Energy-Efficient CSI-Based People Tracking with Spiking Neural Networks

Alisa Snezskaia
University of Padova
Padova, Italy
alisa.snezskaia@studenti.unipd.it

Elena Tonini
University of Brescia & CNIT
Brescia, Italy
elena.tonini@unibs.it

Renato Lo Cigno
University of Brescia & CNIT
Brescia, Italy
renato.locigno@unibs.it

Giovanni Perin
University of Brescia & CNIT
Brescia, Italy
giovanni.perin@unibs.it

Abstract

Device-free indoor human monitoring often leverages Wi-Fi channel state information (CSI), extracted from commodity access points. However, the deep neural networks (DNNs) typically used for CSI-based sensing require a computational capacity exceeding what embedded and internet of things (IoT) devices can sustain. Spiking neural networks (SNNs) are a promising alternative, being naturally suited to temporal signals whose informative content lies in changes between consecutive channel measurements. In this work, we formulate people tracking as a joint presence detection and position estimation problem and propose an SNN-based architecture that processes delta-encoded CSI amplitudes. The model is trained and evaluated on an IEEE 802.11ax CSI dataset spanning five activities in partial line of sight (PLoS) and non-line of sight (NLoS) conditions. The ground truth is obtained through position estimation on a synchronized video. The resulting pipeline *i)* distinguishes occupied from empty rooms, and *ii)* estimates the tracked subject's approximate position from CSI alone. This work suggests that SNNs are a low-power candidate for CSI-based sensing.

CCS Concepts

• **Computing methodologies** → **Bio-inspired approaches**; *Neural networks*; • **Networks** → **Network measurement**; Network privacy and anonymity; Wireless access networks.

Keywords

Channel state information, Spiking neural network, Wi-Fi, Joint communication and sensing, Indoor tracking

1 Introduction

Wi-Fi channel state information (CSI) analysis is one of the most investigated joint communication and sensing (JCAS) techniques. CSI provides a description of the wireless propagation channel, whose variations reflect changes in the surrounding environment. This enables applications ranging from human activity recognition (HAR) to occupancy detection, localization, and tracking using already-deployed communication hardware.

Most state-of-the-art CSI-based sensing tools rely on deep learning (DL) models to extract meaningful patterns from the high-dimensional temporal evolution of the channel. Albeit highly accurate, these approaches are computationally and energetically expensive, making them unsuitable for resource-constrained platforms where sensing applications are increasingly expected to operate. In these contexts, reducing resource requirements is as important as maximizing sensing accuracy.

Spiking neural networks (SNNs) offer a promising alternative: by communicating information through sparse events, they naturally reduce the number of operations performed during inference while preserving the ability to model temporal dynamics. Many CSI-based sensing tasks are well-suited to event-driven processing because their informative content is largely carried by temporal variations of the wireless channel. However, the use of SNNs for CSI-based sensing remains largely unexplored. Existing SNN-based Wi-Fi sensing studies primarily address window-level activity classification, leaving continuous position estimation from CSI largely unexplored. This is a more demanding setting, as the model must preserve fine-grained temporal information while producing a continuous estimate at every CSI frame.

In this paper, we investigate SNNs-based energy-efficient and device-free indoor human presence detection and tracking. The main contributions of this work are:

- A convolutional spiking neural network (Conv-SNN) architecture for joint occupancy detection and position estimation from Wi-Fi CSI;
- The validation of such architecture on an 802.11ax CSI dataset consisting of ∼ 200 minutes of channel monitoring in a real-world environment;
- The evaluation of the proposed Conv-SNN against two comparable convolutional and recurrent neural networks (NNs), showing that it matches or outperforms both, while using substantially less energy in the spiking trunk and remaining the most efficient end-to-end.

2 Related Work

2.1 CSI-based Indoor Sensing

Wi-Fi CSI is one of the most commonly used sources of information for JCAS. Its interpretation as an 'electromagnetic fingerprint' of the environment on the propagating signal offers an opportunity to exploit its information content for ambient sensing tasks. Among the most widespread applications, HAR and device-free localization and tracking are likely those that capture the most attention.

NN- or machine learning (ML)-based approaches are common in works aiming at environment-independent HAR. The authors

of [10] extract micro-Doppler traces from CSI data and feed them into an NN architecture to recognize dynamic activities, while [14] identifies a set of physically explainable features of the delay-Doppler domain representation of CSI to use for support vector machine (SVM)-powered HAR.

Many more studies, instead, rely on DL to extract distinctive information from the CSI. The authors of [15] devise a DL-based approach to differentiate the fingerprints associated with different user locations and reduce classification ambiguity; similarly, [17] proposes a CSI augmentation pipeline to enhance the distinguishability of the fingerprints used for localization. Models estimating the user velocity and location by integrating spatiotemporal Doppler information and angle of arrival of the received signal make indoor device-free user tracking possible [8]. Alternative approaches to this task involve leveraging the CSI difference for fine-grained Doppler speed estimation, resulting in a median tracking error as low as 34 cm [7].

For a broader overview of CSI-based JCAS applications, the interested reader can refer to the survey [11].

2.2 SNN for JCAS

Training SNNs on real-valued data has become more practical with the development of spatio-temporal and surrogate gradient-based methods, which enable standard backpropagation [12]. SNNs have already been applied to CSI sensing, mostly using the computational constraints of Wi-Fi hardware as central motivation. For instance, [9] and [18] propose spiking convolutional architectures for Wi-Fi activity recognition using CSI. Moving beyond CSI, [6] applies a spiking network with unsupervised spike-timing-dependent plasticity (STDP) to a related task, where Wi-Fi frames are detected from raw radio captures rather than channel state. CSI-based works typically frame their task as classification: a window of CSI samples is assigned just one activity label, not a continuous value. For this reason, [9] and [18] are not included as benchmarks in Section 6: both report classification accuracy only, not the presence-accuracy/RMSE metrics used here.

Continuous position estimation from CSI using SNNs remains largely unexplored, although the premises for it exist separately. One of the relevant examples is [1], where wearable sensor data are encoded by change rather than absolute value, and SNN energy is measured directly on neuromorphic hardware. The authors of [13] decode the continuous position and velocity from spiking activity, matching long short-term memory (LSTM) precision. Whether these ideas apply to CSI specifically is still an open question.

3 Principles of Wi-Fi CSI-based Sensing

The CSI is computed at the receiver from frame preambles as the ratio between the received signal and the expected one, providing an approximation of the channel frequency response $\mathbf{H}[f, t]$, such that

$$\hat{\mathbf{x}}[t] = \mathbf{H}[f, t]\,\mathbf{x}[t], \tag{1}$$

where $\mathbf{x}$ is the transmitted signal and $\hat{\mathbf{x}}$ is the received one. For orthogonal frequency division multiplexing (OFDM)-based technologies with N_k subcarriers and N_{SS} spatial streams, one CSI is extracted per spatial stream. The CSI is represented as an array of N_k complex numbers, with the k-th element corresponding to the in-phase and quadrature (IQ) components of the channel response associated with the k-th subcarrier. Some subcarriers are suppressed in transmission as they are used as guard bands; therefore, the CSI value is set to $0 + i0$ at the corresponding positions. The IQ complex values are commonly converted into the equivalent representation using amplitude and phase and undergo a normalization and cleaning process.

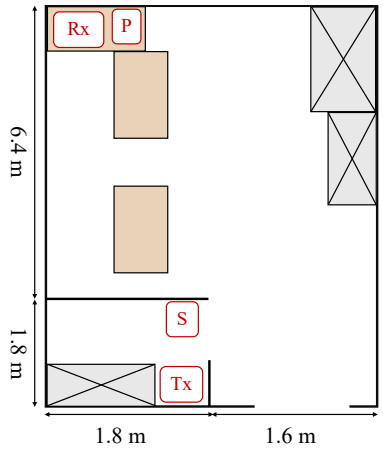

Figure 1: Floor plan of the experimental environment.

CSI Preprocessing. The CSI cleaning techniques used in this work aim at reconstructing both phase and amplitude on the suppressed subcarriers while ensuring a low computational complexity for the preprocessing step. In the IEEE 802.11ax standard, an 80 MHz channel consists of 1024 subcarriers: the external guard bands involve subcarriers $[-512, \ldots, -501]$ and $[+502, \ldots, +511]$, while the central null interval takes up subcarriers $[-2, \ldots, +2]$. The central nulls generate a discontinuity on the frequency axis, thus, their 'expected' amplitude and phase values need to be reconstructed by interpolation of the adjacent data subcarriers. On the contrary, the zero values in the external guard bands need no patching, since they do not create gaps in the frequency domain. Amplitude is also affected by the automatic gain control (AGC), which is compensated for by normalizing the energy of each CSI; finally, phase is aligned at zero on the central subcarrier.

4 Experimental Setting

The dataset considered in this study contains CSI samples collected across multiple days in the month of June 2025.[1] All experiments are thus subject to real-world variability due to minor modifications of the experimental environment from one day to the other; for instance, small pieces of furniture may be displaced during the interval between two measurement sessions. This property of the dataset is intentional and highlights the adaptability of the proposed sensing methodology to a variable and realistic input.

The selected indoor environment is an everyday office (Fig. 1), wherein the transmitter (Tx) and receiver (Rx) are placed in fixed positions at two opposite ends of the room. The Tx is placed on a 40 cm-high support behind a drywall partition, creating partial line of sight (PLoS) signal propagation conditions. The Rx is placed

[1]The dataset is available on Zenodo at https://zenodo.org/records/21335109.

on a table beside a mobile phone ('P' in Fig. 1) used to film the scene during the measurement sessions, providing a ground truth reference. When non-line of sight (NLoS) propagation conditions are tested, a metallic shield ('S' in Fig. 1) is placed in front of the Tx.

CSI samples are collected using the Nexmon CSI Extractor tool, which allows full customization of the experimental configuration of 802.11ax frames [4]. Tx and Rx communicate on the 80 MHz 802.11ax channel with center frequency 5.775 GHz. Both devices are commercial off-the-shelf (COTS) Asus access points (APs): the Tx is an RT-AX82U, while the Rx is an RT-AX86U; both are equipped with the Wi-Fi Broadcom chipset BCM43684. One antenna is active on the Tx, while three receiving antennas are enabled on the Rx. All devices, including the phone P, are synchronized via network time protocol (NTP) with millisecond precision, allowing precise alignment of CSI with video frames.

Hardware limitations impose a maximum sampling rate of 200 frames per second to ensure lossless CSI collection. To support future evaluation across variable frame rates, as is common in real traffic conditions, the dataset consists of CSI collections obtained with different sampling rates: artificial traffic is transmitted on the channel, with frames interspaced by $\Delta t = 5, 10, 50, 100$ ms. Specifically, $\Delta t = 100$ ms corresponds to the Wi-Fi beacons rate: The proposed approach is thus viable using frames that are always present where Wi-Fi is operational.

For each Δt value, five sets of experiments are performed, encompassing different combinations of activities: empty room (`E`), a person entering the empty room and sitting at the desk closer to the Rx (`EN-S`), a person entering the empty room and walking along a random trajectory (`EN-W`), a person sitting at the desk, then standing up and leaving the room (`L`), a person sitting at the desk (`S`).

5 The SNN Model

5.1 Problem Formulation

We cast people tracking as a joint presence detection and position regression problem. Given a window of T feature frames $F[1], \ldots, F[T]$ (Section 5.3), the two Conv-SNN outputs at every timestep t are a presence estimate $\hat{p}[t]$ and a position estimate $\hat{\mathbf{y}}[t]$, instead of a single label per window. The network jointly minimizes a presence classification loss and a position regression loss,

$$\min_{\mathbf{W}} \mathbb{E}\big[\, \mathcal{L}_{\text{presence}} + \mathcal{L}_{\text{position}} \,\big], \tag{2}$$

where $\mathcal{L}_{\text{presence}}$ penalizes the disagreement between $\hat{p}[t]$ and the ground-truth presence label of Section 5.2, and $\mathcal{L}_{\text{position}}$ penalizes the distance between $\hat{\mathbf{y}}[t]$ and the ground-truth position, evaluated only over frames where the subject is present. Training details are given in Section 5.4.

5.2 Ground-Truth Labels

Presence and position labels come from the synchronized room-camera video (Section 4), processed with MediaPipe's pose landmarker.[2] Shoulders are tracked because they stay visible even when the subject faces away. The position label $\mathbf{y}[t] = (x, y, z) \in \mathbb{R}^3$ combines MediaPipe's normalized image-plane tracking of the head/shoulder midpoint, $(x, y) \in [0, 1]^2$, with per-pixel metric depth from Depth-Anything-V2 [16] (metric-indoor checkpoint) sampled at the same point, independently min-max normalized to $[0, 1]$ over the whole dataset. While $\mathbf{y}[t]$ is not the real-world coordinate of the target, which is unavailable, we posit there exists a map G such that $\mathbf{y}'[t] = G(\mathbf{y}[t])$, where $\mathbf{y}'[t]$ is the real-world coordinate. We predict $\mathbf{y}[t]$ as a proxy to showcase the feasibility of retrieving $\mathbf{y}'[t]$.

Ground-truth postprocessing. The landmarker can fail to detect a pose, but labeling every such frame as "absent" would be wrong. Detection gaps at the beginning or end of a trial reflect real absence, since the person has either not yet entered or has already left; gaps surrounded by detections are treated as temporary tracking failures and labeled present. This distinction is the most impactful for the `L` and `S` activities, where the person sits still for long periods, and the detector is likely to lose its track. Short position gaps are filled with linear interpolation between the neighboring detections, while long gaps are left undefined and excluded, regardless of the presence label. The resulting trace is smoothed with a short moving average and resampled to the CSI frame rate, giving the per-timestep presence label and position target.

5.3 Feature Engineering

Let $h_{a,k}[t] \in \mathbb{C}$ denote the raw CSI value on receiving antenna $a \in \mathcal{N} = \{1, \ldots, N_a = 3\}$, subcarrier $k \in \mathcal{K} = \{1, \ldots, N_k = 1024\}$, at frame t. As stated in Section 4, there is only one transmitting antenna. All features are based on two primitives: amplitude $|h_{a,k}[t]|$, and phase $\angle h_{a,k}[t]$.

Amplitude. Amplitude is normalized per frame to unit energy across subcarriers,

$$\left|h^{\text{norm}}_{a,k}[t]\right| = \frac{|h_{a,k}[t]|}{\sqrt{\sum_{k'} |h_{a,k'}[t]|^2}}, \tag{3}$$

to make it robust to AGC variations across sessions.

Phase. All antennas share one local oscillator, so that it carries a common carrier/sampling-offset drift, which is not connected to motion. This drift cancels in the conjugate product between antenna a and a fixed reference antenna, chosen to be antenna 1:

$$u_{a,k}[t] = \frac{h_{a,k}[t]\, h^*_{1,k}[t]}{\left|h_{a,k}[t]\, h^*_{1,k}[t]\right|}, \qquad a \neq 1. \tag{4}$$

The full complex number $u_{a,k}[t]$ with unitary absolute value is retained as the phase $\angle u_{a,k}[t]$. This amounts to feeding the NN with $\text{Re}\{u_{a,k}[t]\}$ and $\text{Im}\{u_{a,k}[t]\}$ in two separate channels to avoid discontinuities coming from feeding $\angle u_{a,k}[t] \in (-\pi, \pi]$ directly, which would be misread as motion.

With $N_a = 3$, amplitude and phase direct processing gives $N_a = 3$ amplitude channels and $(N_a - 1) \times 2 = 4$ phase channels: 7 channels per subcarrier per frame, plus the 4 motion-aware scalar channels below (Eqs. (5) to (7) and (9)), broadcast across all subcarriers.

Motion magnitude. Presence and position depend on the channel's *change* rather than its absolute state. We refer to this change as *motion magnitude*, though it is a property of the CSI signal, not a direct measurement of the subject's physical motion. The average

[2] https://ai.google.dev/edge/mediapipe

frame-to-frame amplitude change across antennas and subcarriers,

$$m[t] = \frac{1}{N_a N_k} \sum_{a,k} \big|\, |h_{a,k}[t]| - |h_{a,k}[t-1]| \,\big|, \tag{5}$$

is broadcast as a single additional channel across all subcarriers. We also consider a normalized variant, dividing $m[t]$ by the median of m over a trailing 20-frame $\approx$ 1 s window,

$$\tilde{m}[t] = \frac{m[t]}{\mathrm{median}\big(m[t-w+1], \ldots, m[t]\big) + \varepsilon}, \tag{6}$$

since $m[t]$ inherits the amplitude's sensitivity to AGC gain drift. Normalizing against a local median keeps $\tilde{m}[t]$ well-behaved as gain drifts within a trial, without being distorted by isolated motion spikes inside the window itself.

Spectral motion ratio. Motion magnitude metrics presented above alone cannot separate genuine motion from a noisier sensor, since both increase $m[t]$. The two differ in *how* the signal varies over time, not *how much*: human motion is rhythmic and concentrated at low frequencies (footsteps at 1–2 Hz; sway even lower), while sensor and AGC noise is approximately white, spread evenly across frequencies regardless of its overall level. We exploit this by comparing energy in a low, motion-associated frequency band against a high, noise-reference band, computed per frame from the power spectrum of the antenna- and subcarrier-averaged amplitude over a trailing, Hann-windowed 3.2 s segment at the pipeline's 20 Hz frame rate,

$$r[t] = \log_{10}\!\left(\frac{P_{[0.15,3]\,\mathrm{Hz}}[t]}{P_{[5,10]\,\mathrm{Hz}}[t]} + \varepsilon\right). \tag{7}$$

Unlike $m[t]$, a rise in the noise floor raises both bands together and leaves $r[t]$ largely unaffected; the logarithm keeps this otherwise wide-ranging ratio on a scale comparable to the pipeline's other, roughly unit-scale channels.

Cross-antenna coherence. The spectral ratio characterizes a single averaged signal. A complementary criterion is the agreement between independent antennas: a body in the room reflects the same physical wavefront to all three antennas simultaneously, so genuine motion produces correlated fluctuations across antennas at shared frequencies, whereas each antenna's thermal and hardware noise is independent. We measure this with the magnitude-squared coherence between antennas. Let $S_a[t, f]$ denote the short-time spectrum of antenna a's amplitude trace at frequency f, computed over a 6.4 s window. A coherence estimate from a single spectral snapshot is trivially 1 by construction, therefore S_a is computed via Welch's method over $N_{\mathrm{sub}} = 7$ Hann-windowed sub-segments of 1.6 s each, at the standard 50% overlap, so that a coherence estimate that can meaningfully fall below 1 is obtained. Coherence between antennas a, b is

$$\gamma^2_{ab}[t] = \left\langle \frac{\big|\langle S_a S_b^* \rangle\big|^2}{\langle |S_a|^2 \rangle \langle |S_b|^2 \rangle} \right\rangle_{\mathrm{band}}, \tag{8}$$

where $\langle\cdot\rangle$ averages over sub-segments and $\langle\cdot\rangle_{\mathrm{band}}$ over the motion frequency band. We take, as the coherence feature, the average over all antenna pairs

$$\bar{\gamma}^2[t] = \frac{2}{N_a(N_a-1)} \sum_{\substack{a,b \in \mathcal{N} \\ a<b}} \gamma^2_{ab}[t]. \tag{9}$$

High coherence $\bar{\gamma}^2[t]$ in the motion band indicates a shared physical event; low coherence indicates uncorrelated per-antenna noise, regardless of its magnitude.

Motion magnitude, spectral ratio, and coherence are each a single scalar per frame, broadcast uniformly across all 1024 subcarriers.

5.4 Network Architecture

The feature stack

$$F[t] = \Big\{ m[t], \tilde{m}[t], r[t], \bar{\gamma}^2[t], \big|h^{\mathrm{norm}}_{a,k}[t]\big|_{a=1,\ldots,N_a}, \\ \big[\mathrm{Re}\{u_{a,k}[t]\}, \mathrm{Im}\{u_{a,k}[t]\}\big]_{a=2,\ldots,N_a} \Big\} \tag{10}$$

is the full candidate feature set; Section 6.1 identifies the subset retained for all results in Section 6. It is segmented into sliding windows of $T = 64$ frames with stride 32, 50% overlap – 3.2 s clips at the 50 ms capture rate; then, it is passed through a per-frame 1D convolutional encoder, two layers of 16 and 32 channels, kernel size 9, each followed by rectified linear unit (ReLU) and 2× max-pooling, applied independently to every frame's (channel × subcarrier) slice.

The resulting compact per-frame feature vector $\mathbf{f}[t]$ is converted into a binary spike train through threshold-crossing (delta) encoding,

$$s_i[t] = \begin{cases} 1, & f_i[t] - f_i[t-1] \geq \tau \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

with $\tau = 0.3$. The spike train $s[t]$ feeds a two-layer leaky integrate-and-fire (LIF) trunk [3], shared between both output heads, with membrane decay $\alpha = 0.9$ and firing threshold $\vartheta = 1$, trained end-to-end via a fast-sigmoid surrogate gradient. The two LIF layers have size 128 and 32, respectively, producing spike outputs $s^{(1)}[t]$ and $s^{(2)}[t]$. Presence and position are read out directly from the second LIF layer's *spike* output $s^{(2)}[t]$ at every timestep t through the heads

$$\hat{p}[t] = \mathbf{W}_{\mathbf{p}}\, s^{(2)}[t] + b_p, \tag{12}$$

$$\hat{\mathbf{y}}[t] = \mathrm{MLP}_y\big(s^{(2)}[t]\big), \tag{13}$$

where $\hat{p}[t]$ is a presence logit obtained via a linear layer (learned parameters $\mathbf{W}_{\mathbf{p}}$ and b_p), $\hat{\mathbf{y}}[t] \in \mathbb{R}^3$ is an (x, y, z) position estimate, and MLP_y is a single hidden layer of width 16 with a ReLU nonlinearity (multi-layer perceptron (MLP)). Both heads operate on the same binary, 32-dimensional code at each 50 ms frame, rather than a membrane potential accumulated over the whole window.

The network is trained end-to-end on the objective of Eq. (2) using Adam with learning rate $5{\times}10^{-4}$ and batch size 32 for a fixed 60 epochs per fold; the checkpoint with the highest validation accuracy is retained for testing.

Model performance is evaluated via leave-activity-out cross-validation. Of the five activity codes, only the four with a person present (`EN-S`, `EN-W`, `L`, `S`) are ever held out; the fifth, the empty-room code `E`, stays in every fold's training set so the model always sees genuine absent-class examples. Each of the four folds holds out one activity's two sessions entirely as the test set, uses a *different* activity's two sessions for validation (checkpoint selection and presence-threshold calibration), and trains on the remaining six sessions.

Table 1: Comparison of the best model against two benchmarks. Energy is reported for the trunk alone and for the full pipeline (encoder + trunk).

Model	Presence Acc.	Tracking RMSE	Energy (trunk) [µJ]	Energy (full) [µJ]
Conv-SNN	**0.751 ± 0.111**	**0.374 ± 0.093**	**4.52**	**1.05×10^3**
Conv-ANN	0.587 ± 0.159	0.873 ± 0.435	3.10×10^2	1.35×10^3
LSTM	0.576 ± 0.117	0.404 ± 0.093	1.26×10^3	2.30×10^3

6 Results

We compare the Conv-SNN model (Section 5.4) against two non-spiking benchmarks trained on the identical feature set and evaluated under the same leave-activity-out 4-fold cross-validation of Section 5.4: an artificial neural network (ANN) and an LSTM, both built on the same convolutional encoder and trunk sizes as the Conv-SNN. We compare the three models with the following metrics.

Presence accuracy. The share of correctly predicted $\hat{p}[t]$.

Root mean squared error (RMSE). Tracking RMSE is

$$\mathrm{RMSE}(\hat{\mathbf{y}}[t], \mathbf{y}[t]) = \sqrt{\mathbb{E}[\|\hat{\mathbf{y}}[t] - \mathbf{y}[t]\|_2^2]}, \tag{14}$$

averaged over present-labeled frames. Since $\mathbf{y}[t]$ is expressed on the normalized $[0, 1]$ scale of Section 5.2, a random guess corresponds to an average value of $\sqrt{3}/2 \approx 0.866$.

Energy consumption. Energy follows the standard multiply-accumulate/accumulate-only model (45 nm CMOS) [5]: 4.6 pJ per dense operation (ANN/LSTM, and the convolutional encoder, shared and always dense) versus 0.9 pJ per spike-gated operation (SNN trunk), charged only on a spike where gating applies. We report both full-pipeline and trunk-only energy, respectively including and excluding the encoder to isolate activation sparsity.

Table 1 summarizes our main results. The Conv-SNN leads in presence accuracy at 0.751, 0.164 points ahead of the convolutional artificial neural network (Conv-ANN), the closest of the two benchmarks, which only achieves 0.587 accuracy. The Conv-SNN also gives the lowest RMSE of the three models at 0.374. The Conv-ANN's high RMSE average and variance at 0.873 ± 0.435, on the other hand, indicate less stable fold-to-fold behavior than either recurrent model, making the output close to a random guess. The validation RMSE is shown in Fig. 2 for the three ML models.

At the trunk level, Conv-SNN uses ∼ 69× less energy per window than Conv-ANN and ∼ 279× less than LSTM, as shown in Table 1. Conv-SNN and Conv-ANN share an identical architecture with 1 059 252 parameters each, differing only in spiking (LIF) vs. dense (ReLU) activation, so the gap isolates activation sparsity as the cause. LSTM's larger parameter count from its four gated weight matrices independently explains its higher energy cost.

Including the shared, always-dense convolutional encoder — which amounts to 99.6% of Conv-SNN's total — narrows the full-pipeline advantage to 1.29× and 2.20× respectively, though Conv-SNN still saves 22.6% and 54.5% in absolute energy per window.

6.1 Feature Ablation

Table 2 isolates the contribution of each engineered feature of Section 5.3: the top block adds one feature at a time to the amplitude + phase (AP) baseline, never combining with other features. The bottom block tests whether these separate features actually hold up once combined.

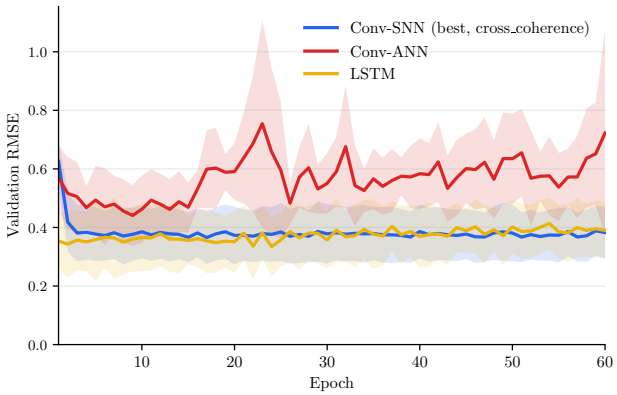

Figure 2: Validation RMSE across training epochs for the Conv-SNN, Conv-ANN, and LSTM.

Table 2: Feature ablation: amplitude and phase (AP), motion magnitude (MM), normalized motion magnitude (NMM), spectral ratio (SR), and cross-coherence (CC).

Feature	Presence Acc.	RMSE
AP	0.570 ± 0.053	0.391 ± 0.072
AP + MM ($m[t]$)	0.672 ± 0.110	0.349 ± 0.084
AP + NMM ($\tilde{m}[t]$)	0.667 ± 0.032	0.382 ± 0.092
AP + SR	0.613 ± 0.060	0.417 ± 0.094
AP + CC	**0.751 ± 0.111**	**0.374 ± 0.093**
AP + CC + NMM	0.708 ± 0.074	0.439 ± 0.203
AP + all four	0.625 ± 0.083	0.341 ± 0.076

AP + CC attains the highest presence accuracy of 0.751 ± 0.111 and is the configuration used for the Conv-SNN row of Table 1. The other three features, $m[t]$, $\tilde{m}[t]$, and $r[t]$ — corresponding to Equations (5) to (7) — all derive from the same subcarrier-averaged amplitude fluctuation and go flat once the subject stops moving. Cross-coherence instead measures whether fluctuations are shared across antennas: the residual motion of a seated subject perturbs all three antennas coherently even when its magnitude is negligible, which is why $\bar{\gamma}^2[t]$ of Equation (9) remains informative where the amplitude-derived features do not. AP + CC's RMSE of 0.374 ± 0.093 is not the lowest: AP + MM reaches 0.349 ± 0.084. However, the two scores are within one standard deviation of each other under our four-fold leave-activity-out cross-validation (CV). We therefore treat the accuracy gain, which exceeds any single fold's deviation, as the reliable finding.

AP + CC + NMM underperforms AP + CC alone on both metrics, with RMSE variance nearly tripling, though the full combination (AP + all four) instead reaches the best RMSE in the table. Since $m[t]$, $\tilde{m}[t]$, and $r[t]$ share a common origin, stacking any of them onto CC adds a channel correlated with information CC already captures indirectly; with only six training sessions per fold, it can inflate overfitting rather than add signal. SR is the weakest individual addition, with the worst RMSE of 0.417±0.094 and smallest accuracy gain over AP; this is likely due to $r[t]$ being a single scalar, averaged

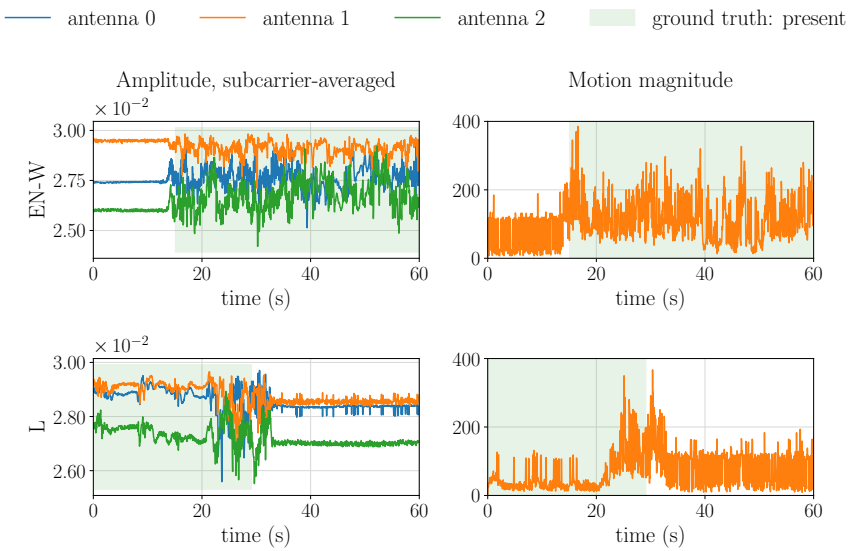

Figure 3: Amplitude and motion magnitude $m[t]$, as defined in Equation (5), in PLoS. For `EN-W` (top) there is continuous motion, for `L` (bottom) motion drops to near-zero while seated, despite continued presence.

over 3.2 s and across every antenna and subcarrier, which is too coarse for position estimation.

`L` (sits, then stands and leaves) is the clearest case, since during the sitting, motion-based features go quiet even though ground truth still labels the subject present, which is the same presence-detection problem handled by the gap-aware relabeling of Section 5.2. Figure 3 contrasts this against `EN-W`, where motion is continuous.

7 Conclusion

This paper presented an energy-efficient approach to device-free indoor people tracking through Wi-Fi CSI using SNN. Specifically, we stacked a spiking LIF trunk onto a dense convolutional encoder processing the CSI. The evaluation on an IEEE 802.11ax dataset collected under both PLoS and NLoS conditions showed that the Conv-SNN outperforms equivalent non-spiking benchmarks on both presence accuracy and position RMSE, while reducing spiking-trunk energy by up to two orders of magnitude, which translates to a 1.3–2.2× end-to-end saving once the shared dense encoder is included. To preserve the full energy gain, future work includes designing a spiking encoder [2], thereby eliminating the need to resort to dense networks across the full pipeline. Feature ablation showed that cross-antenna coherence contributed the largest gain, supporting its utility for disambiguating motion from noise. The results suggest that SNNs are a low-power candidate for CSI-based sensing in resource-constrained devices.

References

[1] Sizhen Bian and Michele Magno. 2023. Evaluating Spiking Neural Network on Neuromorphic Platform for Human Activity Recognition. In *Proc. ACM ISWC*. ACM, New York, NY, USA, 82–86.

[2] Eleonora Cicciarella, Riccardo Mazzieri, Jacopo Pegoraro, and Michele Rossi. 2026. Sparse Spike Encoding of Channel Responses for Energy Efficient Human Activity Recognition. In *Proc. of 2026 IEEE ICC*. IEEE, Glasgow, UK, 1–6.

[3] Jason K. Eshraghian et al. 2023. Training Spiking Neural Networks Using Lessons From Deep Learning. *Proc. of the IEEE* 111, 9 (2023), 1016–1054.

[4] Francesco Gringoli, Marco Cominelli, Alejandro Blanco, and Joerg Widmer. 2021. AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms. In *Proc. ACM WiNTECH*. ACM, New Orleans (USA), 46–53.

[5] Mark Horowitz. 2014. Computing's Energy Problem (and What We Can Do About It). In *Proc. of 2014 IEEE ISSCC*. IEEE, San Francisco, CA, USA, 10–14.

[6] Hyun-Jong Lee, Dong-Hoon Kim, and Jae-Han Lim. 2023. Wi-Fi Frame Detection via Spiking Neural Networks with Memristive Synapses. *Comput. Commun.* 208 (2023), 256–270.

[7] Wenwei Li et al. 2024. WiFi-CSI Difference Paradigm: Achieving Efficient Doppler Speed Estimation for Passive Tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 63 (5 2024), 29 pages.

[8] Xiang Li et al. 2017. IndoTrack: Device-Free Indoor Human Tracking with Commodity Wi-Fi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 72 (Sept. 2017), 22 pages.

[9] Yisha Lu, Liwen Jing, Jiangmao Zheng, and Bowen Zhang. 2026. Spiking-Aided Neural Architecture for Efficient and Robust WiFi Sensing. In *Proc. AAAI Conf. Artif. Intell.* AAAI Press, Seoul (South Korea), 24106–24114.

[10] Francesca Meneghello et al. 2023. SHARP: Environment and Person Independent Activity Recognition With Commodity IEEE 802.11 Access Points. *IEEE Trans. Mob. Comput.* 22, 10 (10 2023), 6160–6175.

[11] Fucheng Miao et al. 2024. Wi-Fi Sensing Techniques for Human Activity Recognition: Brief Survey, Potential Challenges, and Research Directions. *ACM Comput. Surv.* 57, 5 (11 2024), 1–30.

[12] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Process. Mag.* 36, 6 (2019), 51–63.

[13] Elijah A. Taeckens and Sahil Shah. 2023. A Spiking Neural Network with Continuous Local Learning for Robust Online Brain Machine Interface. *J. Neural Eng.* 20, 6 (2023), 066042.

[14] Elena Tonini, Renato Lo Cigno, and Emanuele Viterbo. 2026. Delay-Doppler Domain Feature Extraction for Explainable Wi-Fi Activity Recognition via SVM. In *Proc. of 2026 Medit. Artif. Intell. and Netw. Conf. (MAIN)*. IEEE, Mondello (Palermo), Italy, 1–9.

[15] Jianqiang Xue, Jie Zhang, Zhenyue Gao, and Wendong Xiao. 2023. Enhanced WiFi CSI Fingerprints for Device-Free Localization With Deep Learning Representations. *IEEE Sens. J.* 23, 3 (2023), 2750–2759.

[16] Lihe Yang et al. 2024. Depth Anything V2. In *Adv. Neural Inf. Process. Syst.*, Vol. 37. Curran Associates, Inc., Vancouver (Canada), 21875–21911.

[17] Jie Zhang, Jianqiang Xue, Yanjiao Li, and Simon L. Cotton. 2025. Leveraging Online Learning for Domain-Adaptation in Wi-Fi-based Device-Free Localization. *IEEE Trans. on Mob. Comput.* 24, 8 (2025), 7773–7787.

[18] Nengbo Zhang et al. 2026. Wi-Spike: A Low-power WiFi Human Multi-action Recognition Model with Spiking Neural Networks. arXiv:2603.14475 [cs.NI]