AmbientFM: A Foundation Model for Ambient Sensing

Guozhen Zhu[1] **Yuqian Hu**[1] **Sakila S Jayaweera**[1] **Wei-Hsiang Wang**[1] **Weihang Gao**[1]
Jiaxuan Zhang[1] **Beibei Wang**[1] **Chenshu Wu**[2] **K. J. Ray Liu**[1]

[1]ADT Research [2]The University of Hong Kong

Abstract

Ambient intelligence seeks to continuously understand human presence, activity, and physiology in physical spaces, enabling smart environments, health monitoring, and human-computer interaction. WiFi infrastructure offers a ubiquitous, always-on, and privacy-preserving sensing substrate across billions of IoT devices. Yet ambient sensing with WiFi remains largely fragmented, with most systems relying on task-specific models, labeled data collection, and customized training pipelines.

We present AmbientFM, a foundation model for ambient sensing through ubiquitous WiFi signals. AmbientFM is pre-trained on 9.2 million unlabeled Channel State Information (CSI) samples collected over 439 days from 20 commercial device types deployed in real homes, the largest real-world WiFi CSI pre-training corpus reported to date. It learns transferable wireless representations through contrastive learning, masked reconstruction, and physics-informed objectives tailored to wireless signals. With lightweight adaptation, a single backbone transfers to 9 downstream tasks, achieving above 0.90 AUROC on all classification benchmarks and supporting dense spatial reconstruction. AmbientFM reduces the need for labeled data and task-specific design, enabling scalable ambient intelligence on existing wireless infrastructure. *Project Page.*

1 Introduction

Ambient intelligence that continuously senses and responds to its occupants underpins applications from smart home automation to elderly care and occupational health monitoring. Realizing this vision requires perception that is always-on, unobtrusive, and deployable at scale. Ambient WiFi signals offer a compelling substrate: they propagate through walls, function in any lighting condition, require no worn devices, and preserve visual privacy [35]. The billions of access points, smart speakers, and IoT devices already deployed continuously transmit signals whose channel state information (CSI) captures human motion patterns and subtle physiological signals such as respiration, enabling ambient perception at unprecedented scale without additional hardware.

Despite this potential, ambient sensing research has predominantly relied on task-specific models trained from scratch. Each new task requires custom architectures, large-scale labeled data collection, and specialized training, and knowledge cannot be shared across the diverse sensing capabilities that WiFi enables [30]. Foundation models have shown that large-scale self-supervised pre-training produces general-purpose representations that transfer across tasks and reduce labeling requirements in vision [9, 22], speech [4], and inertial sensing [37]. WiFi is a natural candidate for this paradigm: commodity hardware already generates massive unlabeled CSI streams, and a single backbone could serve applications from room-level occupancy to fine-grained gesture recognition.

However, WiFi is a communication system repurposed for sensing, and this origin introduces challenges that are qualitatively more severe than those in purpose-built sensing modalities. WiFi

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

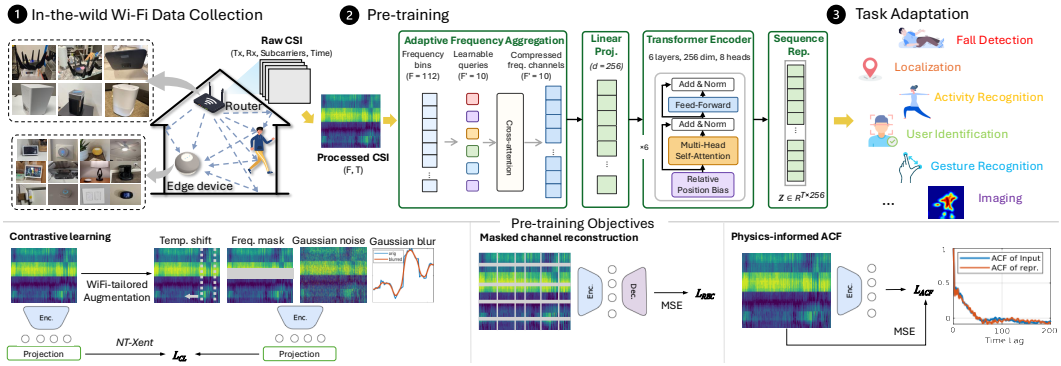


Figure 1: **AmbientFM overview.** AmbientFM pre-trains a shared sensing backbone on in-the-wild CSI with complementary self-supervised objectives, then adapts it to diverse downstream tasks with lightweight tuning.

devices dynamically change their bandwidth, frequency band, and antenna configuration at runtime, producing structurally incompatible observations rather than simple calibration differences. The MAC-layer protocol introduces non-deterministic packet timing and abrupt amplitude shifts, while omnidirectional antennas entangle human motion with environment-specific multipath structure that typically dominates the observed signal. These properties render standard self-supervised objectives insufficient, necessitating methods tailored to wireless signals.

In this paper, we present AmbientFM[1], a foundation model for WiFi-based ambient intelligence, pre-trained at scale on in-the-wild unlabeled CSI data and designed for efficient adaptation across tasks. It is pretrained on 9.2M unlabeled CSI samples collected over 439 days from 20 device types across 11 environments, to our knowledge the largest real-world corpus used for WiFi CSI pre-training. Its self-supervised framework combines contrastive learning, masked reconstruction, and a physics-informed autocorrelation objective. The architecture incorporates adaptive frequency aggregation to handle heterogeneous subcarrier quality and relative temporal encoding for translation-invariant activity patterns, and is evaluated across 9 downstream tasks via parameter-efficient adaptation.

Our key contributions include: 1) **Developing a foundation model for ambient intelligence**: We outline the design and development of a large-scale foundation model for ambient sensing based on WiFi, providing a single backbone for diverse ambient-sensing tasks. 2) **Domain-specific pre-training with large-scale real-world dataset**: We show that with a large-scale unlabeled CSI dataset, a self-supervised learning framework tailored to the characteristics of wireless signals can extract representations for practical ambient perception. 3) **Strong performance of single backbone on diverse tasks with superior data efficiency**: We demonstrate AmbientFM's generalizability and real-world utility across 9 diverse WiFi sensing tasks from public benchmarks spanning physiological sensing, activity understanding, spatial reasoning, biometrics, fine-grained motion, and spatial reconstruction. AmbientFM achieves strong performance with lightweight adaptation and reduces the need for task-specific data and design, offering a practical path toward scalable, privacy-preserving ambient intelligence on existing wireless infrastructure.

2 Related Work

Foundation models for perception. Foundation models have transformed representation learning by enabling transfer from large-scale unlabeled data. In vision, masked autoencoders and contrastive learning produce transferable representations under limited supervision [9, 22, 6, 10]. This paradigm extends to other sensing modalities: wav2vec 2.0 improves sample efficiency in speech recognition [4], and pre-trained models for inertial sensing demonstrate cross-user and cross-device transfer [37]. A consistent finding is that foundation models reduce labeling requirements while enabling knowledge sharing across tasks. For WiFi-based ambient sensing, this paradigm has only recently begun to be explored, as discussed below.

[1]Project webpage: `https://ai-sensing.github.io/ambient-fm/index.html`

Table 1: Comparison with related self-supervised and foundation-model work on wireless signals.

Method	Application domain	Foundation model	Pre-training scale	Real/Sim.	#Tasks
SSL-HAR [38]	Sensing	No	$\leq$45K / setting	Real	1
CIG-MAE [16]	Sensing	No	2.2K–3.8K / dataset	Real	1
CAPC [5]	Sensing	No	5.5K	Real	2
CSI-MAE [13]	Wireless channel modeling	Yes	1.45M CSI	Sim.	3
WiFo-CF [17]	Wireless communication	Yes	160K CSI	Sim.	2
AmbientFM	Ambient sensing with WiFi	Yes	9.2M / 439 days	Real	9

WiFi sensing capabilities and limitations. WiFi CSI captures how radio signals propagate through indoor environments, enabling diverse applications including activity recognition [32, 21], fall detection [33, 24], gesture detection [2, 43], respiration monitoring [1, 8], and indoor localization [15, 31]. WiFi signals propagate through walls, require no wearable devices, and leverage billions of already deployed access points. However, current research relies on task-specific models trained from scratch [7, 45, 31], with each new task requiring custom architectures and labeled data. Prior efforts in domain adaptation [42], adversarial learning [34], and physics-driven feature engineering [44] improve transferability for individual tasks but do not provide general-purpose representations. Whether a single pre-trained model can support multiple WiFi sensing tasks remains largely unexplored.

Self-supervised and foundation-model approaches for wireless signals. SSL has improved sample efficiency in WiFi sensing [23], but existing methods pre-train on small, task-specific datasets and evaluate one or two tasks, mostly activity recognition [38, 16, 5]. Wireless foundation models such as WiFo-CF [17] and CSI-MAE [13] pre-train on simulated channels for communication-layer problems, including CSI feedback and channel modeling, whose objectives and outputs differ from human-centric sensing. Concurrent work has also explored predictive pre-training for WiFi sensing [18, 14]. AmbientFM differs from these efforts in pre-training a task-agnostic backbone on large-scale real-world CSI from heterogeneous commodity devices and in evaluating cross-task transfer across nine ambient-sensing tasks (Table 1).

3 Data

3.1 Pre-training Dataset

We collect a large-scale in-the-wild WiFi CSI corpus for self-supervised pre-training, comprising 439+ days of recording, $\sim$20 TB of data, and 9.2 million CSI samples across 20 commercial IoT device types, 33 physical devices, 11 real-world environments, and 26 participants. To our knowledge, this is the largest and most diverse WiFi sensing dataset reported to date for foundation model pre-training.

The corpus spans substantial hardware and deployment diversity, including 8 chipset families, 1×1 to 2×2 MIMO, 20/40/80 MHz bandwidths, and both 2.4 and 5 GHz bands, with more than 20 dB variation in received power across devices (Appendix B.2). The environments include studios, apartments, houses, and townhouses with diverse layouts, scales, and interference conditions (Appendix B.3). CSI is recorded continuously during natural daily activities with 6–9 concurrent device links per environment, capturing realistic motion, occlusion, multipath, and interference beyond lab settings. Appendix A, Table 3 compares the corpus with representative public WiFi sensing datasets.

3.2 WiFi CSI: Structure and Challenges

CSI describes how wireless signals propagate between a transmitter and a receiver by measuring the amplitude and phase responses of frequency subcarriers. When people or objects move in the environment, they perturb these channel responses, allowing CSI to support sensing applications such as activity recognition, localization, and physiological monitoring [19].

CSI is a rich sensing signal, but its structure differs from measurement produced by dedicated sensors. Because WiFi is designed primarily for communication, devices may adapt their bandwidth, frequency band, antenna configuration, and transmission power at runtime, sometimes even within a single session due to communication demands, making the signal structure like subcarriers inherently

unpredictable for sensing model [49]. The MAC-layer protocol also introduces irregular packet timing and transmission behavior optimized for communication. Automatic Gain Control can further introduce abrupt amplitude changes. In addition, commodity WiFi uses omnidirectional antennas, so motion-induced changes are observed together with environment-specific multipath, whose static components can be 10–100$\times$ stronger than the motion perturbations [11]. These properties make CSI highly informative, but they also require representations that can handle hardware diversity, protocol effects, and environment-dependent propagation.

These signal properties directly guide AmbientFM's design. First, multipath propagation creates non-local and configuration-dependent correlations across distant subcarriers, which vary with bandwidth and band selection [3]. To model this structure, AmbientFM uses adaptive frequency aggregation to compress heterogeneous subcarrier inputs into a unified representation. Second, subcarrier quality varies across hardware, packet timing, and frequency-selective fading [41, 3]. AmbientFM addresses this with learned subcarrier weighting and masked reconstruction, which exposes the model to packet loss and frequency-selective corruption during pre-training. Third, human activities produce characteristic temporal patterns and these patterns are embedded in measurements dominated by static multipath, AmbientFM uses relative temporal encoding and a physics-informed Autocorrelation Function (ACF) objective to emphasize motion-induced temporal structure while reducing sensitivity to static environmental effects [40, 39, 27, 26].

3.3 Data Preprocessing

Raw CSI is parsed into a complex-valued tensor $\mathbf{H} \in \mathbb{C}^{N_{\text{tx}} \times N_{\text{rx}} \times N_{\text{sub}} \times T}$, where $N_{\text{tx}}, N_{\text{rx}}$ denote antenna counts, N_{sub} is the number of subcarriers (varying with channel bandwidth), and T is temporal length. We flatten spatial–frequency dimensions to $\mathbf{X} \in \mathbb{C}^{F \times T}$ with $F = N_{\text{tx}} N_{\text{rx}} N_{\text{sub}}$. CSI streams are collected at a target rate of 100 Hz; we estimate effective rates from inter-packet timestamps and discard streams below two-thirds of the target rate.

We extract CSI amplitude $\mathbf{A} = |\mathbf{X}|$, which is robust to the phase noise common in commodity hardware [36]; the eight chipset families in our corpus also do not share a common phase reference. We segment continuous streams into non-overlapping fixed-length windows, producing segments $\mathbf{S}_i \in \mathbb{R}^{T \times F}$ with $T = 500$, i.e., 5 s at the 100 Hz target rate. All segments are zero-padded to a unified dimension $F_{\max}$ defined by the maximum feature size across devices, normalized to $[0, 1]$ via per-segment min–max normalization, and stored in compressed HDF5 format. No task-driven segmentation or filtering is applied.

4 Foundation Model Development

We present a self-supervised framework for learning transferable WiFi CSI representations. Our design choices are guided by the structural properties of CSI identified in Section 3.2, with the goal of learning representations that generalize across a wide variety of sensing tasks.

4.1 Architecture

The encoder maps raw CSI segments into compact representations using two components: adaptive frequency aggregation and relative temporal encoding.

To address variation in subcarrier quality across devices and environments, we use cross-attention to compress F input subcarriers into $F' = 10$ latent frequency channels:

$$\mathbf{z}_t = \sum_{f=1}^{F} \alpha_{tf} \left(\mathbf{W}_v \mathbf{x}_{tf} + \mathbf{p}_f \right), \quad \alpha_{tf} = \text{softmax}_f \left(\frac{\mathbf{q}^\top (\mathbf{W}_k \mathbf{x}_{tf} + \mathbf{p}_f)}{\sqrt{d_q}} \right), \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^{F' \times d_q}$ are learnable queries, d_q is the query dimension, and $\mathbf{p}_f$ denotes sinusoidal frequency encodings. This module emphasizes informative subcarriers while compressing the frequency dimension by 5–11$\times$.

To capture activity patterns independent of their absolute location in the input window, we use relative position bias in self-attention:

$$\text{Attn}_{ij} = \text{softmax}\left(\frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d_k}} + b_{ij} \right) \mathbf{v}_j, \quad b_{ij} = f_\theta(j - i), \tag{2}$$

where f_θ learns embeddings over relative temporal offsets. This encourages similar representations for the same activity pattern regardless of when it occurs.

The encoder consists of adaptive frequency aggregation followed by a 6-layer Transformer with hidden dimension $d = 256$ and 8 attention heads. A 3-layer decoder is used for reconstruction during pre-training. We refer to this as the Base model (5M parameters), and also study Small (2M) and Large (12M) variants in Section 5.4.

End-to-end data flow. Given a preprocessed CSI segment $\mathbf{X} \in \mathbb{R}^{B\times 1\times T\times F}$ with T=500 and F=112, AmbientFM first applies adaptive frequency aggregation with F'=10 learnable frequency queries for each of c=16 output channels, producing $\mathbb{R}^{B\times 16\times T\times 10}$. The channel and frequency dimensions are then flattened and projected to frame-level embeddings in $\mathbb{R}^{B\times T\times 256}$. A 6-layer Transformer encoder with relative position bias produces the shared representation $\mathbf{Z} \in \mathbb{R}^{B\times T\times 256}$. All pre-training objectives and downstream tasks operate on $\mathbf{Z}$. During pre-training, $\mathcal{L}_{\text{CL}}$ is applied to temporally pooled $\mathbf{Z}$ with a 2-layer MLP projector, $\mathcal{L}_{\text{REC}}$ reconstructs the input CSI with a 3-layer Transformer decoder, and $\mathcal{L}_{\text{ACF}}$ predicts 50 autocorrelation lags from pooled $\mathbf{Z}$. For downstream tasks, $\mathbf{Z}$ is fed to a 1-layer LSTM and a task-specific head, with the encoder either frozen for temporal probing or adapted with bottleneck adapters.

4.2 Self-Supervised Learning Framework

We pre-train AmbientFM with three complementary objectives: contrastive learning, masked reconstruction, and physics-informed ACF prediction. Together, they encourage the encoder to capture discriminative, robust, and temporally meaningful structure from unlabeled CSI.

Contrastive learning. Given two augmented views $\mathbf{x}_1, \mathbf{x}_2$ of the same input $\mathbf{x}$, we obtain temporally pooled representations $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d$ and apply the NT-Xent loss:

$$\mathcal{L}_{\text{CL}} = -\frac{1}{2B}\sum_{i=1}^{2B} \log \frac{\exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_{j(i)})/\tau)}{\sum_{k=1}^{2B} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_k)/\tau)}, \tag{3}$$

where $\mathbf{p} = g(\mathbf{z})$ is an MLP projection, $j(i)$ denotes the positive pair, and $\tau = 0.3$. This objective encourages representations that preserve activity semantics while remaining robust to nuisance variation in channel conditions.

We use nine domain-informed augmentations that mimic realistic wireless perturbations, including additive noise, amplitude modulation, temporal shift, frequency perturbation, temporal crop, frequency masking, and Gaussian blur (Appendix D, Table 6). Each view is formed by independently sampling a subset of these transformations.

Masked reconstruction. We randomly mask 10% of temporal blocks and 10% of frequency bands to simulate packet loss and frequency-selective fading. A reconstruction head then predicts the original signal:

$$\mathcal{L}_{\text{REC}} = \|\hat{\mathbf{X}} - \mathbf{X}\|_2^2. \tag{4}$$

Unlike standard masked autoencoding, we reconstruct the full signal rather than only the masked regions. This encourages the model to capture temporal coherence and cross-frequency structure induced by multipath propagation.

Physics-informed ACF prediction. To encourage sensitivity to motion-related temporal structure, we add an auxiliary objective that predicts the normalized autocorrelation function (ACF) of CSI energy over time. Let $s_t = \mathbb{E}_f[\mathbf{X}_{tf}^2]$. The ACF is defined as

$$\text{ACF}[k] = \frac{\sum_{t=0}^{T-k-1} s_t s_{t+k}}{\sum_{t=0}^{T-1} s_t^2}, \tag{5}$$

which is computed efficiently via FFT. An MLP head predicts the first $K = 50$ lags:

$$\mathcal{L}_{\text{ACF}} = \|\widehat{\text{ACF}} - \text{ACF}(\mathbf{X})\|_2^2. \tag{6}$$

This objective encourages the encoder to capture rhythmic and periodic temporal patterns associated with human activity.

The full pre-training objective is

$$\mathcal{L} = \lambda_{\text{CL}}\mathcal{L}_{\text{CL}} + \lambda_{\text{REC}}\mathcal{L}_{\text{REC}} + \lambda_{\text{ACF}}\mathcal{L}_{\text{ACF}}, \tag{7}$$

with $\lambda_{\text{CL}} = 1$, $\lambda_{\text{REC}} = 4$, and $\lambda_{\text{ACF}} = 1$. We assign a larger weight to $\mathcal{L}_{\text{REC}}$ because its raw magnitude is smaller than the other two losses, and this weighting keeps the optimization balanced.

4.3 Pre-training Details

We optimize the model with AdamW using a learning rate of 5×10^{-5} and weight decay of 10^{-4}. Training uses a 10-epoch linear warmup followed by cosine annealing. We train in mixed precision (FP16) on 8 A100 GPUs with a total batch size of 80. To stabilize optimization, we apply gradient clipping with max norm 1.0 together with standard numerical stabilization. Pre-training runs for 100 epochs and takes approximately 165 hours (about 7 days).

4.4 Downstream Adaptation

A key advantage of foundation models is efficient adaptation to new tasks with minimal labeled data. We evaluate two adaptation strategies that span the efficiency–flexibility tradeoff.

Temporal Probing. The simplest adaptation strategy freezes all encoder parameters and trains only a lightweight classifier on the sequence of frame representations. Since wireless sensing depends on temporal dynamics, we use a lightweight temporal head rather than a purely linear probe:

$$\hat{y} = g_\phi\left(\text{LSTM}(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T)\right), \tag{8}$$

where $\mathbf{z}_t \in \mathbb{R}^d$ are the frozen encoder outputs and g_ϕ is a classification head. This is a common choice in sequential sensing models, where a purely linear probe may underuse temporal structure in frozen representations [37].

Bottleneck Adaptation. For more expressive adaptation, we insert lightweight adapters into each transformer layer while freezing all pre-trained weights. Each adapter uses a down-projection, GELU, and up-projection, and is added to the frozen layer output through a residual connection. The up-projection is zero-initialized so that adapters are identity at initialization. Adapters are attached in parallel to the three linear layers in each encoder block: the attention output projection and the two feed-forward projections. With $r = 192$ and $L = 6$, this adds 2.37M trainable parameters while keeping all 5.57M encoder parameters frozen. Both adaptation methods share the same temporal classifier, isolating the effect of adapter tuning.

Unless otherwise noted, all downstream models are optimized with AdamW using a learning rate of 10^{-3} and weight decay of 10^{-4}. We use cosine annealing with a 10-epoch linear warmup for both adaptation settings.

5 Downstream Evaluation

We now aim to understand whether a single AmbientFM backbone can provide transferable representations for diverse sensing tasks.

5.1 Tasks and Evaluation Protocol

Tasks and Datasets. We evaluate AmbientFM on 9 tasks spanning 6 categories of ambient perception. *Physiological sensing* includes fall detection, which identifies acute events from transient signal disruptions. *Activity understanding* includes human activity recognition (HAR) and motion source recognition (MSR), which distinguishes human from non-human motion [48, 50]. *Spatial reasoning* covers occupancy detection, localization, and proximity estimation. *Biometrics* includes user identification (UID), which distinguishes individuals based on their characteristic motion patterns. *Fine-grained motion* is represented by gesture recognition, a 276-class task involving short-duration micro-motions. *Spatial reconstruction* includes WiFi imaging, which reconstructs spatial structure from reflected signals. Together, these tasks vary in temporal scale, spatial resolution, and output format, providing a broad evaluation of representation quality.

We evaluate on three public benchmarks: CSI-Bench [46], which contains 461 hours of in-the-wild data across 26 environments, 35 users, and 16 device types; SignFi [20] for 276-class gesture recognition; and WiFiCam [25] for spatial reconstruction. Pre-training and downstream evaluation data are fully disjoint in device, environment, time, and participant pool, and no downstream data is seen during pre-training. We use the original train/test split for each benchmark. We release the split files and downstream evaluation code where source licenses permit.

Experiment Set-Up. We evaluate AmbientFM with two adaptation protocols, *Temporal Probe (TP)* and *Bottleneck Adaptation (BA)*, against a supervised baseline. TP trains only a temporal classifier on top of a frozen pre-trained encoder. BA keeps the encoder frozen and trains lightweight adapters with a task head. The *supervised baseline* uses the same encoder architecture trained from scratch with task labels. All methods use identical data splits and downstream training settings.

Our evaluation targets *cross-task* transfer from a single backbone. We report AUROC with 95% bootstrap confidence intervals for classification, and SSIM and PSNR for reconstruction. AUROC is the primary metric because it is threshold-independent and robust to class imbalance. Accuracy, F1, and false alarm rate are provided in Appendix E.1, Table 7.

5.2 Cross-Task Transfer Performance

Table 2 summarizes AmbientFM across 9 downstream tasks using a *single* pre-trained backbone. With bottleneck adaptation, AmbientFM achieves AUROC above 0.90 on all 8 classification tasks, spanning physiological sensing, activity recognition, spatial reasoning, and biometrics. On WiFi imaging, it reaches 0.726 SSIM and 18.87 PSNR, indicating that the learned representation also preserves spatial structure for dense reconstruction.

The supervised baseline differs from AmbientFM only in the absence of pre-training. AmbientFM + BA improves AUROC by at least 0.36 on six of the eight classification tasks; on Fall Detection and UID, where the supervised baseline is already strong, the benefit of pre-training appears with limited labels (Section 5.3). The two adaptation settings further clarify how reusable the learned representation is. Temporal probing already performs strongly on UID, Motion Source Recognition, Localization, Occupancy, and HAR, suggesting that many task-relevant features are directly accessible from the frozen pre-trained representation. Bottleneck adaptation is most helpful for tasks that require finer feature recombination, most notably Gesture Recognition, where performance increases from 0.634 to 0.999. This result is consistent with the difficulty of the task: Gesture Recognition involves 276 classes and short-duration micro-motions, which are harder to separate using only a frozen encoder and a lightweight temporal head.

Overall, these results show that AmbientFM provides a single transferable backbone for diverse wireless sensing tasks, supporting both classification and reconstruction.

Relationship to task-specific methods. Task-specific models can achieve higher performance when each task is optimized independently, although AmbientFM + BA exceeds the published references on HAR and Proximity (Appendix E.2). AmbientFM does not aim to exceed these per-task upper bounds. Instead, it offers a single pre-trained backbone that transfers across all nine tasks with lightweight adaptation, replacing nine separate training pipelines with one. This tradeoff is standard in foundation models such as wav2vec 2.0 [4] and LIMU-BERT [37], which favor transferability and label efficiency over peak task-specific accuracy. The practical advantage is reduced task-specific design and labeled data requirements, both of which remain major bottlenecks for scalable WiFi sensing.

5.3 Data Efficiency

To assess label efficiency in realistic deployments, we restrict supervision to $K \in \{5, 10, 25, 50, 500\}$ samples per class and report mean AUROC over three seeds. Figure 2 shows that AmbientFM consistently outperforms training from scratch in low-data settings. With bottleneck adaptation, Fall Detection reaches 0.754 AUROC at $K = 25$ and 0.816 at $K = 50$, while UID reaches 0.811 at $K = 25$ and 0.857 at $K = 50$. The gains are especially clear for spatial tasks: Localization improves from 0.655 at $K = 5$ to 0.989 at $K = 500$, and Proximity improves from 0.528 to 0.870 over the same range.

Table 2: Cross-task performance across downstream WiFi sensing tasks. AUROC and 95% confidence intervals (CI) are reported for classification tasks and SSIM / PSNR are reported for the reconstruction task Imaging†.

Method	Fall	HAR	Gesture	UID	Localization	MSR	Occupancy	Proximity	Imaging†
AmbientFM + BA	**0.919** (0.902, 0.935)	**0.923** (0.915, 0.932)	**0.999** (0.999, 0.999)	0.993 (0.990, 0.995)	**0.995** (0.992, 0.997)	0.974 (0.970, 0.977)	**0.989** (0.986, 0.992)	**0.936** (0.927, 0.946)	**0.726 / 18.87**
AmbientFM + TP	0.873 (0.851, 0.893)	0.915 (0.906, 0.922)	0.634 (0.623, 0.645)	**0.996** (0.994, 0.997)	0.990 (0.986, 0.994)	**0.992** (0.991, 0.993)	0.989 (0.985, 0.992)	0.928 (0.919, 0.938)	0.722 / 18.43
Supervised baseline	0.916 (0.899, 0.932)	0.527 (0.513, 0.541)	0.564 (0.562, 0.568)	0.995 (0.993, 0.997)	0.631 (0.606, 0.653)	0.593 (0.581, 0.606)	0.566 (0.544, 0.587)	0.494 (0.477, 0.510)	0.726 / 18.41

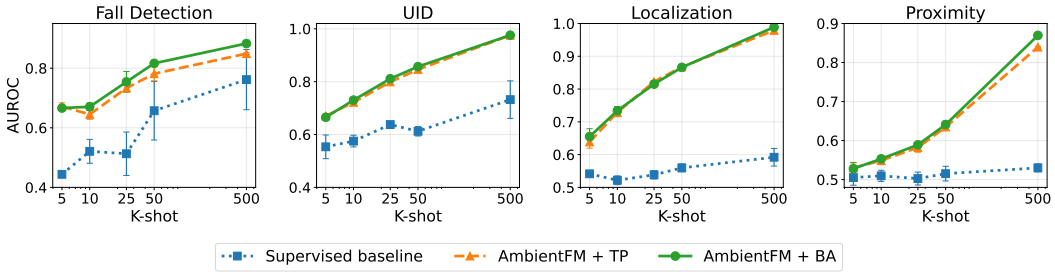


Figure 2: Cross-task transfer performance with small data sample sizes on four downstream tasks.

These results show that AmbientFM substantially reduces the amount of task-specific labeling needed for deployment. Supervised baseline improves much more slowly, reaching only 0.592 AUROC on Localization and 0.530 on Proximity even at $K = 500$, while AmbientFM reaches 0.989 and 0.870, respectively. This gap is consistent with pre-training providing transferable structure that is difficult to learn from limited labeled data alone. In practical terms, AmbientFM turns label-hungry wireless sensing tasks into tasks that can improve reliably with modest annotation effort.

5.4 Scaling Behavior

Model scaling. We pre-train Small (2M), Base (5M), and Large (12M) variants under the same objective and adapt all of them with the same bottleneck protocol (Appendix E.4, Table 9). Base delivers the best overall transfer and improves over Small on nearly every task, e.g., HAR from 0.840 to 0.923 and Proximity from 0.901 to 0.936. Large performs worse than Base on Fall, HAR, MSR, Occupancy, and Proximity while remaining strong on UID (0.990) and Localization (0.985), and this pattern persists across the tested learning-rate and warmup variations. The converged pre-training losses follow the same order: Large converges to 3.27, lower than Small (3.63) but higher than Base (3.09). Increasing capacity from Small to Base therefore improves both pre-training fit and downstream transfer, whereas further scaling under the same configuration improves neither. Base therefore provides the best trade-off under the current corpus and training configuration, and scaling larger variants with broader CSI data is a natural next step.

Pre-training data scaling. With the Base architecture, objectives, optimization, and adaptation protocol fixed, Fall Detection AUROC increases from 0.816 with 1/5 of the corpus to 0.884 with 3/5 and 0.919 with the full corpus, and HAR increases from 0.687 (1/5) to 0.923 (full). The 3/5 point recovers about two-thirds of the total Fall gain, consistent with diminishing returns at larger corpus sizes.

5.5 Representation Quality

To understand why AmbientFM transfers well, we measure inter-class distance using the mean Euclidean distance between class centroids. Compared with randomly initialized encoders, pre-training increases class separation by 14–52× across tasks, with an average gain of 25.9× (Appendix E.5, Table 10). This shows that AmbientFM learns representations where task-relevant classes are already well separated, making simple probing, bottleneck adaptation, and few-shot learning more effective.

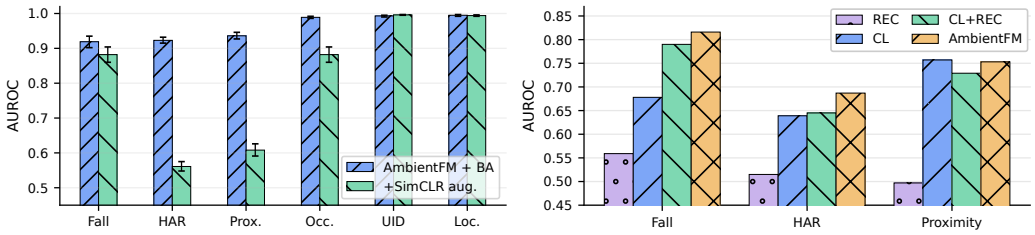


Figure 3: Effect of augmentation strategy on downstream performance (left). Loss component ablation on the 1/5 pre-training corpus (right).

5.6 Discussion

WiFi-tailored augmentations are critical for contrastive learning. Figure 3 (left) shows that standard SimCLR-style image augmentations are harmful when applied to CSI. Temporal tasks degrade sharply: HAR drops from 0.923 to 0.561, and Proximity from 0.936 to 0.608. In contrast, tasks dominated by stable spatial signatures are much less affected, with UID remaining at 0.993 and Localization at 0.995. This gap shows that CSI does not share the same invariances as natural images. Naive cropping or distortion can remove temporal windows and frequency components that carry sensing information. Effective contrastive pre-training therefore requires augmentations that reflect realistic wireless variation.

Each pre-training objective contributes to transfer quality. Figure 3 (right) ablates the pre-training objective using a 1/5 subset of the corpus, allowing us to measure the contribution of each loss component at manageable cost. The full AmbientFM objective performs best overall, especially on motion- and periodicity-related tasks: average AUROC over the three tasks is 0.524 for reconstruction-only (MAE-style) pre-training, 0.691 for contrastive-only, 0.721 for CL+REC, and 0.752 for the full objective. Removing ACF reduces Fall Detection from 0.816 to 0.790 and HAR from 0.687 to 0.645; removing reconstruction further lowers Fall Detection to 0.678. Although the 1/5-subset model has lower absolute performance than the full-corpus model, the *relative ordering* across objective variants is consistent with complementary objectives: contrastive learning captures discriminative structure, reconstruction preserves signal detail, and ACF modeling improves temporal and periodic sensing.

Deployment feasibility. AmbientFM's Base encoder (5.57M parameters, 21.3 MB; 4.72 GFLOPs per 500-step window) runs at 26.3–29.0 ms per window on a commodity CPU (AMD Ryzen 7 PRO 7840U) without GPU acceleration for batch sizes 1, 8, and 32, excluding signal accumulation and the downstream LSTM and head. This is well below the 5 s span of one input window at 100 Hz. We report the bottleneck dimension sensitivity study in Appendix E.6 and the sensitivity to mask ratio and loss weights in Appendix E.7.

Robustness and out-of-distribution transfer. AUROC remains above 0.92 on both 2.4 and 5 GHz for HAR and Proximity without band-specific tuning, and Fall Detection AUROC decreases gradually from 0.917 at the 100 Hz training rate to 0.813 under 40% packet loss (Appendix E.8). Under CSI-Bench's cross-user, cross-device, and cross-environment protocols, 20-shot adaptation reaches 0.704–0.808 AUROC for HAR and 0.631–0.820 for Proximity (Appendix E.9).

6 Limitations and Future Work

AmbientFM is a reusable backbone for adaptation-based WiFi sensing: each task still needs a lightweight adaptation stage rather than a zero-shot interface, and each prediction sees a 500-step context, so longer dependencies require a segment-level aggregator. Under the current corpus and configuration, Base is the best-performing scale, and larger variants have not yet improved transfer. The pre-training corpus covers geographically concentrated residences rather than hospitals or public spaces, and its consent terms prevent public release, which limits end-to-end reproduction of pre-training.

AmbientFM is also bounded by the physical limits of commodity WiFi: severely degraded links remove the temporal and spectral structure needed for sensing, and details beyond WiFi's resolution, such as fine visual appearance, are out of reach. Deployment factors vary jointly across the downstream environments, and zero-padding may expose device- or bandwidth-specific cues that our evaluation does not isolate. Passive inference of presence and activity also carries privacy and surveillance risks (Section 8). Future work includes prompt-based or zero-shot task specification [47], controlled one-factor and session-level evaluation, calibrated phase modeling, architectural sensitivity studies, and broader corpora and reconstruction benchmarks.

7 Conclusion

We present AmbientFM, a foundation model for ambient intelligence via WiFi sensing. Pre-trained on the largest real-world WiFi CSI corpus reported to date, 439 days of unlabeled CSI data spanning 20 device types and 11 environments, AmbientFM transfers effectively across nine diverse tasks using a single unified backbone, exceeding 0.90 AUROC on all classification tasks. Few-shot experiments demonstrate practical data efficiency. Our results establish that the foundation model paradigm is applicable to ambient intelligence: domain-specific architectures and physics-aware objectives enable learning general-purpose representations from unlabeled data, replacing task-specific pipelines with a unified approach to privacy-preserving, always-on perception.

8 Broader Impact

AmbientFM points toward practical ambient sensing systems that can operate on WiFi infrastructure already present in many indoor environments. Unlike cameras or microphones, WiFi CSI does not record identifiable visual or audio content, which gives it a privacy advantage for always-on sensing. This makes WiFi-based ambient intelligence especially relevant for assistive applications such as elderly care, fall detection, respiration monitoring, and smart-home automation.

At the same time, WiFi sensing can still infer human presence, motion, and activity, so responsible deployment is essential. Systems based on AmbientFM should be used with informed consent, clear communication about what is being sensed, and compliance with applicable privacy regulations. Future deployments should also consider safeguards that limit misuse, such as purpose-specific data collection, access control, and privacy-preserving data handling. For the same reason, pre-trained checkpoints are available only upon request and approval.

Funding and Competing Interests Disclosure

This research was supported by ADT Research through institutional research resources and computing infrastructure. No additional external funding was received for this work. The authors declare no additional competing financial interests beyond their disclosed institutional affiliations.

References

[1] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. UbiBreathe: A ubiquitous non-invasive WiFi-based breathing estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, page 277–286, New York, NY, USA, 2015.

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

[3] Iftikhar Ahmad, Arif Ullah, and Wooyeol Choi. WiFi-based human sensing with deep learning: Recent advances, challenges, and opportunities. *IEEE Open Journal of the Communications Society*, 5:3595–3623, 2024.

[4] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2020.

[5] Borna Barahimi, Hina Tabassum, Mohammad Omer, and Omer Waqar. Context-aware predictive coding: A representation learning framework for WiFi sensing. *IEEE Open Journal of the Communications Society*, 5:6119–6134, 2024.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[7] Yi Chu, Kanapathippillai Cumanan, Sathish K. Sankarpandi, Stephen Smith, and Octavia A. Dobre. Deep learning-based fall detection using WiFi channel state information. *IEEE Access*, 11:83763–83780, 2023.

[8] Zhengxin Guo, Wenyang Yuan, Linqing Gui, Biyun Sheng, and Fu Xiao. BreatheBand: A fine-grained and robust respiration monitor system using WiFi signals. *ACM Transactions on Sensor Networks*, 19(4), May 2023.

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

[11] Yuqian Hu, Muhammed Zahid Ozturk, Beibei Wang, Chenshu Wu, Feng Zhang, and K. J. Ray Liu. Robust passive proximity detection using Wi-Fi. *IEEE Internet of Things Journal*, 10(7):6221–6234, 2023.

[12] Yuqian Hu, Guozhen Zhu, Wei-Hsiang Wang, Beibei Wang, and K. J. Ray Liu. What you need is a good CSI. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*, page 1608–1610, New York, NY, USA, 2024.

[13] Jun Jiang, Xiaolong Ruan, and Shugong Xu. CSI-MAE: A masked autoencoder-based channel foundation model. *arXiv preprint arXiv:2601.03789*, 2026.

[14] Doeon Kim, Jungyoon Lee, Seongsin Kim, and Seong-heum Kim. WiFi-JEPA: Self-supervised learning for WiFi-CSI 3D human pose estimation. *arXiv preprint arXiv:2607.11064*, 2026.

[15] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. SpotFi: Decimeter level localization using WiFi. *SIGCOMM Computer Communication Review*, 45(4):269–282, August 2015.

[16] Gang Liu, Yanling Hao, and Yixuan Zou. CIG-MAE: Cross-modal information-guided masked autoencoder for self-supervised WiFi sensing. *arXiv preprint arXiv:2512.04723*, 2025.

[17] Xuanyu Liu, Shijian Gao, Boxun Liu, Xiang Cheng, and Liuqing Yang. WiFo-CF: Wireless foundation model for csi feedback. *IEEE Transactions on Wireless Communications*, 25:15039–15053, 2026.

[18] Xuanhao Luo, Zhizhen Li, and Yuchen Liu. CSI-JEPA: Towards foundation representations for ubiquitous sensing with minimal supervision. *arXiv preprint arXiv:2605.14171*, 2026.

[19] Yongsen Ma, Gang Zhou, and Shuangquan Wang. WiFi sensing with channel state information: A survey. *ACM Computing Surveys*, 52(3), June 2019.

[20] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. SignFi: Sign language recognition using WiFi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2(1), March 2018.

[21] Fucheng Miao, Youxiang Huang, Zhiyi Lu, Tomoaki Ohtsuki, Guan Gui, and Hikmet Sari. Wi-Fi sensing techniques for human activity recognition: Brief survey, potential challenges, and research directions. *ACM Computing Surveys*, 57(5), January 2025.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[23] Ahmed Y. Radwan, Mustafa Yildirim, Navid Hasanzadeh, Hina Tabassum, and Shahrokh Valaee. A Tutorial-Cum-Survey on Self-Supervised Learning for Wi-Fi Sensing: Trends, Challenges, and Outlook. *IEEE Communications Surveys & Tutorials*, 28:2345–2376, 2026.

[24] Sai Deepika Regani, Beibei Wang, Yuqian Hu, Guozhen Zhu, and K. J. Ray Liu. FallAware: An explainable learning approach to robust fall detection with WiFi. *IEEE Journal of Selected Areas in Sensors*, 2:71–83, 2025.

[25] Julian Strohmayer, Rafael Sterzinger, Christian Stippel, and Martin Kampel. Through-wall imaging based on WiFi channel state information. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 4000–4006, 2024.

[26] Wenhang Su, Chengjun Wang, Jiabao Li, Zhipeng Yu, and Li Jin. A novel feature extraction framework based on periodic features for human activity recognition using wearable sensor data. *Digital Signal Processing*, 168:105584, 2026.

[27] Zhiyong Tao, Zhelun Wang, Ying Liu, Yuqing He, and Yikai Wang. Wireless perceptual space modeling method for cross-domain human activity recognition. *ACM Transactions on Embedded Computing Systems*, 24(3), April 2025.

[28] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[29] Fei Wang, Yizhe Lv, Mengdie Zhu, Han Ding, and Jinsong Han. XRF55: A radio frequency dataset for human indoor action analysis. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 8(1), 2024.

[30] Fei Wang, Tingting Zhang, Wei Xi, Han Ding, Ge Wang, Di Zhang, Yuanhao Cui, Fan Liu, Jinsong Han, Jie Xu, and Tony Xiao Han. A survey on Wi-Fi sensing generalizability: Taxonomy, techniques, datasets, and future research prospects. *IEEE Communications Surveys & Tutorials*, 28:5227–5266, 2026.

[31] W. Wang, B. Wang, Y. Hu, G. Zhu, and K. J. R. Liu. Device-free room-level localization with WiFi utilizing spatial-frequency–time diversity. *IEEE Internet of Things Journal*, 11(21):35689–35698, 2024.

[32] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of WiFi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*, page 65–76, New York, NY, USA, 2015.

[33] Yuxi Wang, Kaishun Wu, and Lionel M. Ni. WiFall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):581–594, 2017.

[34] Zhengyang Wang, Sheng Chen, Wei Yang, and Yang Xu. Environment-independent Wi-Fi human activity recognition with adversarial network. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3330–3334, 2021.

[35] Chenshu Wu, Beibei Wang, Oscar C. Au, and K.J. Ray Liu. Wi-Fi can do more: Toward ubiquitous wireless sensing. *IEEE Communications Standards Magazine*, 6(2):42–49, 2022.

[36] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity Wi-Fi. *IEEE Transactions on Mobile Computing*, 18(6):1342–1355, 2019.

[37] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. LIMU-BERT: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, page 220–233, New York, NY, USA, 2021.

[38] Ke Xu, Jiangtao Wang, Hongyuan Zhu, and Dingchang Zheng. Evaluating self-supervised learning for WiFi CSI-based human activity recognition. *ACM Transactions on Sensor Networks*, 21(2), March 2025.

[39] Feng Zhang, Chen Chen, Beibei Wang, and K. J. Ray Liu. WiSpeed: A statistical electromagnetic approach for device-free indoor speed estimation. *IEEE Internet of Things Journal*, 5(3):2163–2177, 2018.

[40] Feng Zhang, Chenshu Wu, Beibei Wang, Hung-Quoc Lai, Yi Han, and K. J. Ray Liu. WiDetect: Robust motion detection with a statistical electromagnetic model. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 3(3), September 2019.

[41] Feng Zhang, Chenshu Wu, Beibei Wang, Min Wu, Daniel Bugos, Hangfang Zhang, and K. J. Ray Liu. SMARS: Sleep monitoring via ambient radio signals. *IEEE Transactions on Mobile Computing*, 20(1):217–231, 2021.

[42] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. CrossSense: Towards cross-site and large-scale WiFi sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*, page 305–320, New York, NY, USA, 2018.

[43] Leqi Zhao, Rui Xiao, Jianwei Liu, and Jinsong Han. One is enough: Enabling one-shot device-free gesture recognition with COTS WiFi. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pages 1231–1240, 2024.

[44] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-effort cross-domain gesture recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, page 313–325, New York, NY, USA, 2019.

[45] Rui Zhou, Xiang Lu, Pengbiao Zhao, and Jiesong Chen. Device-free presence detection and localization with SVM and CSI fingerprinting. *IEEE Sensors Journal*, 17(23):7990–7999, 2017.

[46] Guozhen Zhu, Yuqian Hu, Weihang Gao, Wei-Hsiang Wang, Beibei Wang, and K. J. Ray Liu. CSI-Bench: A large-scale in-the-wild dataset for multi-task WiFi sensing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[47] Guozhen Zhu, Yuqian Hu, Sakila S. Jayaweera, Wei-Hsiang Wang, Beibei Wang, and K. J. Ray Liu. CSI-TextBench: A dataset and benchmark for language-grounded ambient sensing perception. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

[48] Guozhen Zhu, Yuqian Hu, Beibei Wang, Chenshu Wu, Xiaolu Zeng, and K. J. Ray Liu. Wi-MoID: Human and nonhuman motion discrimination using WiFi with edge computing. *IEEE Internet of Things Journal*, 11(8):13900–13912, 2024.

[49] Guozhen Zhu, Yuqian Hu, Chenshu Wu, Wei-Hsiang Wang, Beibei Wang, and K. J. Ray Liu. Scaling WiFi sensing for ubiquitous home monitoring: Lessons from real-world deployment on millions of devices. *arXiv preprint arXiv:2506.04322*, 2025.

[50] Guozhen Zhu, Beibei Wang, Weihang Gao, Yuqian Hu, Chenshu Wu, and K. J. Ray Liu. SrcSense: Robust WiFi-based motion source recognition via signal-informed deep learning. *IEEE Journal of Selected Areas in Sensors*, 2:40–53, 2025.

A Dataset Description

A.1 Pre-Training Dataset

The pre-training dataset was designed for large-scale self-supervised representation learning, without requiring any manual annotations. CSI was collected continuously, 24/7, across diverse residential environments, capturing naturally occurring WiFi dynamics arising from everyday human activities, periods of inactivity, and environmental variations. IoT devices are distributed throughout participants' homes and operate under normal usage conditions, allowing the dataset to reflect realistic, in-the-wild sensing scenarios rather than segmented or scripted activities.

The resulting dataset spans 26 users, 20 IoT device types (33 devices in total), and 11 distinct residential environments, capturing heterogeneous WiFi sensing conditions across both commodity IoT deployments and dedicated sensing devices. In total, the corpus comprises over 439 days of continuous CSI recordings, amounting to approximately 20 TB of raw CSI data, $9{,}206{,}316$ data samples, collected across the 2.4 GHz and 5 GHz bands with channel bandwidths of 20/40/80 MHz.

Table 3 compares the corpus with representative public WiFi sensing datasets. It additionally spans 33 physical devices, 1×1 to 2×2 MIMO, more than 20 dB received-power variation, and 6–9 concurrent links per residence.

Table 3: Comparison of the AmbientFM pre-training corpus with representative public WiFi sensing datasets.

Dataset	Collection	Reported size	Duration	Devices/chipsets	Env.	Subjects	Band/BW
SignFi [20]	Scripted	8,280	–	1/1	2	5	5 GHz/20 MHz
WiFiCam [25]	Scripted	18,148	10 min	1/1	1	1	2.4 GHz/20 MHz
Widar3.0 [44]	Scripted	17K	–	1/1	3	16	5.8 GHz/20 MHz
XRF55 [29]	Scripted	42.9K	59.6 h	1/1	4	39	5 GHz/20 MHz
CrossSense [42]	Scripted	606K	–	1/1	3	100	5 GHz/20 MHz
CSI-Bench [46]	In-the-wild	231.6K	461 h	16 configurations	26	35	2.4/5 GHz, 20–80 MHz
AmbientFM corpus	24/7 unscripted, unlabeled	9.2M	439 days	20 types / 8 families	11	26	2.4/5 GHz, 20/40/80 MHz

A.2 Downstream Datasets

We evaluate AmbientFM on nine downstream tasks drawn from three public benchmarks: CSI-Bench [46] (CC BY-NC-ND 4.0), SignFi [20] (custom non-commercial research license from the College of William & Mary), and WiFiCam [25] (CC BY 4.0). Pre-training and downstream evaluation data are fully disjoint in device, environment, time, and participant pool. Table 4 summarizes each task; full collection protocols, hardware specifications, and data statistics are documented in the respective benchmark papers.

Table 4: Summary of downstream evaluation tasks.

Task	Source	#Classes	#Samples	Category
Fall Detection	CSI-Bench [46]	2	6,700	Physiological
Human Activity Recog.	CSI-Bench [46]	5	41,503	Activity
Motion Source Recog.	CSI-Bench [46]	4	60,900	Activity
Occupancy Detection	CSI-Bench [46]	2	$\sim$6,000	Spatial
Localization	CSI-Bench [46]	6	7,062	Spatial
Proximity Recognition	CSI-Bench [46]	4	64,154	Spatial
User Identification	CSI-Bench [46]	6	20,300	Biometrics
Gesture Recognition	SignFi [20]	276	8,280	Fine-grained motion
WiFi Imaging	WiFiCam [25]	–	18,148	Reconstruction

SignFi [20] captures wireless signal variations induced by 276 American Sign Language gestures from five users across a laboratory and a home setting, yielding 8,280 samples of 200 time steps $\times$ 30 subcarriers $\times$ 3 antenna pairs. WiFiCam [25] provides temporally synchronized CSI and RGB image pairs collected via an ESP32-S3-based transmitter–receiver system at 2.4 GHz (100 Hz CSI, 30 Hz video at 640×480), containing ten minutes of continuous human walking data (18,148 aligned CSI–image pairs).

B Data Collection Protocol

B.1 Subjects and Scenarios

Our pre-training dataset includes CSI data collected from 26 participants aged 5–65 years across multiple residential environments, each with distinct household compositions. The participants comprise 8 adult males, 13 adult females, and 5 children, with heights ranging from 130 to 180 cm and body weights from 30 to 90 kg. All participants signed a consent form prior to participation. The data is collected with IRB approval. Downstream data are from publicly available datasets collected under their respective IRB approvals.

B.2 Devices and Hardware Diversity

To ensure comprehensive coverage of real-world IoT infrastructure, we deploy 20 commercial WiFi-enabled edge devices operating on both 2.4 GHz and 5 GHz bands with 20, 40, and 80 MHz bandwidths. The devices span eight chipset families from eight vendors—MediaTek (MT7697N, MT8516, MT8183, MT6625L, MT6625LN, MT7658CSN), Qualcomm (IPQ4019 and others), Broadcom (BCM4345), Espressif (ESP8266/ESP8285, ESP32-S3), NXP (88W8997), Marvell (88W8887), RealTek (RTL8720DN), and Altobeam (ATBM6062)—covering configurations from low-cost smart plugs to high-performance routers and smart speakers. Antenna setups range from 1×1 to 2×2 MIMO, and the device set includes Amazon Echo variants, Google Nest/Home products, Apple HomePod, Wyze cameras and plugs, and several other IoT devices. Before data collection, each device was validated with an in-house CSI verification tool for signal consistency, sampling stability, and amplitude dynamics; only devices meeting quality thresholds were deployed [12, 49]. All devices connect through an Asus GT-AX11000 router (Broadcom 4912 quad-core CPU at 2 GHz, 1 GB RAM).

B.3 Environments

Data is collected across 11 distinct real-world residential environments spanning studio apartments (as small as 576 sqft), one- and two-bedroom apartments, townhouses, and multi-floor single-family houses (up to 3,800 sqft). The environments range from single-occupant studios to multi-generational households with up to six residents (including children). Floor counts range from one to three, and layouts include open-plan, partitioned, and multi-room configurations. These environments vary in wall materials, furniture density, and ambient interference, introducing rich multipath and occlusion effects. Each environment features a distinct device deployment configuration determined by user availability and spatial constraints, with the number of concurrent sensing links ranging from 6 to 9.

B.4 Continuous Data Recording

We design a scalable data acquisition framework for continuous, 24/7 collection of WiFi CSI in real residential environments. The framework operates transparently within existing home networks, capturing CSI during normal WiFi usage without modifying end devices or disrupting user behavior.

CSI measurements are obtained through collaboration with router chipset vendors, who provide firmware-level access and vendor-specific utilities for CSI extraction at the driver level. We implement custom acquisition software for both Linux- and FreeRTOS-based platforms that directly interfaces with WLAN kernel modules to stream CSI from all associated devices. Data is periodically uploaded to cloud storage via AWS S3 APIs with adaptive scheduling to balance network load. All CSI files are indexed using UTC-based timestamps embedded at creation time, ensuring consistent temporal referencing across deployments. This always-on pipeline captures a wide range of real-world WiFi dynamics; the resulting CSI data are naturally unlabeled and used directly for self-supervised pre-training.

To support optional post-hoc analysis, participants may voluntarily log coarse-grained activities (e.g., presence changes, sleep cycles, room usage) via a lightweight spreadsheet-based annotation interface. These annotations are not required for pre-training but enable investigation of whether observed signal variations correspond to user activities, environmental changes, or network instability.

C CSI Quality Verification and Filtering Pipeline

Raw CSI measurements from commodity WiFi devices exhibit substantial variability arising from differences in chipset design, firmware behavior, antenna configuration, and deployment conditions. Typical failure modes include unstable sampling intervals, timestamp drift, gain fluctuations, and limited motion sensitivity. When such signals are indiscriminately included, these artifacts can bias self-supervised objectives and encourage shortcut learning, ultimately degrading representation quality and cross-domain generalization.

To construct a reliable pre-training dataset, we perform structured screening at the level of each *sensing device* prior to large-scale data ingestion. Rather than filtering individual segments during training, we evaluate whether each sensing device is suitable for inclusion based on the following criteria.

C.1 Verification Criteria

Inspired by existing research in CSI qualification [12], we evaluate each sensing device along four complementary dimensions. *Timestamp consistency*: we analyze inter-packet intervals to estimate the effective sampling rate and its temporal stability, excluding devices with sustained deviation from the nominal rate (30 or 100 Hz), pronounced jitter, or long temporal gaps. *Empty-scene stability*: for segments recorded without human motion, we assess whether CSI amplitude remains temporally stationary, rejecting devices that exhibit abnormal fluctuations, drift, or bursty variations. *Motion leakage under empty conditions*: we further screen empty segments for unintended motion signatures using lightweight amplitude-derived statistics, removing devices that repeatedly exhibit motion-like patterns in supposedly static scenes. *Motion sensitivity in active scenes*: for segments with known human motion, we verify that the device produces reliable motion-induced amplitude deviations relative to the corresponding empty baseline, excluding devices with insufficient sensitivity.

Only sensing devices satisfying all four criteria are retained for pre-training data collection.

C.2 Implementation

The screening procedure is implemented as an offline device-selection stage prior to large-scale data collection, supported by a custom MATLAB-based diagnostic tool. For each candidate device, short calibration sequences are recorded under both empty and motion conditions, from which the tool extracts diagnostic statistics covering temporal stability, amplitude stationarity, and motion sensitivity. The tool provides both quantitative summaries and visual diagnostics—including amplitude time series, time–subcarrier heatmaps, and stability score traces—enabling efficient identification of unstable devices, interference-dominated channels, and devices with insufficient motion sensitivity. Device-specific statistics are compared against empirically determined thresholds derived from deployment-wide distributions. The screening process is applied once per sensing device, requires no manual annotation or task-specific supervision, and ensures that retained devices exhibit both stable empty behavior and sufficient sensitivity to human motion.

C.3 CSI Pre-processing

CSI captures the frequency-domain characteristics of wireless channel propagation. We extract CSI amplitude from raw complex-valued measurements and apply a standardized pre-processing pipeline to ensure consistency between pre-training and downstream task evaluation.

C.3.1 Amplitude Extraction

The foundation model operates on CSI amplitude extracted from IEEE 802.11n/ac WiFi packets. Given the complex-valued channel frequency response $H(f_k) = |H(f_k)|e^{j\phi(f_k)}$, we retain only the amplitude $A_k = |H(f_k)|$. Amplitude is preferred over complex-valued input because phase measurements are highly sensitive to carrier frequency offset, sampling clock offset, and packet detection timing—all of which vary substantially across WiFi hardware—whereas amplitude provides more stable and hardware-invariant representations.

C.3.2 Data Format Standardization

CSI data from different sources arrive in varying tensor layouts, including single-sample tensors of shape $(F, T, 1)$, batch tensors of shape (N, T, F), and direct time–frequency matrices of shape (T, F). Our pipeline automatically detects the input format based on tensor dimensions and converts all data to the canonical format $\mathbf{X} \in \mathbb{R}^{1 \times T_{\text{target}} \times F_{\text{target}}}$, where $T_{\text{target}} = 500$ time steps and $F_{\text{target}} = 112$ frequency bins corresponding to the maximum number of OFDM subcarriers across supported WiFi configurations.

C.3.3 Pre-processing Pipeline

The pre-processing pipeline applies a strict sequence of operations to ensure consistency between pre-training and downstream evaluation. The order of operations is critical: padding must occur before normalization to match the statistics computed during pre-training.

Step 1: Shape Transformation. The input tensor is converted to (T, F) format using format-specific transformations. Two-dimensional inputs pass through unchanged; three-dimensional inputs with trailing singleton dimensions undergo squeeze and transpose operations; batch tensors are indexed to extract individual samples.

Step 2: Zero Padding. Dimensions are standardized to the target shape using zero padding:

$$\mathbf{X}_{\text{padded}}[t, f] = \begin{cases} \mathbf{X}[t, f] & \text{if } t < T_{\text{input}} \text{ and } f < F_{\text{input}} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The temporal dimension is padded or truncated to $T_{\text{target}} = 500$, and the frequency dimension is padded or truncated to $F_{\text{target}} = 112$.

Step 3: Normalization. Sample-wise min-max normalization is applied to the padded tensor:

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{X}_{\text{padded}} - \min(\mathbf{X}_{\text{padded}})}{\max(\mathbf{X}_{\text{padded}}) - \min(\mathbf{X}_{\text{padded}}) + \epsilon} \tag{10}$$

where $\epsilon = 10^{-8}$ prevents division by zero. Min-max normalization is preferred because the bounded output range prevents extreme activations, preserves relative amplitude variations that encode signal propagation and activity information, and accommodates hardware-specific gain control settings. The Pad $\rightarrow$ Normalize ordering is essential: reversing it would produce different normalization statistics because min-max normalization computed on the original data excludes padded zeros, and adding zeros after normalization would alter the effective value distribution.

Step 4: Channel Dimension. A leading channel dimension is added to produce the final output:

$$\mathbf{X}_{\text{final}} = \text{unsqueeze}(\mathbf{X}_{\text{norm}}, 0) \in \mathbb{R}^{1 \times 500 \times 112} \tag{11}$$

D Training Details

D.1 Foundation Model Backbone Pre-training

The input CSI spectrogram $\mathbf{X} \in \mathbb{R}^{1 \times T \times F}$ (where T is the time dimension and F is the frequency/subcarrier dimension) is first processed by an adaptive downsampling encoder that learns to compress the frequency dimension via cross-attention. The encoder operates with a single input channel, using 8 intermediate channels and 16 output channels. We employ $F_{\text{out}} = 10$ learnable query vectors, each of dimension $c_{\text{mid}} = 8$, to attend over the input frequency bins. The encoder computes attention scores between learnable queries and input frequency bins. The flattened features are projected and processed through a transformer encoder with learned relative position bias. Table 5 summarizes the architectural configurations across model scales.

D.1.1 Pre-training Objectives

The backbone is pre-trained with three joint objectives specifically designed for WiFi CSI data.

Table 5: Transformer Encoder Hyperparameters across model scales.

Hyperparameter	**Small (2M)**	**Base (5M)**	**Large (12M)**
Embedding dimension (d_{model})	192	256	768
Encoder layers	4	6	12
Attention heads	6	8	12
Feed-forward dimension	384	512	2048
Dropout	0.1	0.1	0.1
Max sequence length	512	512	512

We employ a SimCLR-style contrastive learning framework adapted for the unique characteristics of WiFi CSI data. Unlike natural images, CSI spectrograms exhibit strong temporal correlations, frequency-domain structure from multipath propagation, and sensitivity to environmental dynamics. Our augmentation pipeline is specifically designed to preserve semantic content while creating diverse views.

Table 6 details the signal-domain augmentations.

Table 6: Contrastive augmentations applied during pre-training. The first eight are each applied independently with $p = 0.5$; time reversal is applied with $p = 0.2$.

Augmentation	**Parameters**	p	**Rationale**
Additive Gaussian Noise	$\sigma = 0.005$	0.5	Receiver thermal noise
Amplitude Modulation	factor $= 0.05$	0.5	AGC and path-loss fading
Frequency Perturbation	scale $= 0.02$ (FFT)	0.5	Hardware variation
Temporal Crop	crop ratio $= 0.8$	0.5	Partial observation
Frequency Masking	mask ratio $= 0.1$	0.5	Frequency-selective fading
Gaussian Blur	$\sigma = 1.0$, kernel $= 5$	0.5	Spectral smoothing
Temporal Masking	mask ratio $= 0.1$	0.5	Packet loss
Temporal Shift	max steps $= \pm 3$	0.5	Asynchronous sampling
Time Reversal	—	0.2	Temporal invariance

We employ a masked channel modeling (MCM) objective that reconstructs masked portions of the CSI spectrogram. This objective is specifically designed to capture the spatial-temporal-frequency structure unique to WiFi channel measurements. The design is motivated by three considerations: (1) *Multipath Structure*—WiFi CSI encodes multipath propagation where different frequency subcarriers experience correlated but distinct phase rotations; (2) *Temporal Dynamics*—human activities create time-varying signatures in CSI; and (3) *Cross-Domain Invariance*—masking entire rows and columns helps learn representations robust to missing or corrupted subcarriers.

Physics-Informed Auxiliary Task: Autocorrelation Function Prediction. We incorporate a physics-informed auxiliary objective based on the Autocorrelation Function (ACF) of the CSI signal power time series. This design is inspired by the theoretical foundations of acoustic and RF sensing, particularly the relationship between signal statistics and physical phenomena.

The ACF auxiliary task is motivated by several factors: (1) *Doppler Signature Encoding*—the ACF captures Doppler shifts induced by human motion; (2) *Motion Pattern Discrimination*—following the theoretical framework in SrcSense [50], the ACF encodes temporal structure of activity-induced signal variations; (3) *Hardware-Invariant Features*—the ACF is robust to hardware-specific gain variations; and (4) *Temporal Scale Information*—the decay rate reveals characteristic time scales of activities.

D.1.2 Pre-training Infrastructure

Pre-training is conducted on 8 NVIDIA A100 40GB GPUs (AWS P4D.24xlarge instances) using PyTorch Distributed Data Parallel (DDP) with the NCCL backend. We employ mixed precision training (FP16/BF16) and gradient checkpointing to maximize memory efficiency. The total batch size is 80 samples (10 per GPU). We use the AdamW optimizer with a learning rate of 5×10^{-5} and

cosine annealing schedule over 100 epochs. Training data is staged from FSx to local NVMe storage via parallel rclone transfers to minimize I/O bottlenecks.

D.2 Downstream Task Adaptation

D.2.1 Temporal Probing

Temporal probing freezes all encoder parameters and trains only the temporal classifier, testing whether pre-trained representations are separable without encoder adaptation.

D.2.2 Bottleneck Adaptation

For each linear layer in the transformer encoder, we add a parallel bottleneck path:

$$\mathbf{y} = \mathbf{W}_{\text{base}}\mathbf{x} + \mathbf{W}_{\text{up}} \cdot \sigma(\mathbf{W}_{\text{down}}\mathbf{x}), \tag{12}$$

where $\mathbf{W}_{\text{base}}$ is frozen, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{r \times d_{\text{in}}}$ projects to the bottleneck with $r = 192$, and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d_{\text{out}} \times r}$ projects back with GELU activation. Dropout of 0.03 is applied within the adapter. The up-projection is zero-initialized so adapters act as identity at initialization.

D.2.3 WiFi-to-image Adaptation

For the WiFi-imaging downstream task, we adapt the architecture by replacing the original encoder introduced in WiFiCam [25] with our pre-trained AmbientFM model and introducing a lightweight output adapter that maps the AmbientFM backbone's embedding dimension to the target dimension required by the image generation pipeline. During training, we employ bottleneck adaptation to fine-tune the AmbientFM model.

D.2.4 Downstream Training Configuration

We use largely consistent training hyperparameters for both temporal probing and bottleneck adaptation to ensure a fair comparison. Both methods are optimized using AdamW with an initial learning rate of 0.001 and a weight decay of 1e-4. Training is performed for up to 100 epochs with early stopping based on validation performance (patience of 10 epochs). Gradient norms are clipped to 1.0, and a cosine annealing learning rate scheduler is applied, with the minimum learning rate set to 1% of the initial value. Cross-entropy loss is used for all downstream classification tasks. The main difference lies in batch size: temporal probing uses a batch size of 64, while bottleneck adaptation uses smaller batches (16–32) due to higher memory consumption during partial fine-tuning.

D.2.5 Few-Shot Learning Configuration

For few-shot evaluation, we sample $k \in \{5, 10, 25, 50, 500\}$ examples per class from the training set. Sampling is stratified by class label to ensure balanced representation across all categories. We report results averaged over three random seeds to ensure statistical reliability. Validation is restricted to the same k labeled samples per class, and hyperparameters are inherited from the full-data configuration rather than re-tuned for each k.

E Evaluation and Ablation Details

E.1 Cross-Task Transfer Results

Table 7 provides cross-task transfer performance on the test split for a single backbone. Overall, pretraining yields strong transfer across heterogeneous sensing objectives, with *Bottleneck Adaptation* achieving consistently high AUROC and low false alarms on most tasks.

High transfer on spatial reasoning and presence sensing. Localization transfers exceptionally well: Bottleneck Adaptation reaches 0.995 AUROC with 0.938 accuracy, 0.916 F1, and 0.014 FAR. Occupancy Detection is similarly strong (0.989 AUROC), with 0.958 accuracy / 0.958 F1 and 0.043 FAR. Proximity Recognition also transfers well, reaching 0.936 AUROC with improved thresholded performance (0.796 accuracy, 0.799 F1) and 0.069 FAR.

Table 7: Test-set performance across downstream tasks. AUROC is reported with 95% CI. FAR denotes false alarm rate.

Task	Method	AUROC	95% CI	Accuracy	F1	FAR
Fall Detection	Supervised baseline	0.916	(0.899, 0.932)	0.827	0.844	0.117
	AmbientFM + TP	0.873	(0.851, 0.893)	0.801	0.833	0.252
	AmbientFM + BA	0.919	(0.902, 0.935)	0.834	0.852	0.125
Human Activity Recognition	Supervised baseline	0.527	(0.513, 0.541)	0.520	0.137	0.200
	AmbientFM + TP	0.915	(0.906, 0.922)	0.765	0.685	0.072
	AmbientFM + BA	0.923	(0.915, 0.932)	0.800	0.729	0.059
User Identification	Supervised baseline	0.995	(0.993, 0.997)	0.943	0.937	0.014
	AmbientFM + TP	0.996	(0.994, 0.997)	0.933	0.932	0.018
	AmbientFM + BA	0.993	(0.990, 0.995)	0.936	0.936	0.017
Localization	Supervised baseline	0.631	(0.606, 0.653)	0.273	0.128	0.162
	AmbientFM + TP	0.990	(0.986, 0.994)	0.914	0.899	0.019
	AmbientFM + BA	0.995	(0.992, 0.997)	0.938	0.916	0.014
Motion Source Recognition	Supervised baseline	0.593	(0.581, 0.606)	0.522	0.273	0.217
	AmbientFM + TP	0.992	(0.991, 0.993)	0.936	0.920	0.027
	AmbientFM + BA	0.974	(0.970, 0.977)	0.886	0.842	0.049
Occupancy Detection	Supervised baseline	0.566	(0.544, 0.587)	0.506	0.021	0.002
	AmbientFM + TP	0.989	(0.985, 0.992)	0.955	0.955	0.048
	AmbientFM + BA	0.989	(0.986, 0.992)	0.958	0.958	0.043
Proximity Recognition	Supervised baseline	0.494	(0.477, 0.510)	0.284	0.111	0.250
	AmbientFM + TP	0.928	(0.919, 0.938)	0.749	0.752	0.085
	AmbientFM + BA	0.936	(0.927, 0.946)	0.796	0.799	0.069

Strong activity understanding with modest adaptation. For HAR, Bottleneck Adaptation improves both ranking and decision metrics, reaching 0.923 AUROC and 0.729 F1 with 0.059 FAR. Motion Source Recognition achieves the highest AUROC with Temporal Probe (0.992) and strong decision metrics (0.936 accuracy, 0.920 F1, 0.027 FAR), while Bottleneck Adaptation maintains high performance (0.974 AUROC) with a slight FAR increase (0.049).

Biometrics and event sensing. User identification is near-saturated across all adaptation strategies, with AUROC $\geq$ 0.993 and very low FAR ($\leq$ 0.018). For Fall Detection, Bottleneck Adaptation slightly improves AUROC over supervised baseline (0.919 vs. 0.916) with comparable thresholded performance (0.852 F1) and low FAR (0.125).

Supervised baseline vs. pre-trained. Supervised baseline underperforms on most tasks, with particularly large gaps on HAR (0.527 AUROC, 0.137 F1), Localization (0.631 AUROC, 0.128 F1), and Proximity (0.494 AUROC, 0.111 F1). In contrast, temporal probing already yields strong transfer on several tasks (e.g., Localization 0.990 AUROC; Motion Source Recognition 0.992 AUROC), while Bottleneck Adaptation provides the best or most consistent improvements in thresholded metrics (e.g., HAR and Proximity) and maintains low false alarm rates.

E.2 Comparison with Task-Specific References

Table 8 compares AmbientFM + BA with published task-specific references, each of which optimizes a separate model for one task. AmbientFM + BA exceeds the references on HAR (0.800 vs. 0.754) and Proximity (0.796 vs. 0.775) while using one pre-trained backbone for all nine tasks.

E.3 Representation Analysis

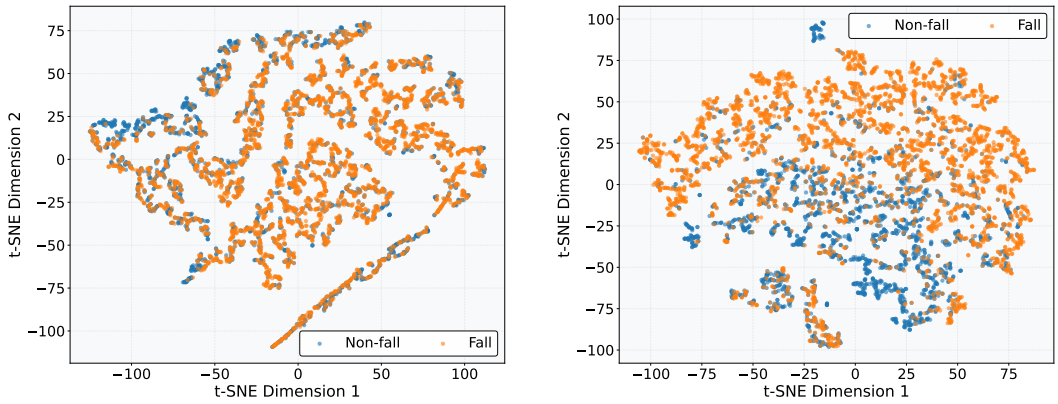
To evaluate the quality of learned representations, we employ t-distributed Stochastic Neighbor Embedding (t-SNE) [28], a widely-used dimensionality reduction technique that projects high-dimensional feature embeddings into a two dimensional space while preserving local neighborhood structures. t-SNE visualization provides intuitive insights into how well the model separates different

Table 8: Comparison with published task-specific references. Classification tasks report accuracy and Imaging reports SSIM / PSNR. References are from CSI-Bench [46] for the seven CSI-Bench tasks, SignFi [20] for Gesture, and WiFiCam [25] for Imaging.

Method	**Fall**	**HAR**	**Gesture**	**UID**	**Loc.**	**MSR**	**Occ.**	**Prox.**	**Imaging**
Task-specific reference	0.949	0.754	0.948	0.995	1.000	0.996	0.988	0.775	0.749 / 20.39
AmbientFM + BA	0.834	0.800	0.939	0.936	0.938	0.886	0.958	0.796	0.726 / 18.87

classes in the learned representation space. Well-separated clusters indicate that the model has learned discriminative features that capture class-specific patterns, while overlapping clusters suggest poor feature learning. We compare the quality of the representation between randomly initialized models and SSL pretrained models to demonstrate the effectiveness of pretraining.

Figure 4 illustrates the effectiveness of SSL pretraining for fall detection. The t-SNE embeddings in Figure 4a show that randomly initialized models produce highly overlapping clusters, where fall and non-fall samples are mixed together, demonstrating the lack of meaningful feature representation. In contrast, Figure 4b reveals that the SSL pretrained model achieves substantially better class separation, with fall and non-fall samples forming distinct clusters in the embedding space. This clear separation indicates that the pretrained model has learned discriminative features during self-supervised pretraining that effectively capture the essential patterns that distinguish fall events from non-fall activities. The improved separability in the embedding space directly translates to better classification performance, validating the value of SSL pretraining for learning robust representations even without labeled data.



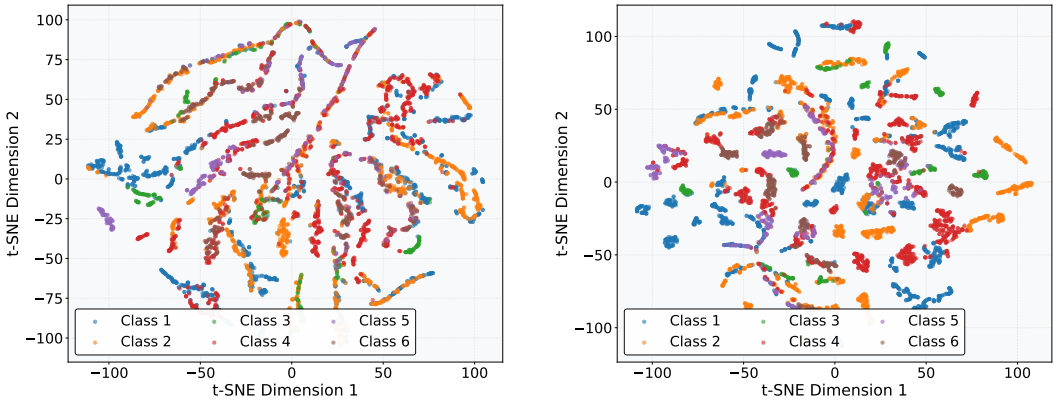
(a) Random initialization

(b) SSL pretrained model

Figure 4: t-SNE visualization of learned representations for fall detection. (a) Random initialization shows overlapping clusters with poor class separation, indicating the model has not learned meaningful features. (b) SSL pretrained model demonstrates separation between fall and non-fall classes, revealing that self-supervised pretraining learns discriminative representations beneficial for downstream tasks.

Figure 5 presents the t-SNE visualization comparing representation quality for the localization task. In Figure 5a, the randomly initialized model produces severely entangled embeddings where all data points of the six classes overlap extensively, forming intertwined patterns with no clear spatial structure. This chaotic distribution indicates that, without proper feature learning, the model cannot distinguish between different locations based on the patterns of wireless signals.

Conversely, Figure 5b demonstrates that the SSL pretrained model achieves notably better separation in the embedding space. While some inter-class overlap persists due to the inherent complexity of the localization task, the pretrained model forms more cohesive class-specific clusters with clearer boundaries. This improvement suggests that SSL pretraining enables the model to learn location-dependent signal characteristics such as propagation patterns, multipath effects, and signal strength variations that are crucial for distinguishing between different locations. The enhanced separability in



(a) Random initialization

(b) SSL pretrained model

Figure 5: t-SNE visualization of learned representations for localization. (a) Random initialization exhibits highly overlapping clusters across all six classes, with no discernible boundaries between different locations, indicating the absence of meaningful spatial features. (b) SSL pretrained model shows improved cluster formation with reduced inter-class overlap, demonstrating that self-supervised pretraining captures spatial patterns that facilitate location-specific classification.

Table 9: Downstream performance (AUROC) across model scales, with the converged pre-training loss.

Model	Fall	HAR	UID	Loc.	MSR	Occ.	Prox.	Pre-train loss
Small (2M)	0.898	0.840	0.984	0.979	0.971	0.980	0.901	3.63
Base (5M)	0.919	0.923	0.993	0.995	0.974	0.989	0.936	3.09
Large (12M)	0.627	0.524	0.990	0.985	0.515	0.651	0.496	3.27

the pretrained embeddings translates to improved localization accuracy, validating the effectiveness of SSL pretraining for learning robust spatial representations from unlabeled wireless sensing data.

E.4 Scaling Behavior

We evaluate model scaling by pre-training three variants under identical objectives and training pipelines: Small (2M parameters), Base (5M parameters), and Large (12M parameters). Architectural configurations for each variant are given in Table 5. We adapt each model using the same bottleneck protocol on CSI-Bench. Table 9 reports the results.

The Base model achieves the best overall transfer performance. Compared to Small, it improves on most tasks: HAR increases from 0.840 to 0.923, Proximity from 0.901 to 0.936, Fall Detection from 0.898 to 0.919, Localization from 0.979 to 0.995, and Occupancy from 0.980 to 0.989. The Small model nevertheless remains competitive at less than half the parameters, with all tasks at or above 0.840 AUROC, offering an attractive trade-off for resource-constrained edge deployment.

The Large model degrades substantially on several tasks, including Fall Detection (0.627), HAR (0.524), MSR (0.515), Occupancy (0.651), and Proximity (0.496), while maintaining strong performance on UID (0.990) and Localization (0.985). This pattern persists across the tested warmup-schedule and learning-rate variations, so it is not explained by one particular hyperparameter choice. All three scales reach stable plateaus in pre-training; Large converges to a loss of 3.27, lower than Small (3.63) but higher than Base (3.09) (Table 9). Increasing capacity from Small to Base thus improves both pre-training fit and downstream transfer, whereas further scaling under the same configuration improves neither. Base therefore provides the best trade-off under the current corpus and training configuration. Scaling AmbientFM to larger variants is a natural direction for future work as broader CSI collection becomes available.

Table 10: Inter-class distance in the representation space.

Metric	Fall	Loc.	UID	Prox.	HAR	MSR
Pre-trained	3.61	4.12	5.16	3.64	1.57	2.58
Random Init	0.25	0.24	0.21	0.14	0.03	0.12
Ratio ($\times$)	14.4	17.4	24.2	26.5	52.0	21.0
Average	Pre: 3.45		Rand: 0.16		Ratio: 25.9$\times$	

Table 11: Bottleneck dimension (r) trade-off on Proximity Recognition. Trainable Params includes adapter parameters across all 6 encoder layers plus the LSTM temporal classifier and task head.

r	**Trainable Params**	**Accuracy**	**AUROC**
64	1.98M	0.778	0.928
128	2.77M	0.787	0.935
192	3.55M	0.794	**0.936**
256	4.34M	0.764	0.932

E.5 Representation Quality

To understand why AmbientFM transfers well, we quantify representation geometry using inter-class distance, defined as the mean Euclidean distance between class centroids in the encoder's output space. We compute this metric using up to 2,000 samples per task and compare pre-trained encoders against randomly initialized ones under the same architecture. Table 10 reports the results.

Pre-training produces substantially larger class separation across all tasks, ranging from 14.4$\times$ (Fall Detection) to 52.0$\times$ (HAR), with an average gain of 25.9$\times$ despite using no downstream labels. This geometric structure has two practical consequences. First, it explains the strong temporal probing results in Section 5.2: when classes form well-separated clusters, a simple LSTM readout on frozen representations can succeed without any encoder adaptation. Second, it explains the few-shot behavior in Section 5.3: well-separated clusters require only a small number of labeled examples to place decision boundaries, making pre-training particularly valuable in low-data regimes.

The per-task separation ratios also correlate with the transfer gaps observed in Table 2. HAR exhibits the highest separation gain (52.0$\times$) and the largest AUROC improvement from pre-training (0.527 to 0.923). Proximity shows 26.5$\times$ higher separation and improves from 0.494 to 0.936. Conversely, UID, which already achieves 0.995 from scratch, has a high absolute separation (5.16) but a comparatively modest ratio (24.2$\times$), consistent with the signal being inherently discriminative even without pre-training. This correspondence between representation geometry and downstream performance is consistent with pre-training organizing the encoder's feature space in a task-relevant manner.

E.6 Bottleneck Dimension Sensitivity

To study the effect of adapter expressiveness, we vary the bottleneck dimension $r \in \{64, 128, 192, 256\}$ while keeping the encoder frozen, and evaluate on Proximity Recognition (Table 11). Performance increases from $r = 64$ to $r = 192$, where accuracy (0.794) and AUROC (0.936) are highest. At $r = 256$, both metrics decline, indicating that excess adapter capacity leads to overfitting with limited downstream labels. The $r = 128$ configuration offers a favorable trade-off for resource-constrained deployments, achieving 0.935 AUROC with 22% fewer trainable parameters than $r = 192$.

E.7 Sensitivity to Pre-training Hyperparameters

We vary one pre-training parameter at a time on the fixed 1/5 pre-training corpus and report HAR AUROC after bottleneck adaptation (Table 12). The default mask ratio of 0.10 improves over 0.05 and 0.25 by 0.040 and 0.012 AUROC, and the default loss weighting λ_{CL}:λ_{REC}:$\lambda_{\text{ACF}} = 1$:4:1 improves over 1:1:1 and 1:8:1 by 0.048 and 0.015. These results support the selected defaults.

Table 12: Sensitivity of HAR AUROC to mask ratio and loss weights (1/5 pre-training corpus, bottleneck adaptation).

Parameter	Lower	Default	Higher
Mask ratio	0.05: 0.647	0.10: **0.687**	0.25: 0.675
Loss weights (λ_{CL}:λ_{REC}:λ_{ACF})	1:1:1: 0.639	1:4:1: **0.687**	1:8:1: 0.672

The remaining architectural settings balance frequency compression, token capacity, and temporal horizon. F'=10 compresses up to 112 input frequency bins into 10 latent channels; c=16 output channels give a 160-dimensional intermediate token before projection to d=256; and K=50 ACF lags cover 0.5 s at 100 Hz.

E.8 Robustness to Carrier Band and Packet Rate

Table 13 reports HAR and Proximity by carrier band with the same backbone and adaptation protocol. AUROC remains above 0.92 on both bands for both tasks without band-specific tuning.

Table 13: Performance by carrier band. Accuracy, AUROC, and F1 are mean $\pm$ standard deviation.

Task	Band	Test samples	Accuracy	AUROC	F1	FAR
HAR	2.4 GHz	1,067	0.876 ± 0.025	0.958 ± 0.007	0.827 ± 0.040	0.039
HAR	5 GHz	2,333	0.789 ± 0.013	0.927 ± 0.004	0.716 ± 0.022	0.068
Proximity	2.4 GHz	527	0.820 ± 0.019	0.953 ± 0.006	0.821 ± 0.021	0.060
Proximity	5 GHz	1,201	0.776 ± 0.021	0.933 ± 0.011	0.778 ± 0.022	0.076

To test unseen effective packet rates, we evaluate the 100 Hz Fall Detection model after randomly removing packets (Table 14). AUROC decreases gradually and remains 0.813 under 40% packet loss.

Table 14: Fall Detection under reduced effective packet rates. Mean $\pm$ standard deviation across repeated evaluations; the 100 Hz mean is consistent with Table 2 (0.919).

Rate	Packet loss	Accuracy	AUROC	F1
100 Hz	0%	0.839 ± 0.019	0.917 ± 0.009	0.860 ± 0.017
95 Hz	5%	0.802 ± 0.023	0.878 ± 0.032	0.829 ± 0.013
90 Hz	10%	0.789 ± 0.028	0.862 ± 0.045	0.813 ± 0.015
80 Hz	20%	0.760 ± 0.041	0.843 ± 0.058	0.777 ± 0.026
60 Hz	40%	0.720 ± 0.069	0.813 ± 0.100	0.729 ± 0.056

E.9 Cross-User, Cross-Device, and Cross-Environment Evaluation

The original benchmark splits do not isolate users, devices, or environments individually. We therefore additionally evaluate HAR and Proximity with CSI-Bench's explicit cross-user, cross-device, and cross-environment protocols (Table 15). Zero-shot accuracy is above chance throughout, and at 20 shots AUROC reaches 0.704–0.808 for HAR and 0.631–0.820 for Proximity on held-out users, devices, and environments.

Device-level statistics. Table 16 reports device-macro metrics for HAR and Proximity, weighting each device equally. Device-macro AUROC reaches 0.920 for HAR and 0.952 for Proximity, consistent with transfer across individual devices.

Table 15: Accuracy / AUROC under CSI-Bench's cross-domain protocols. Proximity has no 10-shot setting.

Task	Split	0-shot	1-shot	5-shot	10-shot	20-shot
HAR	Cross-device	0.411 / 0.604	0.388 / 0.591	0.483 / 0.676	0.537 / 0.769	0.575 / 0.808
	Cross-environment	0.425 / 0.537	0.399 / 0.584	0.400 / 0.615	0.430 / 0.654	0.492 / 0.704
	Cross-user	0.498 / 0.605	0.451 / 0.626	0.477 / 0.670	0.508 / 0.718	0.550 / 0.753
Proximity	Cross-device	0.432 / 0.638	0.430 / 0.661	0.497 / 0.705	–	0.605 / 0.820
	Cross-environment	0.357 / 0.600	0.366 / 0.586	0.405 / 0.618	–	0.516 / 0.738
	Cross-user	0.303 / 0.558	0.320 / 0.560	0.323 / 0.571	–	0.390 / 0.631

Table 16: Device-macro performance. Values are mean ± standard deviation across three seeds.

Task	Device-macro Acc.	Device-macro AUROC	Device-macro F1
HAR	0.730 ± 0.073	0.920 ± 0.038	0.683 ± 0.088
Proximity	0.834 ± 0.076	0.952 ± 0.033	0.833 ± 0.088

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's main claims, and Sections 5.2–5.5 provide the transfer, data-efficiency, scaling, and representation analyses that support them.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses the need for task-specific adaptation, the 500-step context, the best-performing model scale, the representativeness and availability of the pre-training corpus, the physical sensing limits of commodity WiFi, the scope of the deployment evaluation, and privacy risks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results such as theorems or lemmas. Equations in Sections 4.1–4.2 define architectural components and training objectives rather than theoretical claims requiring proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the architecture, objectives, preprocessing, training setup, adaptation methods, and evaluation protocol in Sections 4.1–5.1 and Appendices C.3, D.1, and D.2. Downstream evaluation uses public benchmarks with their original splits; the split files and downstream evaluation code are released, and pre-trained checkpoints are available upon request.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The pre-training corpus is not publicly released because its consent terms do not permit redistribution of in-home recordings, which include data from minors. Downstream evaluation uses publicly available benchmarks with their original splits. We release the split files and downstream evaluation code, together with environment and execution instructions, where source licenses permit. Pre-trained model weights are available to researchers upon request subject to approval, so the main transfer results can be verified on the public benchmarks.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sections 4.3 and 4.4, Tables 5 and 6, and Appendix D.2 provide the key hyperparameters, optimizer, schedules, adaptation settings, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 5.1 and Table 2 report 95% bootstrap confidence intervals for classification tasks, and Section 5.3 reports few-shot results averaged over three random seeds. Appendices E.8 and E.9 report mean $\pm$ standard deviation across repeated evaluations, seeds, or groups, as stated in each table caption.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.3 and Appendix D.1.2 report the GPU type, training setup, batch size, and approximate total pre-training time; Section 5.6 reports CPU inference cost.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper states that pre-training data collection used informed consent and IRB approval (Appendix B) and discusses privacy-related risks and mitigation in Section 8.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 8 discusses both beneficial applications and potential harms, including privacy-sensitive inference and misuse, together with recommended safeguards.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The pre-training corpus is not released, and pre-trained checkpoints are available only upon request and approval to limit misuse of a model that can infer human presence and activity (Section 8). Data collection used informed consent and IRB approval (Appendix B).

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix A.2 cites the public downstream benchmarks and lists their licenses, and the paper states that the pre-training dataset was collected by the authors with participant consent.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released split files and downstream evaluation code are documented with environment and execution instructions. The pre-training corpus is not released because its consent terms do not permit redistribution, and the model is available upon request; their design and collection are documented in Section 3.1 and Appendices A–C.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The pre-training data collection involved 26 participants who signed informed consent forms prior to participation. Appendix B.1 describes participant demographics, and Appendix B.4 describes the collection protocol and annotation interface. Participants voluntarily recorded coarse activity annotations via a lightweight Google Spreadsheet interface. The WiFi CSI collection is passive and non-invasive, operating transparently within existing home networks. Downstream datasets are from publicly available benchmarks collected under their respective IRB approvals.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: All participants signed informed consent forms prior to data collection, and the pre-training data collection was conducted under Institutional Review Board (IRB) approval. Downstream evaluation datasets are publicly available benchmarks collected under their respective IRB approvals as discussed in Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the method, training pipeline, or evaluation. The work focuses on self-supervised learning for WiFi CSI.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.