i) - (\Theta_1(|\hat{R}|) + \Theta_2(|\hat{A}|)), \tag{16}$$

where $|\hat{R}|$ and $|\hat{A}|$ are the total number of selected sub-channels/RUs and antennas. The term $(\Theta_1(|\hat{R}|) + \Theta_2(|\hat{A}|))$ defines a penalty on higher usage of RUs and antennas, where Θ_1 and Θ_2 parameters are penalty weightages. The penalty term gives a tradeoff between accuracy and resource utilization to optimize the performance of f while conserving resources. We implement POMDP using Reinforcement Learning (RL), which provides adaptability to changes in the environment over time. Moreover, RL provides flexibility to model the uncertain environment, which is only partially observable.

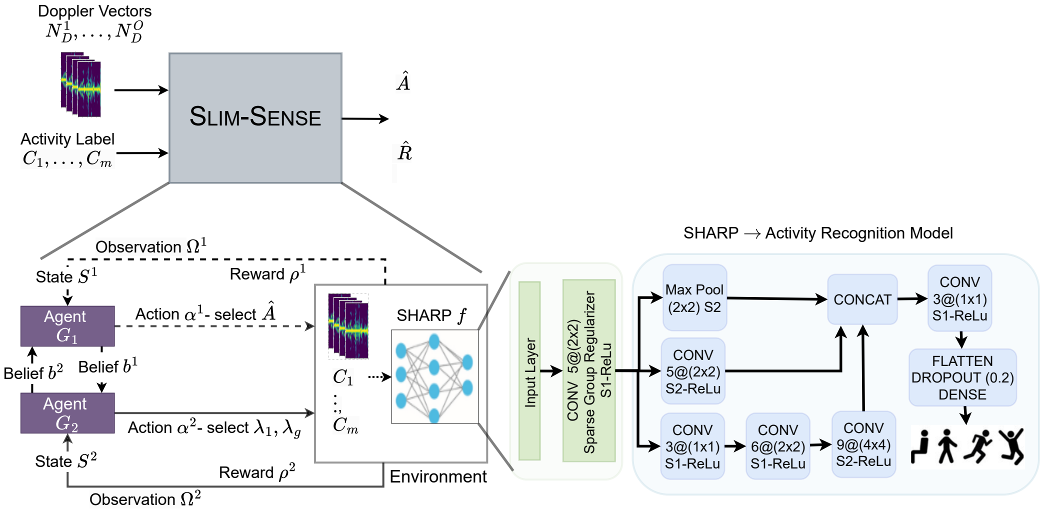


Fig. 3. Slim-Sense mainly consists of SGR and HRL. SGR and HRL provide the most optimal $\hat{R}$ and $\hat{A}$. It takes Doppler vectors and activity labels as input and provides $\hat{R}$ and $\hat{A}$ as output. For the convolution layers, the symbol "@" represents stacking of vectors.

4.2 Components of Slim-Sense

Slim-Sense has two components, activity recognition model (f) with SGR and Hierarchical Reinforcement Learning (HRL) model, as shown in Fig. 3. The HRL model provides the most optimal combination of hyperparameters, whereas f with SGR provides the most optimal resources controlled by hyperparameters. We now discuss each of these components.

4.2.1 Activity Recognition Model (f) with SGR. We utilize SHARP, a state-of-the-art human activity recognition model, to implement the function f (details in Section 2.3). However, the SHARP model only minimizes the loss function Q_1 in Equation (15) and does not consider the amount of bandwidth resources used. The model complexity Q_2 and resource selection Q_3 in Equation (15) are jointly optimized by the Sparse Group Regularizer (SGR). The SGR applies a weight on selection of each channel carrier, both individually and as a group, depending on the sub-channel that the carrier belongs to. This weighted sum is added as a penalty to the other objective of maximum accuracy. In this way, the regularizer pushes the weights of the particular Doppler vector to zero if the Doppler vector does not contribute significantly to minimizing the recognition loss/error. Hence, unimportant Doppler vectors are discarded, and only the important/relevant group of Doppler vectors are retained. To combine this with improvement in accuracy, we fuse the Sparse Group Regularizer (SGR) directly following the input layer using the convolutional layer in the architecture of the SHARP model, as shown in Fig. 3. We strategically position the SGR just after the input layer so that the model can process and interpret the initial input data effectively before it is further analyzed by the SHARP model. It enables an efficient approach to feature regularization just before the data processing pipeline. The SGR and the SHARP model recognize activities with the most relevant resources (Doppler velocities and the group of Doppler vectors). The selection of resources is controlled by hyperparameters tuned by HRL.

4.2.2 Slim-Sense's Hierarchical Reinforcement Learning Technique. We model the problem of hyperparameter tuning as an instance of the Partially Observable Markov Decision Process (POMDP). A standard technique for solving the POMDP problem for hyperparameters defined in §. 4.1 is to utilize hierarchical reinforcement learning (HRL) shown in Fig. 3, as the number of possible actions is large in number. The total possible actions are equal to $|\mathbf{P}(A)| \times |\Lambda_1| \times |\Lambda_g|$. For example, IEEE802.11ax allows up to 160 MHz bandwidth with $R = 74$ RUs and up to $A = 8$ antennas. Furthermore, we empirically observe that 5 distinct values of λ_1 and λ_g are generally used, leading to an overall search space of $(2^8 - 1) \times 5 \times 5 = 6375$ possible choices of actions. Since a major goal of our problem was to have relatively low computation, we use two parallel RL agents, denoted by G_1 and G_2, as shown in Fig. 3. Both agents interact with the environment and choose separate actions (selecting hyperparameters) as part of their learning. The environment is configured by a labeled Doppler vector dataset with a SHARP f model with SGR in a specific indoor location. G_1 selects most optimal combination of antennas $\hat{A}$, whereas G_2 selects most optimal pair of λ_1 and λ_g. Hence, the overall search space is reduced to $(2^8 - 1) + 5 \times 5 = 280$. Both agents interact with the environment by training the SHARP model with SGR using the labeled Doppler vector dataset and selected hyperparameters. Next, the agents G_1 and G_2 obtain the sensing accuracy from the unseen environment and compute the reward function ρ^1 and ρ^2 from Equation (16), respectively. The agents G_1 and G_2 maintain belief state b^1 and b^2 to account for the uncertainty of the environment in the current state. The agent chooses an action (selecting hyperparameter) based on the belief state using the ϵ-greedy strategy. The agents G_1 and G_2 updates their belief state based on the observation received from the environment as well as other agent's belief states (b^2 and b^1 respectively). We used *Q-learning* to train the agents and formulate the Bellman equation to update the Q-value as:

$$Q(b^1, \alpha^1) = Q(b^1, \alpha^1) + \beta[\rho^1 + \gamma \max_{\alpha^1} Q(b^2, \alpha^2)] - Q(b^1, \alpha^1), \text{ and}$$
$$Q(b^2, \alpha^2) = Q(b^2, \alpha^2) + \beta[\rho^2 + \gamma \max_{\alpha^2} Q(b^1, \alpha^1)] - Q(b^2, \alpha^2) \tag{17}$$

where $Q(b^1, \alpha^1)$ and $Q(b^2, \alpha^2)$ are the Q-values of each agent and β is learning rate. Slim-Sense uses an ϵ-greedy method to select the action with $\epsilon = 0.1$ so that a random action is taken 10% of the time. We now discuss the process of actual running of Slim-Sense during both training and testing.

4.3 Working of Slim-Sense

We integrate the resource selection for sensing in the training procedure, allowing Slim-Sense to identify the minimum optimal resources while training. First, Slim-Sense, through the HRL model, determines the hyperparameters needed to train and test the SHARP model. Next, through SHARP with SGR model, obtains the resources for sensing while training with given hyperparameters in the training environment. Thereafter, it evaluates the trained SHARP model in new or unseen environments and obtains the sensing accuracy. Through this, Slim-Sense's HRL model adapts to a new environment by choosing the right set of hyperparameters. This way, Slim-Sense, on convergence, obtains the optimal combination of $\hat{A}$, λ_1, and λ_g, and in turn, minimum optimal resources $\hat{A}$ and $\hat{R}$. We now discuss the process in detail.

Slim-Sense, in a specific environment setup, initializes action space α with available antenna combinations and sets of lambda values as shown in Table 3. The learning agent G_1 chooses the unique combination of antennas from $P(\mathbf{A})$ and utilizes λ_1 and λ_g chosen by G_2. Next, G_1 observes the sensing environment by training the SHARP (f) with SGR in the training location and obtains the sensing accuracy in the new or unseen locations (testing). G_1 gets the immediate reward ρ^1 defined as Equation (16) utilizing testing sensing accuracy and selected resources ($|\hat{A}|$ and $|\hat{R}|$). Note that f can partially observe the environment state. Hence, G_1 obtains the observation of the current state by introducing random uncertainty in the ρ^1. Based on the observation, it updates its belief state b^1 and shares with G_2. Now, G_2 updates its belief state b^2 considering b^1. Next , G_2 chooses λ_1 and λ_g from Λ_1 and Λ_g, respectively and utilizes $\hat{A}$ chosen by G_1. Like G_1, G_2 gets the reward ρ^2 and obtains observation of the current state by introducing random uncertainty in ρ^2 and updates its belief state b^2. G_2 shares its belief state with G_1. Like G_2, G_1 updates its belief state b^1 considering b^2. Both agents compute their Q-values using their own belief states using Equation (17), and then updates its belief states based on the observed environment. On convergence, G_1 provides $\hat{A}$ and G_2 provides λ_1 and λ_g. Using optimal values of $\hat{A}$, λ_1, and λ_g Slim-Sense provides $\hat{R}$.

Note that though we have used ground truth labels in our Slim-Sense, in a new environment where ground truth is not available, we initially allow Slim-Sense to utilize the entire bandwidth for a calibration stage, assuming that the activity recognized using the entire bandwidth is the ground truth. We show later in Section 6.2.2 that this calibration takes about 2 *minutes of time .*

5 Evaluation Methodology

We now describe our evaluation methodology, which consists of a discussion of the datasets, evaluation scenarios, baseline approaches and evaluation metrics.

5.1 Description of Dataset

Table 4 summarizes the details of our collected dataset HeadGest and pre-existing datasets. We have chosen the datasets to evaluate the performance in a diverse variety of challenging conditions, including cluttered areas,

Table 4. Summary of Datasets.#Sub-channels/RUs: Total number of subchannels or RUs #Samples: Total number of samples in each activity, #Duration: Duration of each activity, #Participants: Total number of participants, #Dist: Distance between Tx and Rx.

Dataset Parameter	**HeadGest**	**Exposing_CSI [8]**	**SHARPax [26]**	**SimWiSense [14]**
Standard	802.11n	802.11ax	802.11ax	802.11ac
Access Technology	OFDM	OFDM/OFDMA	OFDM/OFDMA	OFDM
Bandwidth	$40 - MHz$	$160 - MHz$	$80 - MHz$	$80 - MHz$
#Sub-channels/RUs	2	8/74	4/37	4
Sampling Rate L	100 CSI/sec	150 CSI/sec	133 CSI/sec	300 CSI/sec
Tx and Rx	1 Tx and 1 Rx with 1 antenna	1 Tx and 3 Rx with 4 antenna	1 Tx and 1 Rx with 4 antenna	1 Tx and 3 Rx with 1 antenna
Tx-Rx distance	$1 - 5$ m	$2 - 6$ m	4 m	$1.5 - 2$ m
#Participants	33	3	1	3
#Samples/activity	35000 (minimum)	48000	63840	50000 (minimum)
Duration/activity	$2 - 35$ minutes	1.33 minutes	2 minutes	2.7 minutes
Total Duration	51 hours	1.86 hours	16 minutes	2.7 hours
Total Samples	$7,349,761$	$4,032,000$	$251,040$	$750,000$
Activities	Looking Forward, Looking Down, Looking Up, Looking Left, Looking Right, Nodding and Shaking	Walk, Sitting, Wave hands, Wiping, Run, Empty room, Clapping, Squat, Jump, Standing, Lay down, Stretching	Walking, Running, Staying, Empty room	Push Forward, Rotate, Hands up and down, Waive, Brush, Clap, Sit, Eat, Drink, Kick, Bend forward, Wash hands, Call, Browsing phone, Check wrist, Read, Waive while sitting, Writing, Side bend, and Standing
Locations	Lab-1, Lab-2, Lab-3, Meeting Room and Housing Room	Lab, Office, Hall	House corridor	Classroom, Office
#Scenarios	46	7	1	2
Uniqueness	Challenging and diverse experiment setups	11ax CSI data for 12 static, dynamic and physical activities	11ax data in simple scenario	Collected in multi-person activities environment

multiple people, different scenarios of training and testing, as well as fine-grained activities. We now discuss each of them in detail:

(1) **HeadGest**: This dataset is used to evaluate the diversity in training and testing environments and includes a large number of participants. This is ensured by collecting the data in different phases (Phase-1, Phase-2, Phase-3), where each phase consists of a distinct setup of the transmitter-receiver pair. The data collection setup consists of one standard laptop (with Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz with 8 GB RAM) and two WiFi-enabled ESP32s microcontrollers with a single antenna. We use the CSI extraction toolkit [16]

Table 5. Summary of the HeadGest Dataset.

Scenarios	#Participants	Location	Setup	#Experiments	#Samples	Duration
Phase-1	6	Meeting Room	AP and Rx at 4m apart	1	1,390,950	40 hours
	3	Housing Room		1	746,957	
	10	Lab-1		1	2,213,329	
	3	Lab-2		1	879,153	
Phase-2	3	Lab-2	Changing angles	6	54,637	3 hours
			Varying distance	5	173,609	
			Crowded locations	6	97,489	
			AP wall mounted	4	203,111	
			Remote deployment	3	139,628	
			Interferer	2	91,374	
			Obstacles close to AP	4	153,610	
Phase-3	8	Lab-3	Obstacles close to monitor	12	1405082	8 hours

to extract the CSI values. We configured one ESP32 as an active access point (Transmitter) and the other as an active station (monitor). $Transmitter$ transmits 40MHz-IEEE802.11n traffic with a sampling rate of 100 packets/second. The datasets are collected in five different places – (i) Meeting Room ($4m \times 2.5m$) is small in size and consists only of basic furniture and a fan, (ii) Housing Room ($4m \times 3m$) is medium in size and has a wardrobe, two beds, and a study table. Housing room was shared with one person during data collection; (iii) Lab-1 ($12m \times 8m$) is large in size and consists of different furniture and desktops, and very few people were present during data collection (iv) Lab-2 ($12m \times 8m$) have the same characteristics as Lab-1 and was crowded and (v) Lab-3 ($3m \times 5m$) were empty, and only participants were present during data collection. Additional details about the data collection setup are given in Tables 4-5. Phase-1 dataset is collected with 22 participants in Lab-1 (10 Participants), Crowded Lab-2 (3 participants), Meeting Room (6 participants), and Housing Room (3 participants) in 4 different experimental setups in which the distance between transmitter and monitor devices is set to 2 m.

The Phase-2 dataset varies in a number of settings to study performance in various situations, namely, *1) Varying distance:* To capture the impact of distance between transmitter and receiver, we set the initial distance to $1m$, then increment the distance by $1m$ up to $5m$. *2) Changing angles:* Different angles can capture various environmental reflections. We set the distance between AP and Rx to $2m$ and rotate the AP around the monitor. Hence, the angle between AP and monitor varies to different angles such as 0°, 45°, 90°, 135°, 180° and 270°. *3) Crowded locations:* We collected the CSI samples in the crowded scenarios in Lab-3 where $10-15$, $15-20$, and $20-25$ people are present during data collection. Crowded environments lead to a more complex multipath propagation, making the sensing more challenging. *4) AP wall mounted:* To collect the CSI data in a setup where AP is normally wall mounted in a real scenario. The monitor is placed in 4 different locations with varying angles and distances. *5) Remote deployment:* We placed the AP in one location and the monitor in another location to evaluate the sensing performance in the through-wall scenario. *6) Interferer:* In the real scenario, the interferer (non-target) might present close to the target participants. Hence, we collect the CSI data in the scenario where the non-target person sits close to the target participants. *7) Obstacles close to AP:* The obstacle close to the AP creates additional reflections and scattering, altering the signal's multipath components. We placed obstacles of different materials such as wooden door ($1in$ width), wooden panel ($2in$ width), glass ($0.25in$ width), and nylon sheets (4 sheets).

In Phase-3, we collected data with 8 participants in Lab-2 in 12 different experimental setups. In this phase, we placed obstacles of three materials, such as *wood, glass, and fiber*, between the AP and monitor. We use a total of 4 sheets of each material, starting with 1 sheet, then 2 sheets, followed by 3 sheets, and finally 4 sheets. This setting represents the most challenging scenarios of WiFi sensing deployment, where signal attenuated significantly at Rx results in noisy CSI.

We collected a total of 7349761 CSI samples over a duration of 51 hours, involving 33 participants in our own lab. For each participant, a total of 155300 samples are collected. For each head gesticulation, a minimum of 16000 samples are collected. We shared a consent form regarding the usage of the dataset for this study with our institute's IRB approval.

(2) **Exposing_CSI:** Exposing_CSI dataset is one public dataset [8] which used IEEE 802.11ax for a relatively larger number of activities (12) ranging from static, dynamic, and physical. It also uses the highest bandwidth of 160 MHz, making the use of Slim-Sense especially relevant in this case. The dataset is collected in 3 different locations such as *Lab, Office, and Hall.* The *Lab* is a medium-sized room with more clutters, hence rich multipath reflections. The *Office* is a small room with more clutter and rich multipath reflection. However, the hall is a larger room with no clutters; hence, the reflection is only from the ceiling and floor. A total of 3 participants annotated as *A, B, and C* participated in data collection. The dataset reflects scenarios such as *1) S1-Lab*: same person, same environment, *2) S2-Lab*: different person, same environment, *3) S3-Lab, S4-Lab, S5-Lab*: same person, same environment on different days, *4) S6-Office*: Same person different environment, and 5) *S7-Hall*: same person different environment. In scenarios, *S1-Lab, S4-Lab, S5-Lab, S6-Office, and S7-Hall*, CSI data is collected with Participant *A*. Moreover, in *S1-Lab and S4-Lab*, data is collected on the same day at different times. In S5-Lab, data is collected on the next day. In *S2-Lab* and *S3-Lab*, data is collected with Participants B and C. The sampling rate and duration of CSI data collection for each activity are 150 packets/second and 80 seconds, respectively. Hence, for each activity, $150 \times 80 = 12000$ samples are collected at each antenna in 7 different scenarios.

(3) **SHARPax:** SHARPax [28] is the other public dataset that performed activity recognition using the IEEE 802.11ax standard. Compared to Exposing_CSI, this uses a relatively simpler scenario of a house corridor with a relatively fewer (4) number of activities. However, compared to other datasets, the distance between the transmitter-receiver pair is fixed and is on the higher side, i.e., $4m$, making the sensed signals weaker. The house corridor is medium in size, and reflection surfaces include only wall, ceiling, and floor. The dataset is collected in one scenario *S1-House corridor*. The sampling rate and duration of CSI data collection for each activity are 133 samples/second and 2 minute, respectively. Hence, for each activity, $15960 \times 4 = 63840$ samples are collected at 4 antennas of the monitor device.

(4) **SimWiSense Dataset** This dataset is collected in multi-person environments with 3 participants performing 20 activities simultaneously using IEEE 802.11ac standard. The presence of multiple people and 20 different activities add a new layer of complexity to activity recognition on this dataset. The dataset is collected in two scenarios: *Classroom* and *Office*. The distance between AP and monitor is set to $1.5m - 2.0m$. In both scenarios, all 3 participants were performing activities at a distance of $1.5m - 2.0m$ from AP. The sampling rate and duration of CSI data collection for each activity are 300 samples/second and 2.7 minutes. Hence, for each activity, 50000 samples are collected at each monitor device.

Preparing Input Dataset: We prepare the Doppler vector dataset ($[N_D^1 \dots N_D^O]$) from HeadGest, Exposing_CSI, and SHARPax datasets. First, we normalize the CSI values for each CSI sample by dividing them by the mean amplitude over all considered subcarriers to remove unwanted amplification. Second, we remove the phase offset using the phase sanitization algorithm, as used in [28]. We thus obtained the CSI complex-valued vector with amplitude and phase (the real term indicates amplitude, and the imaginary term indicates phase). Third, we compute the Doppler vector ($D_i^{r,a}$) by taking $W = 31$ sanitized CSI samples of all considered subcarriers of each r for each observation window i. The size of Doppler bins is set to V for each $D_i^{r,a}$ before applying Fourier transform. We empirically obtained the optimal value of V (details in Section 6.3.4). Thus, we obtained N_D^i for each observation window i. We obtained the Doppler vector traces dataset ($[N_D^1 \dots N_D^O]$) by sliding the observation window with a step size of 1 row. The total number of observation windows is $O = N - (W - 1)$. *In this work, we considered a maximum of $A = 4$ antennas at the monitoring devices.* The SimWiSense solution used amplitude as input features to train the sensing model. Hence, we compute the amplitude as input features from SimWiSense's CSI samples for each subcarrier.

Table 6. Training and Testing scenario of the datasets.

Dataset	Training Scenario	Testing Scenario
HeadGest	Phase-1	Phase-2 and Phase-3
Exposing_CSI	S1-Lab	S2-Lab, S3-Lab, S4-Lab, S5-Lab
SHARPax	S1-Corridor	S1-Corridor
SimWiSense	Classroom	Office

5.2 Evaluation Scenarios

Table 6 shows the training and testing scenario of all 4 datasets. We train Slim-Sense at a particular training scenario and then evaluate it both at the same as well as unseen locations (testing scenarios). On each dataset, we use 60% of the samples for training, 20% for testing, and 20% for validation, all extracted from training scenarios. Slim-Sense's evaluations in testing scenarios indicate how well Slim-Sense adapts to different persons and environments. To examine the generability of Slim-Sense in applying to a new sensing framework, we also evaluate Slim-Sense with a different sensing model SimWiSense [14]. To do this, we replaced the SHARP model (f) with the SimWiSense's few-shot embedding learning (FREL) model and evaluated Slim-Sense using the SimWiSense dataset.

5.3 Baselines

We compared Slim-Sense with the following five baseline approaches:

(1) **Static configuration:** We compare the performance of Slim-Sense with three static configurations – ≈ 74% **Resources,** ≈ 45% **Resources,** ≈ 30% **Resources**.[2] We use the SHARP model with SGR and tune λ_1 and λ_g to obtain 75%, 45%, and 30% resources and compute the sensing accuracy. The % of resources saved depends on λ values, hence we get selected resources as ≈ 75%, ≈ 45% and ≈ 30%.

(2) **Max_Accu:** In this approach, we use all the antennas A of the monitoring device and sub-channels/RUs R to compute the Doppler vectors and train the SHARP model with SGR setting $\lambda_1 = 0$ and $\lambda_g = 0$. We termed the achieved sensing accuracy as *Maximum Achievable Accuracy (Max_Accu).*

(3) **Random-Selection:** We randomly select the $\hat{A} \in P(\mathbf{A})$ and sub-channels/RUs $\hat{R}$ in this approach. We compute the Doppler vectors and train the SHARP model with SGR setting $\lambda_1 = 0$ and $\lambda_g = 0$.

(4) **Reduced-Redundancy:** First, we compute *cross correlation* amongst RUs/sub-channels for all antennas. Next, we use the *HDBSCAN* clustering algorithm to cluster the sub-channels/RUs based on cross-correlation values. We ensure a minimum of one sub-channel/RU in each cluster. We then select the first sub-channel/RUs iteratively from each cluster.

(5) **WiImg:** As discussed in Section 2.4 WiImg [42] enables WiFi sensing under low-rate CSI samples by generating synthetic samples. As the codebase for WiImg is not publicly available, we re-implement it. As followed in WiImg, we convert the CSI data of the first three antennas $\{0, 1, 2\}$ and the entire channel bandwidth into images. In the case of HeadGest dataset, we use single antennas to represent the RGB. We downsample the collected CSI samples to get the images of low-rate CSI samples. We then apply GAN

[2] The percentage of used resources are controlled by SGR with tuning λ_1 and λ_g, hence the %of Resources might not be exact.

that uses inpainting to generate the images of high-rate CSI samples (original sampling rate used in the specific dataset). To validate the correctness of the implementation, we compute a cosine similarity between inpainted and original images. We obtain a similarity index of 0.9 versus the original study's similarity index of 0.92 for all four datasets, validating the correctness of the implementation. We prepare the training data of high-rate inpainted images (synthetic samples) from training scenarios and testing data from testing scenarios. We use the state-of-the-art THAT [22] model, which has been utilized in WiImg, to obtain the sensing accuracy. While downsampling, we choose the sampling rate that provides maximum sensing accuracy. For example, WiImg obtains 54.3%, 55%, 62.50,and 62% sensing accuracy with downsampled sampling rate of 25, 50, 75, and 100 samples per second, respectively, with Exposing_CSI (150 samples per second). Thus, we choose the sampling rate of 75 to compare with SimWiSense.

6 Evaluation Results

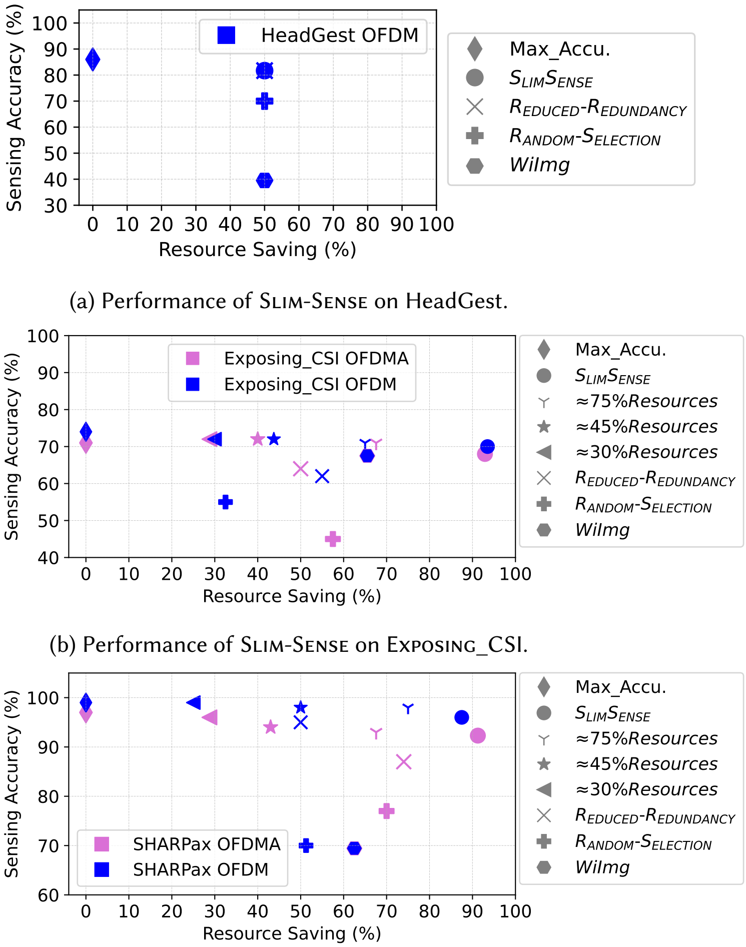
We now present the results of our evaluation. We first present the optimal trade-off between sensing accuracy and resource-saving for Slim-Sense compared with baselines for 4 datasets. We then evaluate the generalization of Slim-Sense across various environments, a new environment without ground truth, and in terms of applying a new sensing solution to it. Finally, we showcase the detailed working of Slim-Sense through resource selection mechanism, confusion matrix across different activities, training and testing time, ablation study, and convergence of Slim-Sense.

6.1 Comparison of Slim-Sense with Baselines

We now discuss the performance of Slim-Sense and compare it with baseline approaches. Fig. 4 shows the trade-off between sensing accuracy and resource saving for HeadGest, Exposing_CSI and SHARPax.

6.1.1 Slim-Sense's Performance on HeadGest Dataset. HeadGest dataset consists of a total of 46 experimental setups in challenging scenarios such as obstacles between transmitter and receiver, interferers close to target person and crowded environment. As shown in Fig. 4a, Slim-Sense is able to provide 50% resource saving with only 4.3% reduction in accuracy compared to Max_Accu for *HeadGest* dataset. Slim-Sense adapts to challenging scenarios by obtaining the most relevant resources and utilizing the Doppler vectors as input features. For *HeadGest*, with only two sub-channels available, the maximum achievable resource saving is limited to 50%. Static configuration ≈ 30 , ≈ 45% and ≈ 75% is not applicable. Compared to Random-Selection, Slim-Sense provides 11.7% more sensing accuracy with the same 50% resource saving. Randomly selecting resources for sensing does not adapt well to challenging scenarios. Compared to Reduced-Redundancy, Slim-Sense provides a similar trade-off between sensing accuracy and resource saving. Reduced-Redundancy in limited available sub-channel, select most relevant sub-channel for sensing. Similarly, WiImg achieves similar resource saving compared to Slim-Sense by downsampling to 50 packets/ second from an initial sampling rate of 100 packets per second. This comes at a cost of 42.18% lower accuracy compared to Slim-Sense. Low-rate CSI samples reduce temporal resolution, hence reducing the sensing accuracy. WiImg's GAN obtains high-rate synthetic CSI samples from low-rate CSI amplitude samples. In the diverse and challenging scenario, obtained synthetic CSI amplitude samples only approximate the high rate of the original samples. Hence, WiImg's sensing model fails to adapt to challenging scenarios trained with synthetic samples of low-rate CSI amplitude samples.

6.1.2 Slim-Sense's Performance on Exposing_CSI Dataset. Exposing_CSI use IEEE 802.11ax (reduced sub-carrier spacing between subcarriers) CSI samples for the relatively large number of activities and the highest bandwidth of 160MHz with 4 antennas. Fig. 4b shows that with Exposing_CSI dataset, Slim-Sense achieves over 92% resource saving with a 3% and 4% reduction in sensing accuracy compared to Max_Accu (71% and 74%) for OFDMA and OFDM respectively. Compared to ≈ 30 , ≈ 45% and ≈ 75%, Slim-Sense incurs a 2.66% mean reduction in



(a) Performance of Slim-Sense on HeadGest.

(b) Performance of Slim-Sense on Exposing_CSI.

(c) Performance of Slim-Sense on SHARPax.

Fig. 4. Trade-off between Sensing Accuracy and resource saving: Slim-Sense vs. baselines across different datasets.

accuracy while achieving 47.34% more resource saving. Static configurations with manual resource selection fail to provide the optimal trade-offs. Compared to Random-Selection, Slim-Sense achieves 23% and 15% more sensing accuracy with 35.4% and 61% more resource saving for OFDMA and OFDM, respectively. Random-Selection fails to provide maximum resource saving due to random selection of resources. Compared to Reduced-Redundancy, Slim-Sense achieves 4% and 8% more sensing accuracy with 42.9% and 38.75% more resource saving for OFDMA and OFDM, respectively. WiImg achieves a sensing accuracy comparable to Slim-Sense with a sampling rate of 75 packets/sec in OFDMA and OFDM, respectively. But Slim-Sense achieves 27.4% and 28.25% more resource saving compared to WiImg in OFDMA and OFDM, respectively. WiImg performs well in testing scenarios using generated synthetic samples, achieving a sensing accuracy comparable to Slim-Sense. However, this comes at the cost of lower resource saving, which affects the overall system performance. This shows that Slim-Sense performs well across diverse activities with IEEE 802.11ax CSI samples.

6.1.3 Slim-Sense's Performance on SHARPax Dataset. SHARPax uses IEEE 802.11ax CSI samples collected in one location to capture fewer (4) number of activities. SHARPax uses a relatively simpler scenario compared to other datasets listed in Table 4. The distance between the transmitter-receiver pair is fixed at $4m$, which is on the higher side. Fig. 4c shows that Slim-Sense achieves 91.25% and 87.5% of resource-saving with a sensing accuracy of 92.3% and 96% with SHARPax dataset for OFDMA/OFDM respectively. The reduction in sensing accuracy is only 3% and 3.7% compared to Max_Accu. Similarly, compared to ≈ 30, $\approx 45\%$ and $\approx 75\%$ Slim-Sense provides better trade-off with 2.06% and 2.33% mean reduction in sensing accuracy with 44.81% and 37.05% more resource-saving in OFDMA/OFDM, respectively. Compared to Random-Selection, Slim-Sense archives 15.3% and 26% more sensing accuracy and 21.25% and 36.25% more resource saving for OFDMA and OFDM respectively. Similarly, compared to Reduced-Redundancy, Slim-Sense achieves 17.25% and 37.5% more resource saving in OFDMA and OFDM. However, Reduced-Redundancy archives similar sensing accuracy compared to Slim-Sense. This indicates that in simple scenarios, Reduced-Redundancy, and static configurations archive similar sensing accuracy as Slim-Sense. However, Slim-Sense achieves more resource saving. Slim-Sense achieves 22.86% and 18.06% more sensing accuracy than WiImg (sampling rate of 65 packets/sec) in OFDMA and OFDM, respectively. Moreover, Slim-Sense achieves 28.75% and 25% more resource saving compared to WiImg in OFDMA and OFDM, respectively. These results show that Slim-Sense outperforms the baselines in simpler environments as well.

6.2 Generalizability of Slim-Sense

In this section, we discuss the generalizability of Slim-Sense in three key areas: adapting to new and challenging environments, deploying to a new or unseen environment, and adapting to a new sensing solution.

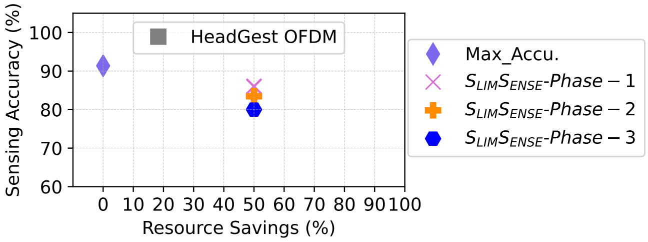


Fig. 5. Adaptability of Slim-Sense by testing in both seen or unseen environments during training present in our HeadGest Dataset.

6.2.1 Performance of Slim-Sense in Challenging Environmental Conditions. Since the HeadGest dataset is collected in the most diverse and challenging environment, with obstacles present between Tx and Rx, as well as the presence of crowd, we utilize this dataset for this experiment. Furthermore, in Phase-2, the placement of Tx and Rx is not controlled. Phase-2 and Phase-3 together represent environmental scenarios with varying noise levels, environmental dynamics, and complexity.

We evaluate Slim-Sense in three scenarios:

(1) Slim-Sense-Phase-1: Trained in Phase-1 and evaluated on Phase-1's new or unseen locations,
(2) Slim-Sense-Phase-2 - Trained in Phase-1 and evaluated on Phase-2, and
(3) Slim-Sense-Phase-3 - Trained in Phase-1 and evaluated on Phase-3. Max_Accu represents the maximum achievable sensing accuracy on Phase-1.

Fig. 5 shows the performance of Slim-Sense. In these scenarios, Slim-Sense achieves 50% resource saving and obtains over 80% sensing accuracy in all scenarios. The sensing accuracy of Slim-Sense is lower by 5.36% than

Max_Accu in Phase-1. Moreover, in challenging environmental settings, Slim-Sense provides a sensing accuracy lower by 7.91% in Phase-2 and 11.36% in Phase-3, respectively, than Max_Accu. Slim-Sense trains the SHARP model with SGR in Phase-1, evaluates in Phase-2 and Phase-3, and adjusts the controlling hyperparameters. Slim-Sense explores different hyperparameters to provide the optimal trade-off between sensing accuracy and resource saving. In this way, the selected hyperparameters and resources used for training the SHARP model with SGR enable the model to be adapted even in challenging settings. This shows that while the accuracy of sensing of Slim-Sense does suffer under challenging environmental conditions, the reduction in accuracy is relatively small.

Table 7. Performance of Slim-Sense in a new environment (*Office*) while trained on a different environment (*Lab*) utilizing Exposing_CSI dataset.

Convergence time	**Max Accuracy**	**Sensing Accuracy**	**Resource Saving**
120s	70%	5.40% reduction	81.25%

6.2.2 Deploying Slim-Sense in a New Environment. We now show the performance of Slim-Sense in a new environment when the ground truth activity label is not available (detail in Section 4.3). We train Slim-Sense with Exposing_CSI dataset of *Lab* environment. Next, we deploy the trained Slim-Sense in *Office* environment of Exposing_CSI. We utilize all antennas and sub-channels/RUs to recognize the activities in *Office* environment to obtain the ground truth C_i. Note that for Slim-Sense to function, the same activities must be present. Slim-Sense utilizes trained *Q-table* to select the $\hat{A}$ and λ_1 and λ_g. Next, Slim-Sense interacts with the environment and returns $\hat{A}$ and $\hat{R}$ (controlled by λ_1 and λ_g). The SHARP with SGR provides recognized activity label $\hat{C}_i$. Thus, we obtained the sensing accuracy (70%). We did not retrain Slim-Sense on *Office* environment. To evaluate the impact of choosing the slimmest possible bandwidth resource ($\hat{A}$ and $\hat{R}$), we compute Max_Accu (74%) utilizing C_i and the actual activity label provided in the *Office* dataset. Compared to Max_Accu, Slim-Sense achieves 81.25% resource saving with 5.40% reduction in accuracy while taking only $\approx 120 seconds$ to find the relevant $\hat{A}$ and $\hat{R}$ as listed in Table 7.

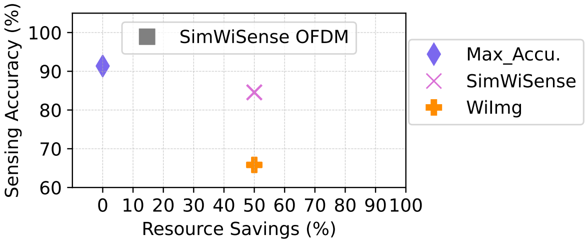
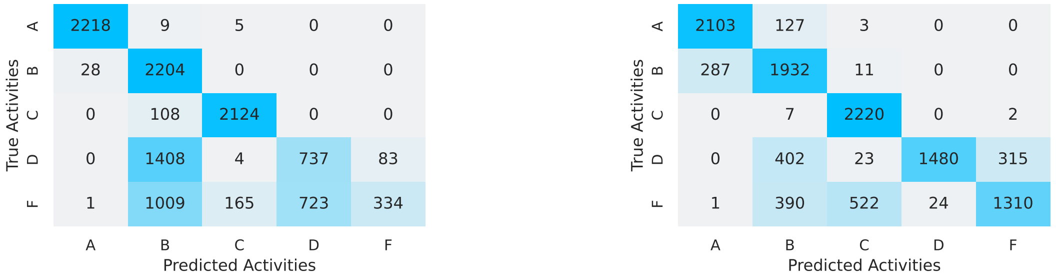


Fig. 6. Performance of Slim-Sense with SimWiSense model compared with WiImg.

6.2.3 Performance of Slim-Sense with SimWiSense's Sensing Model. Now, we investigate the generalizability of Slim-Sense with respect to adapting to a new sensing model. We confirm this aspect of generalizability of Slim-Sense by changing our f from SHARP model to SimWiSense's FREL model. We integrate FREL with sparse group regularizer and reinforcement learning. For evaluation, we utilize the multi-person environment dataset from SimWiSense, collected at a sampling rate of 300 CSI samples/second. Slim-Sense is trained with CSI samples collected in the classroom environment and tested with CSI samples in the office environment. Slim-Sense with FREL achieves 84.56% sensing accuracy with 50% resource saving, as shown in Fig. 6. Next, for WiImg, the sampling rate is selected as 150 CSI samples/second. WiImg achieves similar resource saving compared with Slim-Sense. However, Slim-Sense achieves 18.73% more sensing accuracy compared to WiImg. In conclusion, Slim-Sense can be easily integrated into other sensing solutions and achieve similar sensing accuracy while saving resources for communication.

6.3 Detailed Functioning of Slim-Sense

We now discuss the details of the functioning of Slim-Sense and the ablation study.



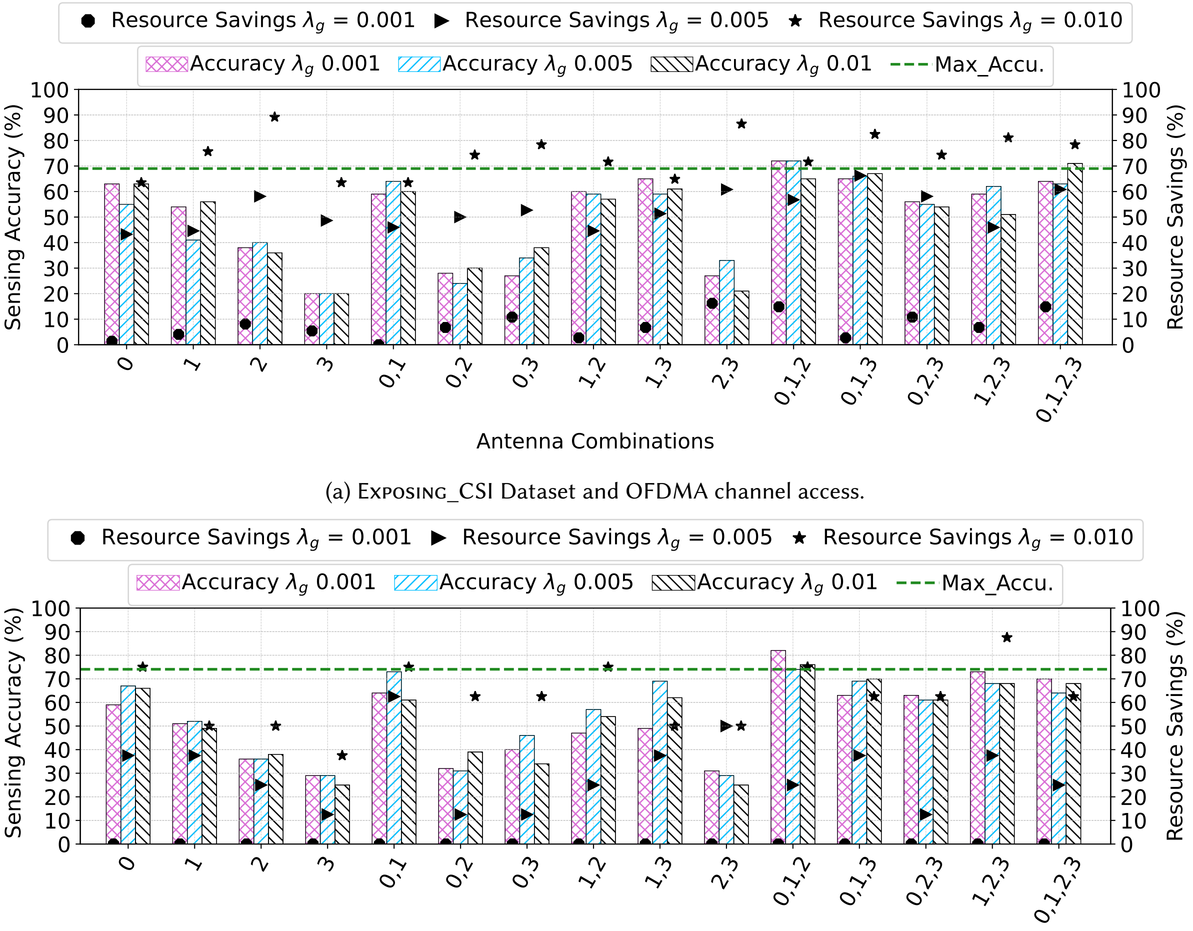
(a) Confusion Matrix for 5 activities for Exposing_CSI with Max_Accu.

(b) Confusion Matrix for 5 activities for Exposing_CSI with Slim-Sense with 92.5% resource saving.

Fig. 7. Confusion matrices for five activities (A: Walk, B: Run, C: Jump, D: Sitting and F: Standing) for Exposing_CSI dataset.

6.3.1 Accuracy of Classification of Slim-Sense Across Activities. Fig. 7 shows the confusion matrices with Max_Accu and Slim-Sense for 5 activities for Exposing_CSI dataset. Fig. 7a shows the activity recognition performance of the SHARP model without SGR (Max_Accu), utilizing the entire spectrum resources. The model demonstrates high accuracy for specific activities but shows notable confusion between *Sitting* and *Standing* and appears overtrained for *Run* activity, resulting in poorer generalization across activities. Fig. 7b shows the activity recognition performance of the SHARP model with SGR, utilizing resources obtained by Slim-Sense with 92.5% resource saving. It demonstrates improved generalization and reduces confusion between *Sitting* and *Standing* and also mitigates overtraining on *Run* activity. To summarize, Slim-Sense effectively provides the minimum quality resources needed for sensing.

6.3.2 Resource Selection Through Slim-Sense. According to our solution approach, Slim-Sense tunes the hyperparameters that control the resource selection to ensure robust WiFi sensing across different scenarios. Slim-Sense adjusts λ_1 and λ_g along with the optimal $\hat{A}$ to minimize resource usage while maintaining a minimal reduction in sensing accuracy. We now show some sample observations of our Slim-Sense's HRL model. Fig. 8 shows the impact of different $\hat{A}$ and λ_1 and λ_g values used by Slim-Sense's HRL during training on Exposing_CSI CSI dataset. During training, Slim-Sense's HRL model trains the SHARP with SGR in the training scenario, with hyperparameters selected by learning agents. The trained model is then evaluated in testing scenarios to determine the sensing accuracy. The learning agents compute the reward based on equation (16), update their belief states, and select the next set of hyperparameters. The SHARP model is subsequently retrained and evaluated with the new hyperparameter set. HRL's agent $G1$ selects different combinations of antennas as shown in Table 3. Slim-Sense prepares training and testing scenarios dataset with the selected combination of

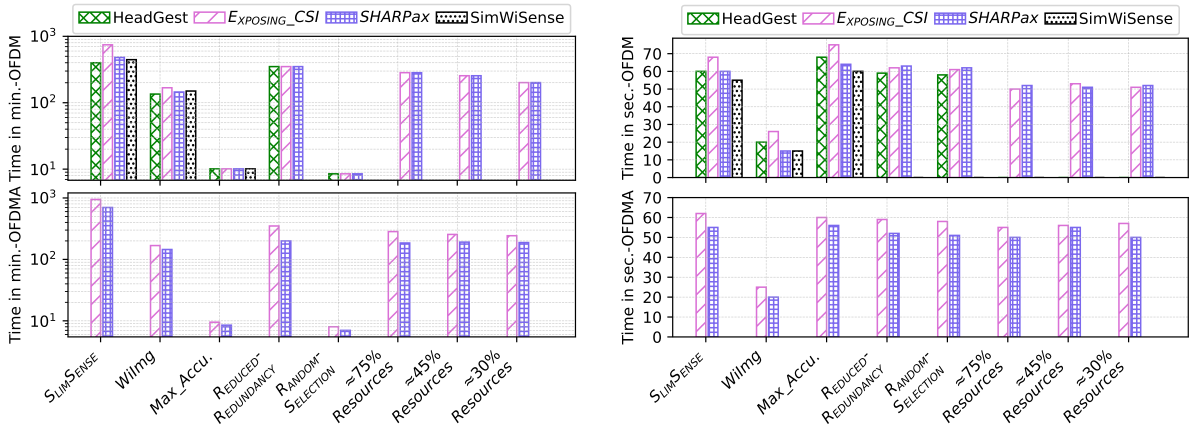


(a) Exposing_CSI Dataset and OFDMA channel access.

(b) Exposing_CSI Dataset and OFDM channel access.

Fig. 8. Impact of $\hat{A}$, λ_1 and λ_g on sensing accuracy and resource saving. Here, $\lambda_1 = \lambda_g$.

antennas. In this way, Slim-Sense explores different sets of antenna combinations and, on convergence, achieves optimal combinations of antennas. As shown in Fig. 8b and Fig. 8a, for both OFDM and OFDMA, we observe that $\hat{A} = \{1\}, \{2\}, \{3\}, \{0, 2\}, \{0, 3\}, \{2, 3\}$ provides mean accuracy of 33.8%. However, $\hat{A} = \{0\}, \{0, 1\}, \{1, 3\}$.. and other combinations provide mean accuracy of 61.5%. Hence, selecting the best antenna combination is crucial in sensing applications. Slim-Sense correctly chooses the antenna configuration as we obtain over 69% accuracy. Tuning λ_1, λ_g values carefully provides the most relevant resources $\hat{R}$ while maintaining the Max_Accu. SGR provides more resource savings by increasing the λ_1, λ_g values. Hence, Slim-Sense's HRL learns to adjust the λ_1, λ_g and $\hat{A}$ to find the optimal values for the same. To summarize, Slim-Sense achieves the optimal combination of antennas and frequency spectrum on convergence.



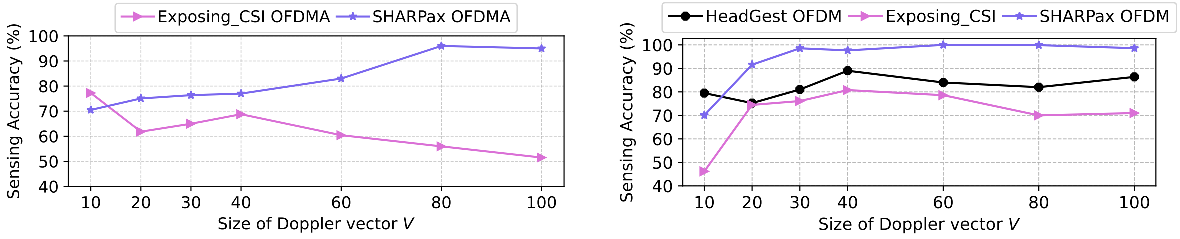
(a) Training time in minutes.

(b) Testing time in seconds.

Fig. 9. Training and testing time. Note that, for WiImg, the training time of the GAN model is also added to the training time of the sensing model. For Reduced-Redundancy, the running time of HDBSCAN is added to the training time of the sensing model. For statistical configuration, the time required to determine an appropriate configuration is included in the training time.

6.3.3 Training and Testing Time. Fig. 9 shows the training and testing (prediction) time of Slim-Sense and Baselines across all datasets. The details on the total number of training and testing samples are provided in Section 5.1. Note that the training and testing are impacted by the shape of input features and the total number of training and testing samples. Moreover, the training and testing times depend separately on the sub-channels (OFDM) and RUs (OFDMA). We train Slim-Sense and other Baseline on Ubuntu 22.04.4-LTS with AMD EPYC 7543, 32-Core Processor, 125GB RAM, with Nvidia $L40$ 48GB GPU. The GPU memory requirement during training ranges from 2GB to 10GB. The testing or evaluation is conducted on the same server, utilizing only the CPU without GPU acceleration. Testing of Slim-Sense requires 1GB to 2GB of RAM.

Statistical configuration requires manual retuning of hyperparameters to determine an appropriate configuration, and the time taken for this is included in the training time. For statistical configuration, the training is in the range of $185 - 283$ minutes for $OFDM - OFDMA$ for all datasets. On the other hand, Reduced-Redundancy first determines correlation matrices and runs the HDBSCAN clustering algorithm, then trains the sensing model based on resources selected by HDBSCAN. For Reduced-Redundancy, the training time is in the range of $200 - 350$ minutes. WiImg is required to generate high-rate CSI samples from low-rate CSI samples through the trained GAN model and train the sensing model with high-rate CSI samples. Hence, the training time, which includes both the training time of the GAN model and the sensing model, ranges from $145 - 168$ minutes. Max_Accu and Random-Selection each train the model only once, utilizing the entire resource and randomly selected resources, respectively. The training time for both is in the range of $7 - 10$ minutes. The training time of Slim-Sense is the combined training and testing times of the SHARP model with SGR across all the steps during convergence. We train Slim-Sense over 10 episodes, with each episode consisting of 15 steps. The training time of Slim-Sense is around 445 and 945 minutes for OFDM and OFDMA respectively. Slim-Sense's HRL determines the minimum resources by selecting the optimal hyperparameter through exploration and exploitation strategy. Thus, Slim-Sense takes the maximum time to train compared to baselines. However, it achieves minimum resources with minimal impact on sensing accuracy compared to baselines. The testing or prediction time for all approaches remains the same, at approximately 60 seconds. This observation indicates that Slim-Sense does not induce extra overhead in activity prediction. Similarly, when deployed in a new or unseen location, Slim-Sense takes ≈ 120 seconds to identify the minimum resources.



(a) Sensing Accuracy vs different sizes of the Doppler vector V with OFDMA as channel access technique.

(b) Sensing Accuracy vs different sizes of the Doppler vector V with OFDM as channel access technique.

Fig. 10. Comparison of sensing accuracy with different Doppler vector sizes.

6.3.4 Searching Optimal Size of Doppler Vector V. We find the optimal size of the Doppler vector V for each dataset for both OFDM/OFDMA channel access mechanisms. V defines the granularity of the Doppler velocities in the frequency domain that is captured during the short-time Fourier Transform (FFT) ($\mathcal{F}$) process. The higher/lower value of V allows for a more fine/coarse-grained analysis of the Doppler vector but requires more/fewer computational resources. The optimal V makes the activity recognition computationally efficient yet maintains the same performance. We use all the antennas A and R to obtain the Doppler vector and set the λ_1 and λ_g to *zero*. Fig. 10b and Fig. 10a show that the optimal size of V is different for all three datasets for OFDM and OFDMA, respectively. We observe that the size of V directly affects the performance of the SHARP model with SGR. We observe that the optimal value of V is 40, 60 and 40 with sensing accuracy of 80.77%, 100.0% and 91% respectively for Exposing_CSI, SHARPax and *HeadGest* dataset as shown in Fig. 10.

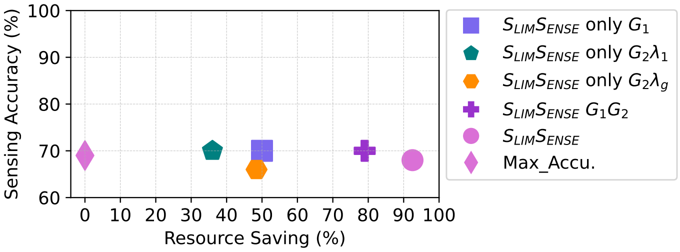


Fig. 11. Trade-off between Sensing Accuracy and resource saving: Ablation Study of Slim-Sense by removing one component at a time with Exposing_CSI dataset.

6.3.5 Ablation Study. We now justify the design choices of Slim-Sense by performing an ablation study. We retain only one component of Slim-Sense, remove the rest, and evaluate the performance of Slim-Sense. We perform this exercise with Exposing_CSI dataset. We observe that adding each of these parameters leads to substantial improvement in resource saving while giving similar levels of accuracy, as shown in Fig. 11. This confirms that each parameter introduced in the design of Slim-Sense is useful.

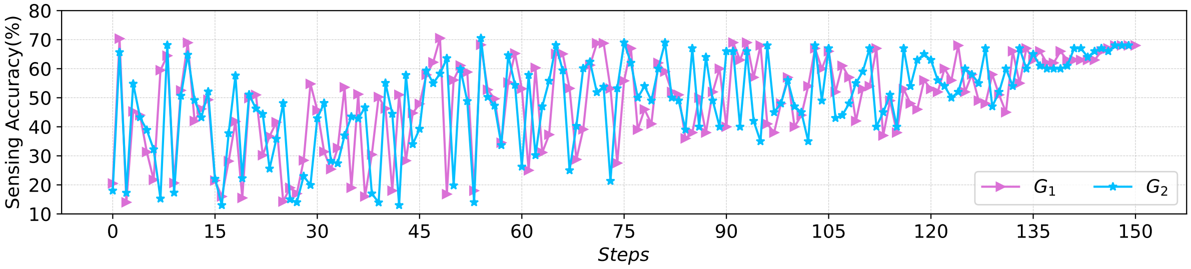


Fig. 12. Convergence time of Slim-Sense to provide the optimal $\hat{R}$ and $\hat{A}$.

6.3.6 *Convergence of Slim-Sense.*

Fig. 12 shows the convergence of Slim-Sense to obtain the optimal values of three hyperparameters λ_1, λ_g, and $\hat{A}$ and providing the maximum accuracy while saving the maximum resources for Exposing_CSI dataset. In Slim-Sense's HRL, G_1 and G_2 observe the environment by tuning three hyperparameters. Fig. 12 shows the convergence of G_1 and G_2 to achieve the *maximum Sensing Accuracy* while saving the maximum resources for Exposing_CSI dataset with OFDMA channel access. We train the HRL model for 10 episodes, and for each episode, the number of steps is set to 15. The average time for each step is 150s. In each step, both agents observe the environment separately. Initially, the fluctuation in measured Sensing Accuracy is in the range of 10% – 71%. After the 90th step, the HRL model starts converging, and at the 135th step, it converges as the values in subsequent sensing accuracy are within a smaller range of 61% – 68% and finally gets 68% sensing accuracy. This observation shows that Slim-Sense's HRL technique converges to maximum sensing accuracy after around 150 episodes.

7 Discussion

Now, we discuss the limitations and scope of future works to address them.

Longer Convergence Time: The convergence time of Slim-Sense includes the training and testing time of the sensing model f with SGR in a specific environment across all the steps during convergence. This time is influenced by factors such as environment dynamics, model complexity, and the number of samples for training and testing. The convergence time of Slim-Sense is relatively longer compared to the baseline approaches. Hence, a promising direction of future work is the integration of Slim-Sense with lightweight sensing models specifically designed for embedded devices. By replacing the sensing model f with a lightweight sensing model, Slim-Sense can be optimized for faster convergence.

Deployment Challenges and Hardware Requirement: We train Slim-Sense on the server with GPU acceleration. When deployed in real scenarios after training, Slim-Sense with inherent sensing model with SGR is able to run on sensing devices (e.g., Laptop or Desktop, smart TV, Xbox Kinect, so on [43]) to obtain the minimum resources for the target application. Hence, one of the primary challenges is ensuring that Slim-Sense can effectively run on various hardware configurations, which may not have the same computational capabilities as the training server. One key solution is to utilize a lightweight sensing model adaptable to resource-constrained devices. Our Slim-Sense's generalizability of integration to any sensing model enables incorporating a lightweight sensing model. We envision the integration of Slim-Sense to enable integrated sensing and communication in real scenarios. However, current hardware does not support ISAC, which requires the sharing of available spectrum resources between communication and sensing. The new WiFi standard IEEE 802.11bf [1] aims to address this limitation by proposing amendments to both the MAC and Physical layer of the WiFi devices. In the future, we plan to work on a live integration of Slim-Sense on WiFi devices.

Opportunistic Use of Passive Mode: In active mode, sensing devices trigger the access point to send the special probing packets at a specific interval to collect the CSI. Thus, active sensing is detrimental to communication. However, active sensing enables robust and efficient sensing solutions [30, 36, 37, 43]. We currently aim to reduce spectrum resources for active WiFi Sensing. A recent work, SenCom [15], enables a passive mode of sensing. In passive mode, sensing devices collect the CSI data from ongoing communication traffic. When communication traffic is not available, the sensing device switches to active mode. In future work, we plan to integrate Slim-Sense with the passive mode of sensing. However, such passive sensing can only be used when there is a sufficient number of communication packets.

Cost of Scaling to New Activities or a Null Activity Class: When deployed in a new or unseen environment where the class label of activities is missing (i.e., null class activities), Slim-Sense first utilizes the specific trained sensing model to recognize the class label of the activities. Then, Slim-Sense utilizes the HRL model to obtain the minimum resources. In this work, we have used the SHARP model with SGR to obtain the minimum resources for recognizing target activities. In the case of new activities, Slim-Sense is needed to retrain to adapt to new activities. However, its generalizability to new sensing solutions allows integration with systems that adapt to new activities, such as OneFi [36]. OneFi proposed a lightweight few-shot learning framework using transductive fine-tuning to adapt to new activities without retraining the entire model. A similar strategy of retraining would also be needed in case a complete change of the environmental conditions, such as a blocked antenna, leads to a strong drop in sensing accuracy.

More Extensive Evaluation: While we have evaluated Slim-Sense on 4 different types of datasets. We can extend this exercise and evaluate for more possible datasets. Further, in the paper, we have shown results where we have considered only course-grained activities such as walking, running, head movement, etc. An interesting future work would be to evaluate Slim-Sense for fine-grained activities such as finger movement, breath rate monitoring, and so on to evaluate its effectiveness.

Usage of DeepRL: To solve the POMDP problem, we design hierarchical reinforcement learning. However, an alternate choice would have been to use deep-RL. We refrain from doing this in the paper as training a deep-RL model would increase the complexity and training time. Since Slim-Sense is designed to run on WiFi devices, the aim should be to minimize the complexity. However, it would be possible to design a lightweight deep-RL to be employed on WiFi devices. For example, [4, 11, 19] all modify original YoLo [40] a DNN to run object detection on smaller edge devices like Jetson Nano. On the same lines, FastDeepIoT [38] and DeepAdaptor [20] identify and then prune the nodes in the neural network to make it light enough for execution on embedded systems.

8 Conclusion

In this paper, we propose Slim-Sense that performs WiFi sensing utilizing the minimum possible spectrum resources with a minor impact on sensing performance compared to using complete spectrum resources. Such a technique enables smooth integration of sensing and communication by minimizing the impact of sensing on communication. We have showcased the performance of the Slim-Sense through four diverse datasets. The key feature of Slim-Sense is providing environment-independent and application-specific resources, which provides robust WiFi sensing across different environments. We showcase the adaptability of the Slim-Sense in a new or unseen environment. We highlight the generalizability of Slim-Sense in relation to existing sensing solutions and varied environments. We believe Slim-Sense will act as an enabler to support ISAC in future WiFi networks.

Acknowledgments

This work was funded by Department of Science and Technology via the Promotion of University Research and Scientific Excellence (DST-PURSE) Grant (Grant number: SR/PURSE/2023/172). The primary author also acknowledges financial support from the UGC NET Fellowship (Application number: SRF-200510308057), awarded by the University Grants Commission (UGC) of the Government of India.

References

[1] 11bf 2024. *IEEE P802.11 - TASK GROUP BF (WLAN SENSING).* Retrieved Apr 13, 2024 from https://www.ieee802.org/11/Reports/tgbf_update.htm

[2] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. 2015. UbiBreathe: A Ubiquitous non-Invasive WiFi-based Breathing Estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Hangzhou, China) *(MobiHoc '15)*. Association for Computing Machinery, New York, NY, USA, 277–286. doi:10.1145/2746285.2755969

[3] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 1472–1480. doi:10.1109/INFOCOM.2015.7218525

[4] Yuxuan Cai. 2020. *YOLObile: Real-time object detection on mobile devices via compression-compilation co-design.* Ph. D. Dissertation. Northeastern University.

[5] Zhijie Cai, Tingwei Chen, Fujia Zhou, Yuanhao Cui, Hang Li, Xiaoyang Li, Guangxu Zhu, and Qingjiang Shi. 2023. FallDeWideo: Vision-Aided Wireless Sensing Dataset for Fall Detection with Commodity Wi-Fi Devices. In *Proceedings of the 3rd ACM MobiCom Workshop on Integrated Sensing and Communications Systems* (Madrid, Spain) *(ISACom '23)*. Association for Computing Machinery, New York, NY, USA, 7–12. doi:10.1145/3615984.3616501

[6] Cheng Chen, Hao Song, Qinghua Li, Francesca Meneghello, Francesco Restuccia, and Carlos Cordeiro. 2022. Wi-Fi sensing based on IEEE 802.11 bf. *IEEE Communications Magazine* 61, 1 (2022), 121–127. https://doi.org/10.1109/MCOM.007.2200347

[7] Zhenghua Chen, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. 2019. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. *IEEE Transactions on Mobile Computing* 18, 11 (2019), 2714–2724. https://doi.org/10.1109/ICCT59356.2023.10419414

[8] Marco Cominelli, Francesco Gringoli, and Francesco Restuccia. 2023. Exposing the CSI: A Systematic Investigation of CSI-based Wi-Fi Sensing Capabilities and Limitations. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 81–90. https://doi.org/10.1109/PERCOM56429.2023.10099368

[9] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-net: a unified meta-learning framework for RF-enabled one-shot human activity recognition. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (Virtual Event, Japan) *(SenSys '20)*. Association for Computing Machinery, New York, NY, USA, 517–530. doi:10.1145/3384419.3430735

[10] Chao Feng, Nan Wang, Yicheng Jiang, Xia Zheng, Kang Li, Zheng Wang, and Xiaojiang Chen. 2022. Wi-Learner: Towards One-shot Learning for Cross-Domain Wi-Fi based Gesture Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 114 (sep 2022), 27 pages. doi:10.1145/3550318

[11] Prakhar Ganesh, Yao Chen, Yin Yang, Deming Chen, and Marianne Winslett. 2022. YOLO-ReT: Towards High Accuracy Real-time Object Detection on Edge GPUs. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. doi:10.1109/WACV51458.2022.00138

[12] Ruiyang Gao, Mi Zhang, Jie Zhang, Yang Li, Enze Yi, Dan Wu, Leye Wang, and Daqing Zhang. 2021. Towards position-independent sensing for gesture recognition with Wi-Fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–28. https://doi.org/10.1145/3463504

[13] Yu Gu, Jinhai Zhan, Yusheng Ji, Jie Li, Fuji Ren, and Shangbing Gao. 2017. MoSense: An RF-Based Motion Detection System via Off-the-Shelf WiFi Devices. *IEEE Internet of Things Journal* 4, 6 (2017), 2326–2341. doi:10.1109/JIOT.2017.2754578

[14] Khandaker Foysal Haque, Milin Zhang, and Francesco Restuccia. 2023. Simwisense: Simultaneous multi-subject activity classification through wi-fi signals. In *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 46–55. 10.1109/WoWMoM57956.2023.00019

[15] Yinghui He, Jianwei Liu, Mo Li, Guanding Yu, Jinsong Han, and Kui Ren. 2023. SenCom: Integrated Sensing and Communication with Practical WiFi. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–16. https://doi.org/10.1145/3570361.3613274

[16] Steven M. Hernandez and Eyuphan Bulut. 2020. Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility. In *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 277–286. https://doi.org/10.1109/WoWMoM49955.2020.00056

[17] Jingzhi Hu, Tianyue Zheng, Zhe Chen, Hongbo Wang, and Jun Luo. 2023. MUSE-Fi: Contactless MUti-person SEnsing Exploiting Near-field Wi-Fi Channel Variation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[18] Yuqian Hu, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. 2022. DeFall: Environment-Independent Passive Fall Detection Using WiFi. *IEEE Internet of Things Journal* 9, 11 (2022), 8515–8530. doi:10.1109/JIOT.2021.3116136

[19] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. 2018. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In *2018 IEEE International Conference on Big Data (Big Data)*. 2503–2510. doi:10.1109/BigData.2018.8621865

[20] Yakun Huang, Xiuquan Qiao, Jian Tang, Pei Ren, Ling Liu, Calton Pu, and Junliang Chen. 2020. DeepAdapter: A Collaborative Deep Learning Framework for the Mobile Web Using Context-Aware Network Pruning. In *IEEE Conference on Computer Communications*. doi:10.1109/INFOCOM41043.2020.9155379

[21] Sijie Ji, Yaxiong Xie, and Mo Li. 2023. SiFall: Practical Online Fall Detection with RF Sensing. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems* (, Boston, Massachusetts,) *(SenSys '22)*. Association for Computing Machinery, New York, NY, USA, 563–577. doi:10.1145/3560905.3568517

[22] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. 2021. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 286–293.

[23] Chenning Li, Manni Liu, and Zhichao Cao. 2020. WiHF: Enable User Identified Gesture Recognition with WiFi. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 586–595. doi:10.1109/INFOCOM41043.2020.9155539

[24] Heju Li, Xin He, Xukai Chen, Yinyin Fang, and Qun Fang. 2019. Wi-Motion: A Robust Human Activity Recognition Using WiFi Signals. *IEEE Access* 7 (2019), 153287–153299. doi:10.1109/ACCESS.2019.2948102

[25] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: Talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 250–261.

[26] Francesca Meneghello, Cheng Chen, Carlos Cordeiro, and Francesco Restuccia. 2023. IEEE 802.11AX CSI Dataset for Human Activity Recognition. doi:10.21227/7bd6-xs14

[27] Francesca Meneghello, Cheng Chen, Carlos Cordeiro, and Francesco Restuccia. 2023. Toward integrated sensing and communications in IEEE 802.11 bf Wi-Fi networks. *IEEE Communications Magazine* 61, 7 (2023), 128–133.

[28] Francesca Meneghello, Domenico Garlisi, Nicolò Dal Fabbro, Ilenia Tinnirello, and Michele Rossi. 2022. Sharp: Environment and person independent activity recognition with commodity ieee 802.11 access points. *IEEE Transactions on Mobile Computing* (2022).

[29] Francesco Restuccia. 2021. IEEE 802.11 bf: Toward ubiquitous Wi-Fi sensing. *arXiv preprint arXiv:2103.14918* (2021).

[30] Bidisha Sharma, Maulik Madhavi, and Haizhou Li. 2021. Leveraging acoustic and linguistic embeddings from pretrained speech and language models for intent classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7498–7502.

[31] Vijay Kumar Singh, Pragma Kar, Ayush Madhan Sohini, Madhav Rangaiah, Sandip Chakraborty, and Mukulika Maity. 2023. WiFiTuned: Monitoring Engagement in Online Participation by Harmonizing WiFi and Audio. In *Proceedings of the 25th International Conference on Multimodal Interaction*. 670–678.

[32] Sheng Tan, Jie Yang, and Yingying Chen. 2022. Enabling Fine-Grained Finger Gesture Recognition on Commodity WiFi Devices. *IEEE Transactions on Mobile Computing* 21, 8 (2022), 2789–2802. doi:10.1109/TMC.2020.3045635

[33] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human respiration detection with commodity wifi devices: do user location and body orientation matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 25–36. doi:10.1145/2971648.2971744

[34] Xuanzhi Wang, Kai Niu, Jie Xiong, Bochong Qian, Zhiyun Yao, Tairong Lou, and Daqing Zhang. 2022. Placement matters: Understanding the effects of device placement for WiFi sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–25.

[35] Chenhao Wu, Xuan Huang, Jun Huang, and Guoliang Xing. 2023. Enabling Ubiquitous WiFi Sensing with Beamforming Reports. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 20–32.

[36] Rui Xiao, Jianwei Liu, Jinsong Han, and Kui Ren. 2021. OneFi: One-Shot Recognition for Unseen Gesture via COTS WiFi. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (Coimbra, Portugal) *(SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 206–219. doi:10.1145/3485730.3485936

[37] Kun Yang, Xiaolong Zheng, Jie Xiong, Liang Liu, and Huadong Ma. 2022. WiImg: pushing the limit of WiFi sensing with low transmission rates. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[38] Shuochao Yao, Yiran Zhao, Huajie Shao, ShengZhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. 2018. FastDeepIoT: Towards Understanding and Optimizing Neural Network Execution Time on Mobile and Embedded Devices. In *16th ACM Conference on Embedded Networked Sensor Systems* (Shenzhen, China). 278–291. doi:10.1145/3274783.3274840

[39] Enze Yi, Dan Wu, Jie Xiong, Fusang Zhang, Kai Niu, Wenwei Li, and Daqing Zhang. 2024. {BFMSense}:{WiFi} Sensing Using Beamforming Feedback Matrix. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 1697–1712.

[40] yolov5 2022. *Ultralytics YOLOv5*. Retrieved 15 April 2024 from https://github.com/ultralytics/yolov5

[41] Feng Zhang, Chenshu Wu, Beibei Wang, Hung-Quoc Lai, Yi Han, and K. J. Ray Liu. 2019. WiDetect: Robust Motion Detection with a Statistical Electromagnetic Model. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 122 (sep 2019), 24 pages. doi:10.1145/3351280

[42] Xiaolong Zheng, Kun Yang, Jie Xiong, Liang Liu, and Huadong Ma. 2024. Pushing the Limits of WiFi Sensing with Low Transmission Rates. *IEEE Transactions on Mobile Computing* (2024).

[43] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-effort cross-domain gesture recognition with Wi-Fi. In *MobiSys 2019*. 313–325.