# mmSLP: Semantic-Level Privacy Protection for mmWave Human Activity Recognition

**Anonymous authors**
Paper under double-blind review

## Abstract

Millimeter-wave (mmWave) radar enables privacy-friendly human activity recognition (HAR) without visual sensing. However, mmWave-based HAR may still reveal sensitive behaviors, raising privacy concerns for users. Therefore, we aim to reduce the sensitive-activity recognition capability of HAR systems. We observe that HAR systems commonly require user-provided data for fine-tuning, and these data remain under the user's control before sharing. We therefore apply carefully designed perturbations to these data before fine-tuning to reduce the recognition accuracy of sensitive activities. Based on this idea, we propose mmSLP, a semantic-level privacy protection framework for mmWave HAR. Experiments show that mmSLP reduces sensitive-activity accuracy from 85.41% to 42.12%, while maintaining normal-activity accuracy at 88.09% compared with 84.89% under clean fine-tuning, with low computational overhead.

## 1 Introduction

Millimeter-wave (mmWave) radar has emerged as a promising sensing modality for human activity recognition (HAR), owing to its contactless operation, robustness to illumination conditions, and reduced visual privacy exposure compared with camera-based sensing (Singh et al., 2019; Yu et al., 2022; Diraco et al., 2025). However, the absence of visual imagery does not eliminate privacy concerns, as radar signals can still encode rich and potentially sensitive behavioral information (Meng et al., 2020; Zhao et al., 2021; Sengupta et al., 2020). Recent studies have shown that mmWave sensing can recognize social interactions such as shaking hands and hugging (Wang et al., 2024a; Zhou et al., 2025). Such activities may reveal users' social interactions, raising potential privacy concerns.

Several studies have explored privacy protection in mmWave sensing. For example, AE-IPP (Sheng et al., 2026) focuses on privacy concerns associated with user identity and suppresses identity-related information during activity recognition. However, activity recognition may also reveal sensitive information. To protect behavioral privacy, mmFilter (Liu et al., 2023) perturbs radar signals to disable activity recognition, and MuFilter (Liu et al., 2025) extends this idea to multi-user scenarios. However, existing methods either do not consider activity-level privacy or disable activity recognition entirely.

In this paper, we target a finer-grained privacy protection setting for mmWave HAR. We observe that only a subset of activities may be considered sensitive by users, while the recognition of most normal activities should be preserved to maintain the utility of HAR for understanding users' daily activities. We refer to this selective protection objective as semantic-level privacy protection, where user-selected sensitive activities are protected by reducing the fine-tuned model's ability to recognize them, while normal activities remain recognizable.

To realize this protection objective, we examine where privacy protection can be introduced throughout the mmWave HAR pipeline, which typically includes radar data collection, data sharing, downstream fine-tuning of a pretrained model, and subsequent activity recognition, as illustrated in Figure 1. Among these stages, we identify the point before data sharing and downstream fine-tuning as the most suitable for introducing protection. At this stage, users still retain control over their locally collected data, allowing sensitive samples to be modified before they are released. Introducing protection here can influence what the adapted model learns without modifying the radar hardware or



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

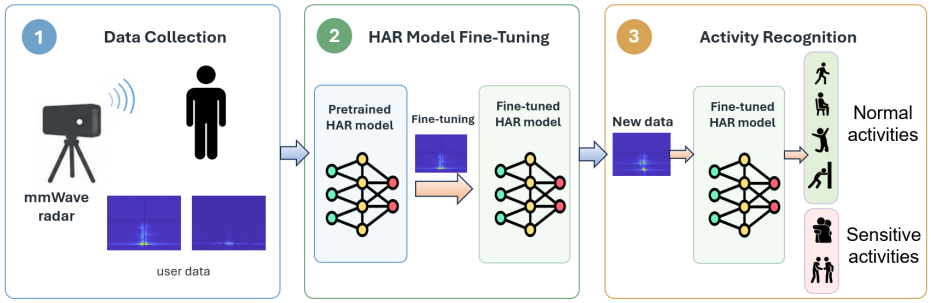



Figure 1: Overview of the mmWave HAR workflow. User-specific radar data are collected for fine-tuning a pretrained HAR model, which is subsequently used for activity recognition. We distinguish the activity space into normal activities and user-selected sensitive activities, motivating semantic-level privacy protection during downstream adaptation.

the downstream fine-tuning procedure, while also avoiding repeated processing of future measurements.

However, two challenges remain. The first challenge is determining how the data should be modified to achieve selective protection. Simply adding noise may either fail to protect sensitive activities or unnecessarily degrade the recognition of regular ones. Following the idea of Unlearnable Examples (UEs) (Huang et al., 2021), which use optimized perturbations to interfere with downstream learning, we design perturbations for sensitive fine-tuning samples so that the downstream model learns them as shortcuts for the corresponding activities. Since these shortcuts are absent from clean sensitive samples at inference time, sensitive activities become harder to recognize, while normal samples remain unchanged.

The second challenge arises from the limited computational and memory resources available on the user side. Accounting for downstream model adaptation can make perturbation generation computationally expensive. To reduce this overhead, mmSLP introduces two lightweight designs: directly optimizing one shared perturbation per sensitive activity and adopting a first-order approximation that fixes the adapted surrogate during perturbation updates, avoiding backpropagation through the temporary fine-tuning steps and higher-order gradients.

Based on these considerations, we propose mmSLP, a semantic-level privacy protection framework for mmWave HAR. mmSLP selectively protects user-defined sensitive activities through perturbation-induced shortcut learning, while preserving normal-activity recognition and reducing perturbation-generation overhead through a lightweight optimization procedure.

Extensive experiments demonstrate that mmSLP provides effective semantic-level privacy protection while preserving normal-activity utility with substantially lower computational overhead. We evaluate mmSLP on three mmWave HAR datasets, XRF55 (Wang et al., 2024a), Deep-Soli (Wang et al., 2016), and TinyRadar (Scherer et al., 2021). Averaged across the three datasets, normal-activity accuracy increases from 84.89% under clean fine-tuning to 88.09%, while sensitive-activity accuracy is substantially reduced from 85.41% to 42.12%. Meanwhile, compared with second-order optimization, the lightweight design reduces perturbation-generation time from 18.09 to 1.75 minutes per user and peak memory usage from 5.38 to 2.61 GiB.

The main contributions of this paper are as follows:

- We propose mmSLP, a semantic-level privacy protection framework for mmWave HAR, which enables users to protect selected privacy-sensitive activities while preserving the recognition utility of normal activities.
- We design a user-side perturbation strategy that makes downstream fine-tuning rely on the added perturbations when learning sensitive activities, while keeping normal-activity samples unchanged to preserve their recognition utility. The perturbations are generated with a lightweight optimization procedure to reduce computational overhead.

- Extensive evaluations demonstrate that mmSLP consistently provides effective privacy protection while preserving HAR utility across different datasets and model architectures.

# 2 RELATED WORK

## 2.1 PRIVACY PROTECTION IN MMWAVE AND SEMANTIC-LEVEL SENSING

Prior work on mmWave privacy protection has mainly considered identity privacy and sensing-function privacy. AE-IPP suppresses identity-related information while preserving activity recognition (Sheng et al., 2026). For behavioral privacy, mmFilter perturbs radar signals to disable activity recognition (Liu et al., 2023), and MuFilter extends this approach to multi-user settings (Liu et al., 2025). These methods either protect user-specific attributes or suppress activity recognition as a whole, leaving selective protection of sensitive activities within HAR largely unexplored.

Beyond mmWave sensing, semantic-level privacy has been explored in wearable and Wi-Fi sensing by suppressing sensitive activities while preserving non-sensitive sensing utility (Malekzadeh et al., 2020; Zhou et al., 2019; Zhang et al., 2020; Wang et al., 2024b). Poison2Cure further distinguishes private and regular activities and protects user-defined private activities while retaining regular sensing services (Hu et al., 2025). Despite these efforts, selective activity protection remains insufficiently explored in mmWave HAR.

## 2.2 UNLEARNABLE EXAMPLES FOR DATA PROTECTION

Unlearnable Examples (UEs) protect training data using optimized perturbations that induce shortcut learning and reduce generalization to clean samples (Huang et al., 2021). Subsequent studies improve robustness through stable or robust perturbations (Liu et al., 2024; Fu et al., 2022) and explore simpler class-wise or model-free protection strategies (Wu et al., 2022; Sadasivan et al., 2023). Other works improve transferability and broaden applicability through transferable, label-agnostic, or universal perturbations (Ren et al., 2022; Zhang et al., 2023; Chen et al., 2024). UE effectiveness, however, can still vary across downstream learning settings (Sandoval-Segura et al., 2023; Ye et al., 2025).

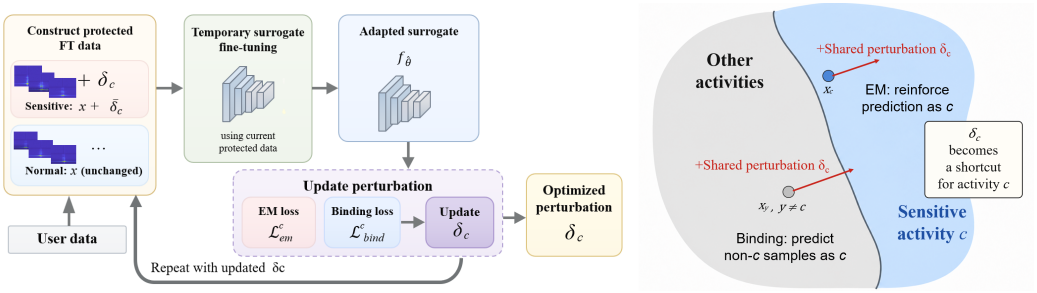
More recently, BAIT addresses the weakening of perturbation-induced shortcuts by pretrained semantic priors through perturbation–target binding and bi-level optimization (Li et al., 2026). Existing UEs mainly target general data unlearnability rather than selective suppression of sensitive activities, while BAIT also introduces substantial optimization overhead. Our work instead adapts shortcut-based protection to semantic-level privacy in mmWave HAR with lightweight perturbation optimization.

# 3 DESIGN OF MMSLP

## 3.1 OVERVIEW

Given that users retain control over the data used for fine-tuning, mmSLP introduces protection before these data are shared. Our goal is to reduce the fine-tuned HAR model's ability to recognize user-selected sensitive activities while preserving normal-activity recognition. As illustrated in Fig. 2a, mmSLP adds an activity-specific perturbation to each sensitive activity while leaving normal samples unchanged. The protected data are used to briefly adapt a surrogate model, which then guides perturbation optimization. After optimization, the protected data can be directly used for standard downstream fine-tuning.

The key idea is to make each perturbation an artificial shortcut for its corresponding sensitive activity. As illustrated in Fig. 2b, the error-minimizing and binding objectives encourage the model to associate the perturbation with the target activity, so that clean sensitive samples, which do not contain this shortcut at inference time, become harder to recognize. To make this protection practical under limited user-side resources, we introduce two lightweight designs in mmSLP: we directly optimize one shared perturbation for each sensitive activity instead of training an additional perturbation generator, and we adopt a first-order approximation that avoids backpropagating through the temporary fine-tuning process.




(a) Perturbation optimization process.

(b) Shortcut formation mechanism.

Figure 2: Overview of mmSLP perturbation generation and its underlying shortcut mechanism.

### 3.2 Protection Setting and Objective

We next define the desired protection behavior. The key point is that mmSLP optimizes perturbations applied to the user's fine-tuning data, such that the model fine-tuned on these protected data becomes less reliable at recognizing clean sensitive activities while preserving normal-activity recognition.

For a user $u$, let $\mathcal{D}^u = \{(x_i^u, y_i^u)\}_{i=1}^{N_u}$ denote the user-provided fine-tuning dataset. Here, $N_u$ is the number of fine-tuning samples from user $u$, $x_i^u$ denotes the radar representation of the $i$-th sample, and $y_i^u \in \mathcal{Y}$ is its activity label, where $\mathcal{Y}$ denotes the complete set of activity classes. The user specifies a subset $\mathcal{Y}_s \subset \mathcal{Y}$ as sensitive activities. The remaining activities are treated as normal and are denoted by $\mathcal{Y}_n = \mathcal{Y} \setminus \mathcal{Y}_s$. Accordingly, $\mathcal{D}_s^u$ and $\mathcal{D}_n^u$ denote the sensitive and normal subsets of $\mathcal{D}^u$, respectively. Given a pretrained HAR model $f_{\theta_0}$, mmSLP optimizes one shared perturbation $\delta_c$ for each $c \in \mathcal{Y}_s$, with $\Delta = \{\delta_c\}_{c \in \mathcal{Y}_s}$.

mmSLP modifies only the sensitive samples. For a sample $(x_i^u, y_i^u)$, the protected input is defined as

$$\tilde{x}_i^u = \begin{cases} x_i^u + \bar{\delta}_{y_i^u}(x_i^u), & y_i^u \in \mathcal{Y}_s, \\ x_i^u, & y_i^u \in \mathcal{Y}_n, \end{cases} \tag{1}$$

where $\tilde{x}_i^u$ denotes the sample released for downstream fine-tuning, and $\bar{\delta}_{y_i^u}(x_i^u)$ denotes the perturbation actually applied to the sample.

Radar samples may have substantially different signal magnitudes. To keep the perturbation strength comparable, we constrain it relative to the power of each radar input. For a sample $x$, we define $\bar{\delta}_c(x) = \Pi_\eta(\delta_c; x)$, where $\Pi_\eta$ denotes a projection operation that rescales $\delta_c$ only when the constraint is violated:

$$\frac{\|\bar{\delta}_c(x)\|_2^2}{\|x\|_2^2} \leq \eta. \tag{2}$$

Here, $\|\cdot\|_2$ denotes the Euclidean norm and $\eta$ is the relative perturbation-power budget. The optimized variable remains the shared perturbation $\delta_c$, while $\bar{\delta}_c(x)$ is its constrained form for sample $x$. Thus, samples from the same sensitive activity share the same underlying perturbation, while its applied magnitude is adjusted according to the signal power of each sample.

Applying Eq. 1 to all user samples yields the protected fine-tuning dataset $\tilde{\mathcal{D}}^u(\Delta)$, which is then used for standard downstream fine-tuning: $\theta_u^*(\Delta) = \mathrm{FT}(\theta_0, \tilde{\mathcal{D}}^u(\Delta))$. Our objective is to optimize $\Delta$ such that the resulting model $f_{\theta_u^*(\Delta)}$ has low recognition accuracy on clean sensitive activities while maintaining high accuracy on normal activities. The perturbations are applied only to the shared fine-tuning data, and protection is evaluated on clean samples after fine-tuning.

### 3.3 PERTURBATION OPTIMIZATION

To make the perturbation more effective, we optimize each activity-specific perturbation as an easy-to-learn shortcut for its corresponding sensitive activity. For a sensitive activity $c \in \mathcal{Y}_s$, its shared perturbation $\delta_c$ is evaluated using an adapted surrogate model $f_{\hat{\theta}}$, obtained by briefly fine-tuning on the current protected data as described in Sec. 3.4. For clarity, we omit the iteration index $t$ in this subsection.

Let $\mathcal{D}^u_{s,c} \subset \mathcal{D}^u_s$ denote the sensitive samples of activity $c$. Following the error-minimizing objective in Unlearnable Examples (Huang et al., 2021), we define the error-minimizing loss as

$$\mathcal{L}^c_{\mathrm{em}} = \mathbb{E}_{(x,c)\sim\mathcal{D}^u_{s,c}} \left[ \ell \left( f_{\hat{\theta}} \left( x + \bar{\delta}_c(x) \right), c \right) \right]. \tag{3}$$

The error-minimizing loss encourages perturbed samples of activity $c$ to remain easy to classify as $c$. However, this objective alone does not ensure shortcut reliance, since the original sample already contains the target activity pattern.

To strengthen perturbation–target association, following BAIT (Li et al., 2026), we add $\delta_c$ to samples with $y \neq c$ and encourage the adapted surrogate to predict them as activity $c$. The binding loss is defined as

$$\mathcal{L}^c_{\mathrm{bind}} = \mathbb{E}_{(x,y)\sim\mathcal{D}^u,\, y\neq c} \left[ \ell \left( f_{\hat{\theta}} \left( x + \bar{\delta}_c(x) \right), c \right) \right]. \tag{4}$$

Since samples with $y \neq c$ do not contain the target activity pattern, the binding loss encourages predictions toward $c$ to rely more strongly on $\delta_c$, thereby strengthening the association between the perturbation and its target activity.

During downstream fine-tuning, $\delta_c$ is added to sensitive samples of activity $c$, encouraging the model to use it as a shortcut. At inference time, clean sensitive samples do not contain $\delta_c$, making the learned shortcut unavailable and the corresponding activity harder to recognize.

Finally, we define the overall perturbation loss $\mathcal{L}_\Delta$ by combining the two losses over all sensitive activities:

$$\mathcal{L}_\Delta = \sum_{c\in\mathcal{Y}_s} \left( \lambda_{\mathrm{bind}} \mathcal{L}^c_{\mathrm{bind}} + \lambda_{\mathrm{em}} \mathcal{L}^c_{\mathrm{em}} \right), \tag{5}$$

where $\lambda_{\mathrm{bind}}$ and $\lambda_{\mathrm{em}}$ balance the two loss terms. With the adapted surrogate fixed, mmSLP optimizes $\Delta = \{\delta_c\}_{c\in\mathcal{Y}_s}$ by minimizing $\mathcal{L}_\Delta$.

### 3.4 LIGHTWEIGHT OPTIMIZATION ALGORITHM

To account for the model state after downstream adaptation while keeping perturbation optimization lightweight, mmSLP alternates between briefly adapting a surrogate model on the current protected data and updating the perturbations against the adapted surrogate. We further introduce two lightweight designs to reduce user-side overhead: direct activity-wise perturbation optimization and a first-order approximation.

**Lightweight Design 1: Direct activity-wise optimization.** To reduce the parameter and training overhead of perturbation generation, we directly optimize one shared perturbation for each sensitive activity instead of introducing an additional perturbation generator.

At each iteration, mmSLP first constructs the protected fine-tuning dataset using the current perturbations and briefly fine-tunes a temporary copy of the surrogate model. During this adaptation, sensitive samples contain their corresponding activity-specific perturbations, while normal samples remain unchanged. We define the fine-tuning loss $\mathcal{L}_{\mathrm{ft}}(\theta;\Delta)$ as

$$\begin{aligned} \mathcal{L}_{\mathrm{ft}}(\theta;\Delta) = \mathbb{E}_{(x,y)\sim\mathcal{D}^u_s} \left[ \ell \left( f_\theta \left( x + \bar{\delta}_y(x) \right), y \right) \right] \\ + \lambda_n \mathbb{E}_{(x,y)\sim\mathcal{D}^u_n} \left[ \ell \left( f_\theta(x), y \right) \right], \end{aligned} \tag{6}$$

where $\lambda_n$ controls the relative contribution of normal-activity samples. The first term corresponds to perturbed sensitive samples, while the second term corresponds to unchanged normal samples. We use $\lambda_n = 2$ to place additional emphasis on preserving normal-activity recognition. During model adaptation, $\Delta$ is fixed and the model parameters $\theta$ are updated by minimizing $\mathcal{L}_{\text{ft}}$.

Let $t \in \{1, \ldots, T\}$ denote the outer optimization iteration, where $T$ is the total number of iterations. Let $\theta^{(t)}$ denote the persistent surrogate parameters at iteration $t$, and let $\Delta^{(t)} = \{\delta_c^{(t)}\}_{c \in \mathcal{Y}_s}$ denote the current perturbation set. At iteration $t$, mmSLP creates a temporary copy of the persistent surrogate and adapts it for $K$ steps by minimizing $\mathcal{L}_{\text{ft}}$ with $\Delta^{(t)}$ fixed:

$$\hat{\theta}^{(t)} = \mathrm{FT}_K \left( \theta^{(t)}; \tilde{\mathcal{D}}^u \left( \mathrm{sg}[\Delta^{(t)}] \right) \right). \tag{7}$$

Here, $\hat{\theta}^{(t)}$ denotes the temporarily adapted model parameters, $K$ is the number of temporary adaptation steps, $\mathrm{FT}_K$ denotes $K$ model updates minimizing $\mathcal{L}_{\text{ft}}$, and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. This temporary adaptation allows the perturbations to be optimized against a model that has already responded to the current protected data.

**Lightweight Design 2: First-order approximation.** To reduce computation and memory overhead while retaining awareness of the adapted model state, we adopt a first-order approximation when updating the perturbations. Specifically, after temporary adaptation, $\hat{\theta}^{(t)}$ is treated as fixed, and gradients are computed only with respect to the perturbations. This avoids backpropagating through the temporary fine-tuning steps and the resulting higher-order gradients.

With $\hat{\theta}^{(t)}$ fixed, mmSLP minimizes the perturbation loss $\mathcal{L}_\Delta$ with respect to each $\delta_c^{(t)}$. We define the perturbation gradient $g_c^{(t)}$ and update $\delta_c^{(t)}$ as

$$g_c^{(t)} = \nabla_{\delta_c^{(t)}} \mathcal{L}_\Delta, \qquad \delta_c^{(t+1)} = \delta_c^{(t)} - \gamma \, \mathrm{sign}(g_c^{(t)}), \tag{8}$$

where $\gamma$ is the perturbation step size and $\mathrm{sign}(\cdot)$ is applied element-wise.

After updating all sensitive-activity perturbations, mmSLP reconstructs the protected dataset using the updated perturbations and updates the persistent surrogate by minimizing $\mathcal{L}_{\text{ft}}$. The updated surrogate and perturbations are then used in the next outer iteration. Together, direct activity-wise perturbation optimization and the first-order adaptation strategy allow mmSLP to account for downstream fine-tuning without introducing an additional perturbation generator or higher-order gradient computation. Algorithm 1 summarizes the complete procedure.

After optimization, the learned perturbations are applied only to the user-selected sensitive samples before data sharing, while normal samples remain unchanged. The protected dataset can then be used by the downstream learner with standard fine-tuning. By directly optimizing activity-wise perturbations and avoiding backpropagation through the temporary adaptation process, mmSLP accounts for downstream fine-tuning with substantially lower parameter, computation, and memory overhead.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

**Datasets, Models, and Protocol.** We evaluate on three radar HAR datasets: XRF55 (Wang et al., 2024a), Deep-Soli (Wang et al., 2016), and TinyRadar (Scherer et al., 2021). Across all datasets, we consider ResNet-18 (He et al., 2016), ConvLSTM (Shi et al., 2015), and ViT (Dosovitskiy et al., 2020) as downstream HAR models. Only sensitive training samples are perturbed, while normal samples remain clean, and evaluation is performed on held-out clean samples after fine-tuning. The main experiments assume access to the architecture and parameters of the pretrained model during perturbation generation (Hu et al., 2025). We report normal-activity accuracy $\mathrm{Acc}_n \uparrow$ and sensitive-activity accuracy $\mathrm{Acc}_s \downarrow$ to measure utility preservation and privacy protection, respectively.

**Algorithm 1** Lightweight Optimization of mmSLP

**Input:** User data $\mathcal{D}^u$, pretrained surrogate $f_{\theta_0}$, sensitive activities $\mathcal{Y}_s$, perturbation budget $\eta$, iterations $T$, temporary steps $K$, step size $\gamma$, weights $\lambda_{\text{bind}}, \lambda_{\text{em}}, \lambda_n$

**Output:** Protected dataset $\tilde{\mathcal{D}}^u$

Initialize $\theta^{(1)} \leftarrow \theta_0$ and $\delta_c^{(1)} \leftarrow 0$ for each $c \in \mathcal{Y}_s$
**for** $t = 1, \ldots, T$ **do**
  Set $\Delta^{(t)} = \{\delta_c^{(t)}\}_{c \in \mathcal{Y}_s}$
  Construct $\tilde{\mathcal{D}}^u(\Delta^{(t)})$ using Eq. 1
  Adapt temporary model $\hat{\theta}^{(t)}$ for $K$ steps by minimizing $\mathcal{L}_{\text{ft}}$ in Eq. 6, with $\Delta^{(t)}$ fixed
  Fix $\hat{\theta}^{(t)}$ and compute $\mathcal{L}_\Delta$ using Eq. 5
  **for all** $c \in \mathcal{Y}_s$ **do**
    Compute $g_c^{(t)} \leftarrow \nabla_{\delta_c^{(t)}} \mathcal{L}_\Delta$ using Eq. 8
    Update $\delta_c^{(t+1)} \leftarrow \delta_c^{(t)} - \gamma \operatorname{sign}(g_c^{(t)})$ using Eq. 8
  **end for**
  Reconstruct $\tilde{\mathcal{D}}^u(\Delta^{(t+1)})$ using Eq. 1
  Update persistent surrogate $\theta^{(t+1)}$ by minimizing $\mathcal{L}_{\text{ft}}$ in Eq. 6
**end for**
**return** $\tilde{\mathcal{D}}^u(\Delta^{(T+1)})$

**Implementation Details.** Unless otherwise specified, we use 10 adaptation samples per activity and optimize the activity-wise perturbations for 30 iterations with a relative power budget of $\eta = 0.2$. The perturbation step size is $\gamma = 0.1\sqrt{\eta}\, s_{\text{rms}}$, where $s_{\text{rms}}$ is the average RMS magnitude of the user's sensitive samples. The surrogate is optimized with Adam (Kingma & Ba, 2014) at a learning rate of $5 \times 10^{-4}$, with one temporary adaptation step ($K = 1$) on the protected support set per perturbation update. We set $\lambda_{\text{bind}} = 3.5$, $\lambda_{\text{em}} = 0.1$. Results are averaged over three random seeds. All experiments are conducted on a single NVIDIA GeForce RTX 3060 GPU with 12 GB of memory, using PyTorch 2.8.0 and CUDA 12.8.

**Baselines.** We compare against two reference settings, *No Fine-tuning* and *Clean Fine-tuning*, together with four protection baselines: *Random Noise*, *Random Transformation*, an *FSP-inspired* baseline following feature-space perturbation (Inkawhich et al., 2019), and a *PGD-style* baseline based on iterative projected-gradient optimization (Madry et al., 2017). The latter two are adapted to our semantic-level protection setting rather than exact reproductions of the original methods.

### 4.2 Main Results

We first evaluate whether mmSLP can suppress the recognition of user-selected sensitive activities while preserving the utility of normal activities after downstream fine-tuning. We conduct this evaluation across three radar datasets, multiple HAR architectures, and different radar representations. The results are summarized in Table 1.

On XRF55, mmSLP reduces the average sensitive-activity accuracy from $87.03\%$ under clean fine-tuning to $40.80\%$, while increasing the average normal-activity accuracy from $84.96\%$ to $87.92\%$ across three HAR architectures and three radar representations. Similar trends are observed on DeepSoli and TinyRadar: the average sensitive-activity accuracy decreases from $81.50\%$ to $36.94\%$ and from $87.70\%$ to $48.62\%$, respectively, while normal-activity accuracy is maintained or improved ($91.35\%$ vs. $92.34\%$ and $78.37\%$ vs. $84.00\%$). These results show that mmSLP consistently reduces the recognition of sensitive activities while retaining normal-activity utility across different datasets, radar representations, and HAR architectures.

### 4.3 Gray-box Transferability

To assess whether the protection generalizes beyond the surrogate model, we evaluate our method in a gray-box setting. Perturbations are generated using a ResNet-18 surrogate and transferred to independently trained victim models for downstream fine-tuning. Importantly, even when the victim

Table 1: Main results (%) across XRF55, Deep-Soli, and TinyRadar. Each entry reports $\mathrm{Acc}_n \uparrow / \mathrm{Acc}_s \downarrow$, where higher normal-activity accuracy indicates better utility preservation and lower sensitive-activity accuracy indicates stronger privacy protection. Bold denotes the best result for each metric among the protection methods.

| **XRF55** | **ResNet-18** | | | **ConvLSTM** | | | **ViT** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **RD+RA** | **RD** | **RA** | **RD+RA** | **RD** | **RA** | **RD+RA** | **RD** | **RA** |
| No Fine-tuning | 80.75 / 70.16 | 79.88 / 70.78 | 68.56 / 70.47 | 73.88 / 66.72 | 69.12 / 53.91 | 68.88 / 65.00 | 65.25 / 64.38 | 64.88 / 57.50 | 63.06 / 67.97 |
| Clean Fine-tuning | 89.56 / 91.09 | 91.44 / 93.28 | 89.94 / 94.38 | 82.69 / 84.22 | 78.69 / 75.00 | 84.75 / 85.47 | 82.31 / 87.66 | 80.94 / 83.12 | 84.31 / 89.06 |
| Random Noise | 90.06 / 86.25 | 92.38 / 78.12 | 91.56 / 83.91 | 82.19 / 30.00 | 77.19 / 35.94 | 85.25 / 83.75 | 80.94 / 62.19 | 79.62 / 80.00 | 84.19 / 88.28 |
| Random Transformation | 91.37 / 90.62 | 91.31 / 91.72 | 90.50 / 93.91 | **84.68** / 81.56 | 79.50 / 72.03 | 85.75 / 85.31 | 80.00 / 80.46 | 80.31 / 80.47 | 84.56 / 89.06 |
| FSP-inspired | 87.25 / 60.16 | 92.50 / 72.97 | 92.69 / 69.06 | 83.12 / **26.72** | 76.94 / 32.34 | 81.31 / 45.78 | 83.31 / 53.75 | 81.44 / 44.69 | 83.25 / 65.16 |
| PGD-style | 92.62 / 47.66 | 94.00 / 59.22 | 93.19 / 53.59 | 81.38 / 38.91 | 78.88 / 31.88 | 80.38 / **34.38** | 84.38 / 48.75 | 82.38 / **39.78** | 85.06 / 47.97 |
| **mmSLP** | **94.50 / 40.16** | **94.12 / 54.53** | **93.88 / 43.91** | 83.62 / 30.31 | **80.88 / 29.84** | **88.25** / 36.56 | **86.19 / 42.66** | **83.81** / 42.50 | **86.06 / 46.72** |

| | **Deep-Soli** | | | **TinyRadar** | | |
|---|---|---|---|---|---|---|
| **Method** | **ResNet-18** | **ConvLSTM** | **ViT** | **ResNet-18** | **ConvLSTM** | **ViT** |
| No Fine-tuning | 80.56 / 65.87 | 71.11 / 73.33 | 76.94 / 60.74 | 74.78 / 74.72 | 66.56 / 75.32 | 69.50 / 71.98 |
| Clean Fine-tuning | 92.78 / 79.07 | 89.24 / 88.15 | 92.04 / 77.28 | 83.77 / 91.07 | 77.90 / 84.77 | 73.43 / 87.26 |
| Random Noise | 93.19 / 73.33 | 89.58 / 87.04 | **92.71** / 85.74 | 84.84 / 88.93 | 78.36 / 84.31 | 74.50 / 86.78 |
| Random Transformation | 95.07 / 81.67 | 86.32 / 83.70 | 92.36 / 85.56 | 85.15 / 88.67 | 77.98 / 84.50 | 74.96 / 86.51 |
| FSP-inspired | **96.94** / 55.56 | 84.10 / 54.63 | 91.39 / 70.62 | 85.57 / 70.63 | 76.26 / 75.50 | 77.34 / 69.51 |
| PGD-style | **96.94** / 33.33 | 85.97 / **37.59** | 91.88 / 53.70 | 87.05 / 57.20 | 79.26 / 53.94 | 81.45 / 54.39 |
| **mmSLP** | 94.79 / **21.48** | **89.72** / 40.74 | 92.50 / **48.60** | **87.53 / 41.40** | **82.36 / 51.42** | **82.10 / 53.04** |

Table 2: Gray-box protection performance (%) with ResNet-18 as the surrogate model and RD+RA as input. Each entry reports $\mathrm{Acc}_n \uparrow / \mathrm{Acc}_s \downarrow$. $\Delta\mathrm{Acc}_n$ and $\Delta\mathrm{Acc}_s$ are calculated relative to clean fine-tuning.

| **Victim Model** | **Clean FT** | **Gray-box mmSLP** | $\boldsymbol{\Delta}\mathrm{Acc}_n$ | $\boldsymbol{\Delta}\mathrm{Acc}_s$ |
|---|---|---|---|---|
| ResNet-18 | 91.06 / 92.19 | 93.44 / 54.84 | +2.38 | -37.35 |
| ViT | 81.69 / 85.16 | 81.44 / 53.75 | -0.25 | -31.41 |
| ConvLSTM | 83.06 / 84.53 | 81.31 / 41.72 | -1.75 | -42.81 |
| Average | 85.27 / 87.29 | 85.40 / 50.10 | +0.13 | -37.19 |

architecture is also ResNet-18, the victim model is a separate model instance from the surrogate used for perturbation generation. We further evaluate cross-architecture transfer to ViT and ConvLSTM. All experiments use the RD+RA representation, and the generated perturbations remain fixed during transfer.

As shown in Table 2, mmSLP remains effective across victim architectures. On average, sensitive-activity accuracy decreases by 37.19 percentage points ($87.29\%$ to $50.10\%$), while normal-activity accuracy remains essentially unchanged ($85.27\%$ vs. $85.40\%$), demonstrating that the learned protection transfers across downstream architectures.

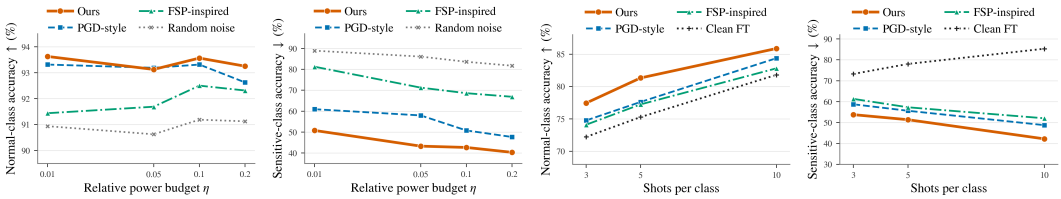
### 4.4 Ablation Studies

We evaluate whether the lightweight optimization in mmSLP can reduce user-side overhead while maintaining protection performance. We compare direct activity-wise perturbation optimization with a generator-based variant, and our first-order approximation with a second-order variant. All variants use the same experimental settings described in Sec. 4.1, with only the corresponding component changed. Results are shown in Table 3. For resource profiling, runtime measures only the perturbation-generation procedure, while peak GPU memory is measured as the maximum allocated memory during generation. Both metrics are averaged over the 16 evaluated users.

Adding a perturbation generator increases the number of optimized parameters, runtime, and peak GPU memory, while also weakening both sensitive-activity protection and normal-activity recognition. This indicates that, in our setting, directly optimizing activity-specific perturbations is not only more lightweight but also more effective. Similarly, the second-order variant provides no clear

Table 3: Component ablation on RD+RA using ResNet-18. Results are averaged over 16 users with 10-shot adaptation and $\eta = 0.2$. Higher $\mathrm{Acc}_n$ indicates better normal-activity utility, while lower $\mathrm{Acc}_s$ indicates stronger sensitive-activity protection.

| Method | $\mathrm{Acc}_n \uparrow$ (%) | $\mathrm{Acc}_s \downarrow$ (%) | Time/User $\downarrow$ (min) | Peak GPU Mem. $\downarrow$ (GiB) | Opt. Params $\downarrow$ (M) |
|---|---|---|---|---|---|
| mmSLP | 94.50 | 40.16 | 1.75 | 2.61 | 2.228 |
| mmSLP + Generator | 92.38 | 62.97 | 1.96 | 3.06 | 12.607 |
| mmSLP + Second-order | 93.25 | 41.88 | 18.09 | 5.38 | 2.228 |




(a) Budget: $\mathrm{Acc}_n$ (b) Budget: $\mathrm{Acc}_s$ (c) Shots: $\mathrm{Acc}_n$ (d) Shots: $\mathrm{Acc}_s$

Figure 3: Sensitivity analysis on RD+RA with ResNet-18. (a)–(b) Effect of the perturbation budget $\eta$ on normal- and sensitive-activity accuracy, respectively. (c)–(d) Effect of the number of adaptation shots. Higher $\mathrm{Acc}_n$ and lower $\mathrm{Acc}_s$ are preferred.

performance benefit over our first-order approximation, despite incurring much higher runtime and memory cost. This indicates that backpropagating through the temporary fine-tuning process provides no additional benefit in our setting, and that the first-order design offers a better efficiency–performance trade-off.

### 4.5 Sensitivity Analysis

Figure 3 examines the effects of the perturbation budget $\eta$ and the number of adaptation shots. As $\eta$ increases from $0.01$ to $0.2$, our sensitive-activity accuracy decreases from about $51\%$ to $40\%$, while normal-activity accuracy remains above $93\%$, indicating stronger privacy protection with little utility degradation. Increasing the number of shots improves normal-activity recognition for all methods, while our method consistently maintains the lowest sensitive-activity accuracy among the compared protection baselines.

## 5 Discussion

Our evaluation still has several limitations. Although mmSLP is tested on XRF55, Deep-Soli, and TinyRadar, these datasets cannot fully represent the variation in radar hardware, sensing environments, and activity distributions encountered in real deployments. The main setting also assumes access to a pretrained surrogate model, while our gray-box evaluation covers only a limited set of victim architectures. In addition, the protection is evaluated under fixed downstream fine-tuning protocols, and the set of user-selected sensitive activities is assumed to be specified before data sharing. Future work could study broader deployment conditions, more diverse downstream adaptation settings, and more flexible user privacy preferences.

## 6 Conclusion

We present mmSLP, a lightweight semantic-level privacy protection framework for mmWave HAR. mmSLP protects user-selected sensitive activities through direct activity-wise perturbation optimization and a first-order fine-tuning-aware approximation, while preserving the utility of normal activities. Experiments on XRF55, Deep-Soli, and TinyRadar demonstrate consistent protection across datasets, radar representations, and HAR architectures. Gray-box and ablation results further show cross-architecture transferability and substantially lower user-side computation and memory overhead.

## AI USE STATEMENT

We used generative AI tools to aid and polish the writing of this paper, including improving clarity, organization, grammar, and English expression. Generative AI tools were also used to assist with drafting and revising portions of the manuscript. We used generative AI tools for retrieval and discovery, including searching for related work, summarizing relevant literature, and identifying potentially relevant references. All literature-related information and citations were manually checked against the original sources before inclusion in the paper.

Generative AI tools were also used to assist with research ideation and execution, including discussions of experimental design, ablation settings, interpretation of experimental results, and presentation of the methodology. In addition, they were used to provide suggestions for improving the presentation and layout of figures and tables, and to assist with code implementation, modification, and debugging.

All AI-assisted content was reviewed and revised by the authors. AI-assisted code was manually inspected and tested, and the reported experimental results were obtained and verified by the authors. Technical descriptions, mathematical formulations, experimental analyses, and claims were checked against our implementation, experimental records, and the cited literature. The authors made the final decisions regarding the methodology, experiments, analysis, and presentation of the paper. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

## ETHICS STATEMENT

This work studies privacy protection for mmWave radar-based human activity recognition. Our experiments are conducted on existing publicly available datasets, and we do not collect additional human-subject data or personally identifiable information.

The proposed method is designed to reduce the learnability of user-selected sensitive activities from user-provided fine-tuning data while preserving the recognition of normal activities. Its intended use is to provide users with greater control over the behavioral information that can be learned from their sensing data. We acknowledge that techniques for manipulating training data may have potential dual-use implications if applied outside the intended privacy-preserving setting. In this work, we evaluate the method only for privacy protection in human activity recognition and do not consider its use for degrading unrelated sensing or safety-critical systems.

We follow the usage terms of the datasets and prior work used in this study and report experimental settings and limitations to support reproducibility and responsible interpretation of the results.

## REPRODUCIBILITY STATEMENT

We provide the complete formulation and optimization procedure of mmSLP in Sec. 3, including the perturbation construction, surrogate objective, first-order approximation, and the overall optimization algorithm. The experimental setup in Sec. 4.1 specifies the datasets, HAR architectures, evaluation protocol, implementation details, hyperparameters, baseline methods, and random-seed setting used in our experiments. Additional details on the activity partitions, dataset splits, and baseline implementations are provided in the appendix. Together, these materials are intended to provide the information necessary to reproduce the reported results.

## REFERENCES

Chaochao Chen, Jiaming Zhang, Yuyuan Li, and Zhongxuan Han. One for all: A universal generator for concept unlearnability via multi-modal alignment. In *Forty-first International Conference on Machine Learning*, 2024.

Giovanni Diraco, Gabriele Rescio, and Alessandro Leone. Radar-based activity recognition in strictly privacy-sensitive settings through deep feature learning. *Biomimetics*, 10(4):243, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An

image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data against adversarial learning. *arXiv preprint arXiv:2203.14533*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jingzhi Hu, Xin Li, Jin Gan, and Jun Luo. Poison to cure: Privacy-preserving wi-fi multi-user sensing via data poisoning. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*, pp. 47–62, 2025.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *arXiv preprint arXiv:2101.04898*, 2021.

Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7059–7067, 2019. URL https://api.semanticscholar.org/CorpusID:191525888.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Zhihao Li, Gezheng Xu, Jiale Cai, Ruiyi Fang, Di Wu, Qicheng Lao, Charles Ling, and Boyu Wang. When priors backfire: On the vulnerability of unlearnable examples to pretraining. *arXiv preprint arXiv:2603.04731*, 2026.

Hankai Liu, Xiulong Liu, Xin Xie, Xinyu Tong, Tuo Shi, and Keqiu Li. Application-oriented privacy filter for mmwave radar. *IEEE Communications Magazine*, 61(12):168–174, 2023.

Xiulong Liu, Hankai Liu, Jiaqi Zhang, Xin Xie, and Keqiu Li. Multi-user behavioral privacy filtering for mmwave radar sensing. *IEEE Transactions on Mobile Computing*, 24(9):8347–8361, 2025.

Yixin Liu, Kaidi Xu, Xun Chen, and Lichao Sun. Stable unlearnable example: Enhancing the robustness of unlearnable examples via stable error-minimizing noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3783–3791, 2024.

Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Privacy and utility preserving sensor-data transformations. *Pervasive and Mobile Computing*, 63:101132, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017. URL https://api.semanticscholar.org/CorpusID:3488815.

Zhen Meng, Song Fu, Jie Yan, Hongyuan Liang, Anfu Zhou, Shilin Zhu, Huadong Ma, Jianhua Liu, and Ning Yang. Gait recognition for co-existing multiple people using millimeter wave sensing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 849–856, 2020.

Jie Ren, Han Xu, Yuxuan Wan, Xingjun Ma, Lichao Sun, and Jiliang Tang. Transferable unlearnable examples. *arXiv preprint arXiv:2210.10114*, 2022.

Vinu Sankar Sadasivan, Mahdi Soltanolkotabi, and Soheil Feizi. Cuda: Convolution-based unlearnable datasets. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3862–3871. IEEE, 2023.

Pedro Sandoval-Segura, Vasu Singla, Jonas Geiping, Micah Goldblum, and Tom Goldstein. What can we learn from unlearnable datasets? *Advances in Neural Information Processing Systems*, 36:75372–75391, 2023.

Moritz Scherer, Michele Magno, Jonas Erb, Philipp Mayer, Manuel Eggimann, and Luca Benini. Tinyradarnn: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars. *IEEE Internet of Things Journal*, 8(13):10336–10346, 2021.

Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE sensors journal*, 20(17):10032–10044, 2020.

Biyun Sheng, Wangquan Qin, Hui Cai, Jun Li, Li Lu, Tie Qiu, and Fu Xiao. Ae-ipp: Adversarial example enabled identity privacy preserving with mmwave signals. *IEEE Transactions on Mobile Computing*, 2026.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, pp. 51–56, 2019.

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*, 8(1):1–34, 2024a.

Guangjing Wang, Hanqing Guo, Yuanda Wang, Bocheng Chen, Ce Zhou, and Qiben Yan. Protecting activity sensing data privacy using hierarchical information dissociation. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9. IEEE, 2024b.

Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th annual symposium on user interface software and technology*, pp. 851–860, 2016.

Shutong Wu, Sizhe Chen, Cihang Xie, and Xiaolin Huang. One-pixel shortcut: On the learning preference of deep neural networks. *arXiv preprint arXiv:2205.12141*, 2022.

Kai Ye, Liangcai Su, and Chenxiong Qian. How far are we from true unlearnability? In *International Conference on Learning Representations*, volume 2025, pp. 23504–23520, 2025.

Chengxi Yu, Zhezhuang Xu, Kun Yan, Ying-Ren Chien, Shih-Hau Fang, and Hsiao-Chun Wu. Noninvasive human activity recognition using millimeter-wave radar. *IEEE Systems Journal*, 16(2): 3036–3047, 2022.

Jiaming Zhang, Xingjun Ma, Qi Yi, Jitao Sang, Yu-Gang Jiang, Yaowei Wang, and Changsheng Xu. Unlearnable clusters: Towards label-agnostic unlearnable examples. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3984–3993. IEEE, 2023.

Wei Zhang, Siwang Zhou, Dan Peng, Liang Yang, Fangmin Li, and Hui Yin. Understanding and modeling of wifi signal-based indoor privacy protection. *IEEE Internet of Things Journal*, 8(3): 2000–2010, 2020.

Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. Human tracking and identification through a millimeter wave radar. *Ad Hoc Networks*, 116:102475, 2021.

Rui Zhou, Songlin Li, Hongwang Zhang, Chenxu Liu, and Jiajun Sun. mmmulti: Multi-person action recognition based on multi-task learning using millimeter waves. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(2):1–25, 2025.

Siwang Zhou, Wei Zhang, Dan Peng, Yonghe Liu, Xingwei Liao, and Hongbo Jiang. Adversarial wifi sensing for privacy preservation of human behaviors. *IEEE Communications Letters*, 24(2): 259–263, 2019.

Table 4: Activity partition and subject split used in our experiments. For Deep-Soli and TinyRadar, the sensitive subset is introduced only as a controlled experimental partition for evaluating semantic-level protection.

| Dataset | Normal activities | Sensitive activities | Subjects |
|---|---|---|---|
| XRF55 | Throwing; picking up an object; clapping; jumping; running; sitting down; standing up; falling down; pushing; pulling | Using a phone; drinking water; shaking hands; hugging | 14 train / 16 test |
| Deep-Soli | Pinch index; palm tilt; finger slider; pinch pinky; slow swipe; fast swipe; push; pull | Finger rub; circle; palm hold | 6 train / 4 test |
| TinyRadar | Pinch index; pinch pinky; finger slide; finger rub; slow swipe; fast swipe; push; pull | Palm tilt; circle; palm hold | 6 train / 4 test |

# A Appendix

## A.1 Additional Experimental Details

**Activity partition and subject split.** For XRF55, we select 14 representative activities from the original 55 activity categories: *throwing*, *picking up an object*, *clapping*, *jumping*, *running*, *sitting down*, *standing up*, *falling down*, *pushing*, *pulling*, *using a phone*, *drinking water*, *shaking hands*, and *hugging*. The first ten activities are treated as normal activities, while the remaining four, *using a phone*, *drinking water*, *shaking hands*, and *hugging*, are treated as sensitive activities. These activities are selected as a representative privacy partition because they may reveal personal habits, social interactions, or contextual information that a user may prefer not to expose; the designation does not imply that these activities are inherently sensitive.

We use data from 30 subjects, with 20 repetitions per selected activity for each subject, resulting in 8,400 samples. We randomly select 14 subjects to train the subject-independent pretrained HAR models, while the remaining 16 subjects are used as unseen test users. For each test user, a small support set is used for downstream adaptation and perturbation generation, while the remaining clean samples are used only for evaluation.

Deep-Soli and TinyRadar are further included to evaluate whether the proposed method generalizes from full-body activity recognition to fine-grained radar gesture recognition. For these two datasets, the normal/sensitive partition is introduced only as a controlled experimental split and does not imply that the corresponding gestures are inherently privacy-sensitive.

For Deep-Soli, gesture labels 0–7 are treated as normal activities: *pinch index*, *palm tilt*, *finger slider*, *pinch pinky*, *slow swipe*, *fast swipe*, *push*, and *pull*. Labels 8–10, *finger rub*, *circle*, and *palm hold*, are treated as sensitive activities.

For TinyRadar, the normal activities are *pinch index*, *pinch pinky*, *finger slide*, *finger rub*, *slow swipe*, *fast swipe*, *push*, and *pull*. The sensitive activities are *palm tilt*, *circle*, and *palm hold*.

For both Deep-Soli and TinyRadar, six subjects are used to train the subject-independent pretrained models and the remaining four subjects are used as unseen test users. We use 20 samples per activity for each subject. The same subject split and activity partition are used across all compared methods and HAR architectures. Table 4 summarizes the activity partition and subject split used for all three datasets.

**Radar input construction.** The three datasets use different radar representations, which are converted into fixed-size tensors before being fed to the HAR models. Table 5 summarizes the input representation and tensor size used for each dataset.

For XRF55, each stored sample has shape $[1, 17, 256, 128]$. We remove the leading singleton dimension to obtain $[17, 256, 128]$. The last dimension consists of a 64-bin range–Doppler (RD) representation followed by a 64-bin range–angle (RA) representation. We therefore construct the three input modes as

Table 5: Radar representations and model input dimensions used in our experiments. The batch dimension is omitted.

| Dataset | Input representation | Input size |
|---|---|---|
| XRF55 | Range–Doppler (RD) | $17 \times 256 \times 64$ |
| XRF55 | Range–Angle (RA) | $17 \times 256 \times 64$ |
| XRF55 | RD+RA | $17 \times 256 \times 128$ |
| Deep-Soli | Range–Doppler | $64 \times 32 \times 32$ |
| TinyRadar | Range–Doppler | $17 \times 256 \times 64$ |

$$\mathrm{RD} = x[..., :64], \qquad \mathrm{RA} = x[..., 64:], \qquad \mathrm{RD{+}RA} = x.$$

Accordingly, RD and RA have input size $[17, 256, 64]$, while the combined RD+RA representation has size $[17, 256, 128]$. RD+RA is formed by directly concatenating the RD and RA representations along the last spatial dimension, rather than by channel stacking or feature-level fusion.

For Deep-Soli, each HDF5 sample contains a sequence of range–Doppler images from multiple radar channels. In our experiments, we use CH0 as the input channel. The selected channel is stored with shape $[T, 1024]$, where each frame is reshaped into a $32 \times 32$ range–Doppler map. We then linearly resample only the temporal dimension to 64 frames, resulting in a model input of size $[64, 32, 32]$.

For TinyRadar, we construct range–Doppler maps from the raw complex I/Q recordings of the two radar sensors. Each gesture instance is represented by 17 uniformly distributed temporal windows, with 32 sweeps in each window. For each window, we first apply RMS normalization in the complex domain and then perform an FFT along the slow-time dimension, followed by FFT shifting and magnitude extraction. The resulting maps from the two sensors are fused using RMS energy fusion. We then apply log compression and robust min–max normalization, and resize each map to $256 \times 64$. The final model input therefore has size $[17, 256, 64]$.

### A.2 Baseline Implementation Details

For a fair comparison, all protection baselines modify only the sensitive samples in the user support set, while normal-activity samples remain unchanged. Evaluation samples are not used during perturbation generation. All methods follow the same perturbation budget and downstream fine-tuning protocol as mmSLP.

**Random Noise.** This baseline adds independently sampled Gaussian noise to each sensitive support sample. The noise magnitude is scaled according to the corresponding radar sample and constrained by the same relative perturbation budget used for mmSLP. Normal samples remain unchanged.

**Random Transformation.** This baseline applies lightweight random transformations to sensitive support samples, including small shifts along the input dimension and amplitude scaling. The resulting change is treated as the perturbation and constrained by the same perturbation budget. Normal samples are not transformed.

**FSP-inspired baseline.** This baseline adapts feature-space perturbation to our selective protection setting. We first extract features using the pretrained HAR model and compute class prototypes from the user support set. Each sensitive class is assigned a target class, and its samples are optimized so that their representations move toward the target prototype while moving away from the original class prototype. The resulting perturbations are constrained using the same budget as mmSLP.

**PGD-style baseline.** This baseline directly optimizes class-wise perturbations for the sensitive activities using gradient-based updates. During perturbation generation, a lightweight differentiable proxy adaptation is used to approximate the effect of downstream fine-tuning. The perturbations are

Table 6: Average normal- and sensitive-activity accuracy (%) under different fine-tuning settings. Each entry reports $\mathrm{Acc}_n \uparrow / \mathrm{Acc}_s \downarrow$. XRF55 results are averaged across three HAR architectures and three radar representations, while Deep-Soli and TinyRadar results are averaged across three HAR architectures. The final column reports the average across the three datasets.

| Method | XRF55 | Deep-Soli | TinyRadar | Average |
|---|---|---|---|---|
| Clean FT | 84.96 / 87.03 | 91.35 / 81.50 | 78.37 / 87.70 | 84.89 / 85.41 |
| Normal-only FT | 87.52 / 45.94 | 93.63 / 56.54 | 83.89 / 53.59 | 88.35 / 52.02 |
| Poison2Cure | 83.62 / 42.93 | 79.51 / 37.10 | 83.57 / 47.13 | 82.23 / 42.39 |

optimized to reduce the recognition of clean sensitive activities after adaptation, while clean normal-activity samples are retained to limit unnecessary degradation of normal-activity recognition. The same perturbation budget and downstream evaluation protocol are used as for mmSLP.

### A.3 Effect of the Fine-Tuning Data Setting

Our main experiments assume that the user-provided fine-tuning dataset contains samples from both normal and sensitive activities. We further examine an alternative setting in which only normal-activity samples are provided for downstream fine-tuning. Table 6 summarizes the results together with an adapted implementation of Poison2Cure (Hu et al., 2025).

Poison2Cure considers a different data-sharing setting from ours. In its original formulation, the user provides labeled fine-tuning data only for regular activities and withholds private-activity samples from the fine-tuning dataset (Hu et al., 2025). The authors show that model adaptation on regular activities can nevertheless improve the recognition of private activities, motivating the use of poisoned regular-activity samples to suppress this unintended improvement.

We observe a different behavior in our mmWave HAR experiments. As shown in Table 6, simply removing sensitive activities from the fine-tuning set already substantially reduces their recognition accuracy. Averaged across the three datasets, sensitive-activity accuracy decreases from 85.41% under clean fine-tuning to 52.02% under normal-only fine-tuning, while normal-activity accuracy increases from 84.89% to 88.35%. Similar reductions are observed on all three datasets. This indicates that, in our setting, withholding sensitive samples itself already provides a considerable reduction in sensitive-activity recognition.

For this reason, we adopt a more stringent fine-tuning setting in the main experiments, where both normal and sensitive activities are included in the user-provided dataset. In this setting, the downstream learner receives correctly labeled examples of the sensitive activities and therefore obtains direct supervision for learning them. Effective privacy protection must counteract this supervision while simultaneously preserving the utility of normal activities. This setting allows us to evaluate whether the protection method itself can suppress sensitive-activity learning, rather than benefiting primarily from the absence of sensitive samples in the fine-tuning data.

Because Poison2Cure is designed around the regular-only fine-tuning assumption, directly including it in the main comparison would mix two different data-sharing settings. We therefore report its adapted results here for completeness, while keeping the main evaluation focused on methods that operate under the same fine-tuning setting as mmSLP.

### A.4 Effect of Normal-Activity Weighting

As described in Eq. 6, mmSLP assigns a larger weight to normal-activity samples during surrogate adaptation to better preserve normal-activity recognition. In the main experiments, we set $\lambda_n = 2$, while $\lambda_n = 1$ corresponds to equal weighting between the sensitive- and normal-activity terms. This weighting only affects surrogate adaptation and does not change the perturbation objective.

To examine its effect, we remove the additional emphasis on normal activities by setting $\lambda_n = 1$, while keeping all other settings unchanged.

As shown in Table 7, setting $\lambda_n = 1$ consistently reduces normal-activity accuracy by 1.33, 1.25, and 1.01 percentage points for ResNet-18, ConvLSTM, and ViT, respectively. Sensitive-activity

Table 7: Effect of removing additional normal-activity weighting on RD+RA. $\Delta\mathrm{Acc} = \mathrm{Acc}_{\mathrm{w/o\ weighting}} - \mathrm{Acc}_{\mathrm{mmSLP}}$, where the full mmSLP results are reported in the main paper.

| **Model** | $\mathrm{Acc}_n$ (%) | $\Delta\mathrm{Acc}_n$ | $\mathrm{Acc}_s$ (%) | $\Delta\mathrm{Acc}_s$ |
|---|---|---|---|---|
| ResNet-18 | 93.17 | -1.33 | 41.56 | +1.40 |
| ConvLSTM | 82.37 | -1.25 | 33.75 | +3.44 |
| ViT | 85.18 | -1.01 | 42.96 | +0.30 |

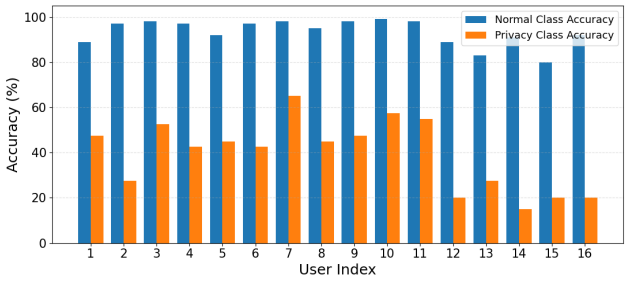



Figure 4: Per-user protection performance on XRF55 using ResNet-18 with the RD+RA representation. Results are reported for the 16 unseen test users. Higher normal-activity accuracy $\mathrm{Acc}_n$ indicates better utility preservation, while lower sensitive-activity accuracy $\mathrm{Acc}_s$ indicates stronger privacy protection.

accuracy also increases in all three cases, indicating slightly weaker protection. These results suggest that placing additional emphasis on normal-activity samples helps preserve normal-activity utility while also slightly improving the protection of sensitive activities. This weighting is therefore used as a training detail rather than an additional privacy-protection mechanism.

### A.5 Per-User Protection Performance

To further examine the consistency of mmSLP across individual users, we report the per-user results on XRF55 using ResNet-18 with the combined range–Doppler and range–angle (RD+RA) representation. The 16 users correspond to the unseen test users described in Appendix A.1. For each user, we report the normal-activity accuracy $\mathrm{Acc}_n$ and sensitive-activity accuracy $\mathrm{Acc}_s$ after downstream fine-tuning on the protected support set.

As shown in Fig. 4, mmSLP maintains high normal-activity accuracy for most users while substantially reducing sensitive-activity recognition. Although the protection strength varies across users, the overall trend is consistent, indicating that the average results reported in the main paper are not driven by only a small subset of users.