Neuro-Symbolic Fusion of Wi-Fi Sensing Data for Passive Radar with Inter-Modal Knowledge Transfer

Marco Cominelli*, Francesco Gringoli†, Lance M. Kaplan‡, Mani B. Srivastava§, Trevor Bihl‖, Erik P. Blasch‖, Nandini Iyer‖, and Federico Cerutti†

*DEIB, Politecnico di Milano, Italy. marco.cominelli@polimi.it
†DII, University of Brescia, Italy. {francesco.gringoli, federico.cerutti}@unibs.it
‡DEVCOM Army Research Lab, USA. lance.m.kaplan.civ@army.mil
§ECE Department, University of California, Los Angeles, USA. mbs@ucla.edu
‖Air Force Research Laboratory, USA. {trevor.bihl.2, erik.blasch.1, nandini.iyer.2}@us.af.mil

***Abstract*—Wi-Fi devices, akin to passive radars, can discern human activities within indoor settings due to the human body's interaction with electromagnetic signals. Current Wi-Fi sensing applications predominantly employ data-driven learning techniques to associate the fluctuations in the physical properties of the communication channel with the human activity causing them. However, these techniques often lack the desired flexibility and transparency. This paper introduces DeepProbHAR, a neuro-symbolic architecture for Wi-Fi sensing, providing initial evidence that Wi-Fi signals can differentiate between simple movements, such as leg or arm movements, which are integral to human activities like running or walking. The neuro-symbolic approach affords gathering such evidence without needing additional specialised data collection or labelling. The training of DeepProbHAR is facilitated by declarative domain knowledge obtained from a camera feed and by fusing signals from various antennas of the Wi-Fi receivers. DeepProbHAR achieves results comparable to the state-of-the-art in human activity recognition. Moreover, as a by-product of the learning process, DeepProbHAR generates specialised classifiers for simple movements that match the accuracy of models trained on finely labelled datasets, which would be particularly costly.**

***Index Terms*—neuro-symbolic AI, data fusion, Wi-Fi sensing**

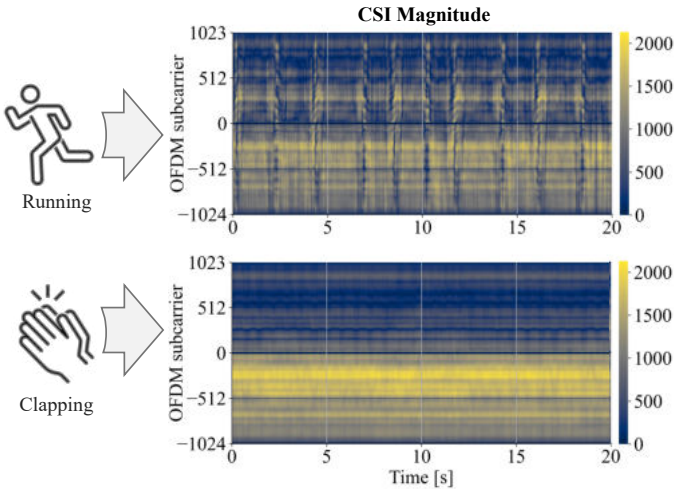


Figure 1: Magnitude of the CSI collected by the same antenna when a person performs two different activities, namely running (top) and clapping (bottom). CSI values are dimensionless and are reported as measured by the Wi-Fi chipset.

I. Introduction

Wi-Fi devices can be used as *passive radars* to recognise specific human activities in indoor environments because of the physical interaction of the human body with communication signals [1]. In Wi-Fi, the channel state information (CSI) is a complex-valued vector computed at the receiver for every incoming frame that measures the wireless channel's properties and equalises the received signal. However, the CSI also provides an electromagnetic fingerprint of the environment.

Figure 1 (top) illustrates a snippet of the CSI captured by one single antenna while a person runs. As the person moves around the room, the environment's effect on the signal changes due to the varying scattering on the human body. The result is captured in a *spectrogram* that highlights how the relative intensity of the signal changes over time and frequency. The fundamental assumption of CSI-based human activity recognition (HAR) is that it is possible to trace these variations back to the human activity that caused them, and in particular, to distinguish different types of activities, like running instead of standing still and clapping, *cf.*, Fig. 1 (bottom). However, such Wi-Fi sensing applications employ techniques that often lack the desired flexibility and transparency.

In this paper, we introduce DeepProbHAR, a neuro-symbolic approach to HAR using a passive Wi-Fi radar, providing initial evidence that Wi-Fi signals can differentiate between simple movements, such as leg or arm movements, which are integral to human activities like running or walking. DeepProbHAR builds on top of a recent [2] pre-processed dataset of human activities sensed through commercial Wi-Fi devices [3], that makes use of Variational Auto-Encoders (VAEs) [4] to identify a generative latent distribution seen as a compressed view of the original CSI signal. Specifically, the paper discusses (Section II) the differences between the two main approaches to HAR: declarative and data-driven. Declarative approaches provide classification rules for defining activities but struggle with unstructured data — indeed, to our knowledge, they have not been proposed for CSI data; while

data-driven approaches handle complex data types but are less flexible and more opaque. The paper expands on relevant references to the literature concerning data-driven approaches, including a description of our previous work published in [2], which provides a principled way to compress the CSI data and several architectures to fuse the signals coming from the different Wi-Fi antennas of the passive radar.

A third method for HAR, provided by neuro-symbolic approaches (Section III), combines symbolic (declarative) reasoning techniques and neural network (NN) methods, aiming to improve the performance of AI systems [5]. Neuro-symbolic systems can merge the approximation capabilities of NNs with the abstract reasoning abilities of symbolic methods, enabling them to extrapolate from limited data and produce interpretable results. In particular, we extract declarative knowledge (a decision tree) of the human activities from a video feed captured by a camera observing the same environment as the Wi-Fi receiver senses. We then transfer such knowledge to train with just the label of the activity DeepProbHAR, a neuro-symbolic architecture seeing different data modality (the CSI) and that builds on top of DeepProbLog [6], [7].

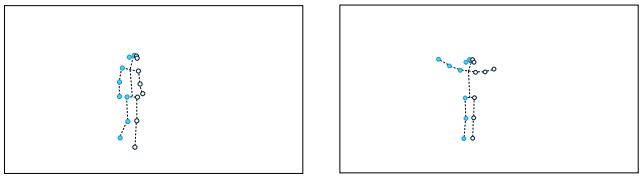
Our experimental results (Section IV) demonstrate that DeepProbHAR achieves comparable if not better results than the state-of-the-art approaches while at the same time reaching a higher degree of transparency. In particular, DeepProbHAR can identify occurrences of simple movements (such as *moving the upper arm*) without requiring specific labelling for such concepts. Finally, Section V highlights the ability to distinguish simple movements of the subject.

II. Background

A. Activity Recognition Approaches

There are two primary approaches to activity recognition: *declarative* and *data-driven*. Declarative approaches [8]–[10] provide classification rules that can be utilised to define the activity. An example of such a rule — in natural language — could be: *running is the rapid, alternating action of pushing off and landing on the ground with one's feet*.[1] However, the types of input data declarative approaches can handle are often limited. Specifically, declarative rules typically require direct processing of the input data, which can pose challenges for unstructured data such as Wi-Fi CSI (further discussed in Section II-B). Indeed, the authors are unaware of any declarative approaches for HAR operating over CSI data.

Data-driven approaches (*e.g.*, [2], [11]–[13]) are specifically designed to handle data types for which it is difficult to define rules directly. Despite their advantages, these approaches are more opaque and less flexible than their declarative counterparts, often making it impossible for the system's end-user to define patterns entirely. Indeed, several HAR systems work by deriving some physically-related quantity from some sensors (*e.g.*, the CSIs) that is then used to train a deep learning classification system [11]–[13]. In a previous work, we showed a principled approach to HAR using a VAE generative model to compress the sensors' information and various architectures for fusing multiple antennas' signals [2].



a) walking b) waving both hands

Figure 2: Sample of the video dataset for two different activities: a) *walking* and b) *waving both hands*. The key points in every video frame help to discern the right side (highlighted with coloured dots) from the left side of the candidate.

B. Dataset

In this work, we rely on a CSI dataset publicly released by members of the author list.[2] Further details about the dataset are available in [3]. The experimental testbed comprises two Asus RT-AX86U devices placed on opposite sides of an approximately 46-square-metre room. One device generates dummy IEEE 802.11ax (Wi-Fi 6) traffic at a constant rate of 150 frames per second using the frame injection feature in [14]. The other device (also called *monitor*) receives the Wi-Fi frames and stores the associated CSI for each of its four receiving antennas independently. Meanwhile, one candidate performs different activities in the middle of the room.

The CSI dataset is coarsely synchronised with a video recording of the activities, collected using a smartphone camera placed in a fixed location and then anonymised. Specifically, to preserve the participants' identity, VideoPose3D [15] was used to extract a model of the candidate performing the activities. VideoPose3D identifies 17 key points to track the motion of the main human joints, as shown in Fig. 2. The key points are stored as a list of (x, y) coordinates in the camera viewport for each video frame. Even though the dataset includes the CSI data of twelve different activities, there are seven activities in total for which both CSI and video data are available: *walking, running, jumping, squatting, waving both hands, clapping,* and *wiping*. For each activity, the dataset contains 80 seconds of CSI data (sampled at 150 CSI per second) and the corresponding video data (*i.e.*, the key points of the candidate, sampled at 30 fps). VideoPose3D can also reconstruct a 3D model of the candidate using a deep learning algorithm; however, we found some numerical instability in the 3D coordinates reconstructed by the tool. Hence, in this work, we only consider the 2D coordinates of the joints extracted from the original video traces.

C. Dataset Pre-Processing and VAE Architectures

The work in [2] introduced several modular architectures for HAR using CSI data, practically splitting the problem into two separate sub-tasks. First, a VAE provided a concise (yet informative) characterisation of the different activities as

[1] Microsoft Copilot on 14th March 2024.

[2] https://github.com/ansresearch/exposing-the-csi

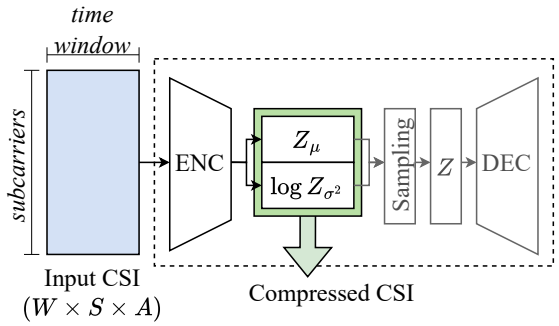


Figure 3: The VAE models map the input CSI data onto the four parameters of a bivariate Gaussian distribution (mean Z_μ and variance Z_{σ^2} along two axes), which we can be used as a compressed representation of the input CSI.

perceived by the Wi-Fi monitor through the CSI. Specifically, the VAE mapped short sequences of CSI data — sampled using a sliding window in the time domain — onto a latent bivariate Gaussian distribution defined by 4 parameters (*i.e.*, mean and variance along two dimensions). Second, a multi-layer perceptron (MLP) trained on the latent space parameters of the VAE was used to classify the different activities. The unsupervised training of the VAE can be carried out separately from the training of the MLP. However, since the Wi-Fi monitor is sensing the environment using four physically-spaced antennas, several VAE architectures were proposed in [2] to evaluate different strategies to fuse the information sensed by every antenna. Figure 3 summarises the general architecture of the VAEs. The input data structure is a tensor of size $(W \times S \times A)$, where W is the number of CSI samples in a given time window, S is the number of subcarriers in a Wi-Fi frame, and A is the number of antennas considered by the VAE. We fixed the time window to 3 seconds, so W=450, while the considered dataset contains Wi-Fi 6 frames with S=2048 subcarriers (160-MHz bandwidth).

In this work, we start from the compressed representation of the CSI data windows in the VAE's latent space to develop a neuro-symbolic architecture for HAR. This pre-processed dataset was obtained by training the VAEs described in [2] on the complete set of activities in the original dataset. Here, we briefly report the resultant architectures for convenience.

First, we consider a set of architectures called **No-Fused-x**. These architectures have been trained using the data incoming from one single antenna of the Wi-Fi monitor (A=1 in Fig. 3). We denote with the letter x the antenna of the monitor whose data we used to train the VAE. Hence, we define four separate architectures, one for each antenna: **No-Fused-1**, **No-Fused-2**, **No-Fused-3**, and **No-Fused-4**.

While using data from one single antenna can be enough for some HAR applications [16], we proved in previous work that there are consistent advantages in fusing the CSI data from different antennas [2]. Therefore, we also consider a second type of architecture, called **Early-Fusing**. In this case, the CSI data from all four monitor antennas are stacked together in the input data structure (A=4 in Fig. 3). Still, the latent space of VAE-F has a bivariate latent normal distribution which can condense together cross-antennas regularities.

The last architecture we consider, called **Delayed-Fusing**, employs all the four VAEs trained independently on each monitor antenna and concatenates their latent space parameters into a single vector of 16 elements (4 features for each VAE) that becomes the input of the following classification stage.

III. Methodology

In this section we present DeepProbHAR, the first neuro-symbolic approach to human activity recognition fusing information from Wi-Fi CSIs. First, we briefly introduce neuro-symbolic AI (Section III-A), particularly the DeepProbLog approach [6], [7]. Then, we discuss how we extracted domain-dependent knowledge for classifying different activities using a more interpretable modality, *viz*. the video recording of the performed activities (Section III-B). For this work, we relied on such a knowledge extraction to reasonably assume that the Wi-Fi sensor could have captured the way we would describe activities, as both the camera and the antennas were looking at the same environment. Finally, we describe in detail the DeepProbHAR architecture (Section III-C).

A. Primer on Neuro-Symbolic AI: the DeepProbLog Approach

Neurosymbolic AI, *e.g.*, [5], is often referred to as the combination of symbolic reasoning techniques and neural network methods to improve the performance of AI systems. These systems can merge the robust approximation capabilities of neural networks with the abstract reasoning abilities of symbolic methods, enabling them to extrapolate from limited data and produce interpretable results. Neurosymbolic AI techniques can be broadly categorised into two groups. The first considers techniques that condense structured symbolic knowledge for integration with neural patterns and reason using these integrated neural patterns. The second considers techniques that extract information from neural patterns to facilitate mapping to structured symbolic knowledge (*i.e.*, *lifting*) and carry out symbolic reasoning.

This paper focuses on a specific approach within the second group of neurosymbolic AI techniques, *viz*. DeepProbLog [6], [7]. To present it, we first need to briefly introduce ProbLog, [17], which is a probabilistic logic programming language. A ProbLog program comprises a collection of probabilistic facts, denoted as F, and a set of rules, denoted as R. Facts are expressed in the form $p :: f$, where f is an atom symbolising a notion that can either be true or false. p is a probability value ranging from 0 to 1, which signifies the probability of the fact being true. Rules are expressed in $h \leftarrow b_1, ..., b_n$, where h is an atom and b_i are literals. A literal can be an atom or the negation of an atom.

ProbLog includes Annotated Disjunctions (ADs) as a syntactic extension of the form $p_1 :: h_1; ...; p_n :: h_n \leftarrow b_1, ..., b_m$. where the p_i are probabilities such that $\sum p_i = 1$, and h_i and b_j are atoms. It is immediate to see that they encode categorical distributions over the possible results of a random

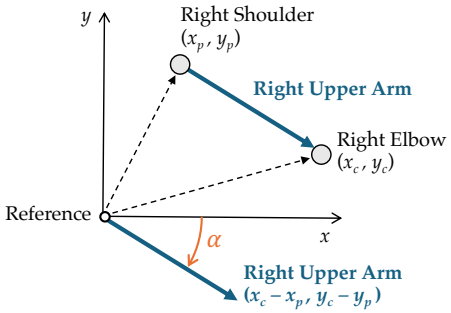


Figure 4: Computation of the right upper arm's angle α. The same operation applies to all the other limb segments.

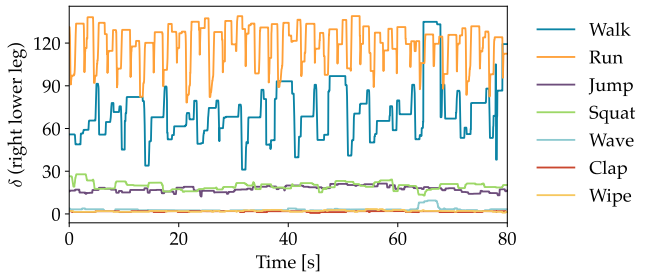


Figure 5: The feature δ_l corresponding the right lower leg indicates the motion of that limb for each target activity.

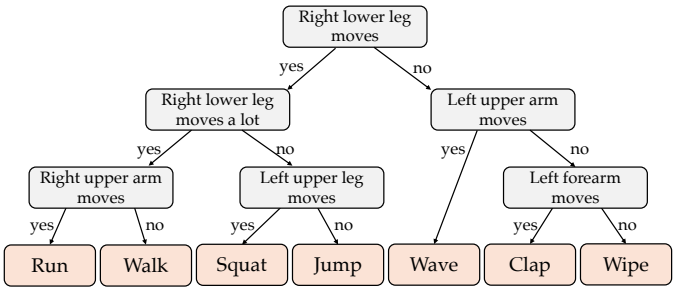


Figure 6: Decision tree derived from the video data analysis. Leaf nodes represent the target activities. At every decision node, a single feature δ_l is tested against a threshold to determine whether the corresponding limb is moving.

variable that can take on one of K possible categories. Moreover, it is also a way to capture a Bernoulli distribution where we wish to name the two outcomes explicitly, *e.g.*, *head* and *tail* as possible results of tossing a coin, rather than limiting ourselves to just one, let us say *head*, and deriving the other — *i.e.*, *tail* — as the negation.

A Problog program can be encoded in a probabilistic circuit [18], which is a graphical model that compactly represents probability distributions. Each fact and rule in a Problog program can be translated into a node or a set of nodes in a probabilistic circuit, and the probabilities associated with the facts correspond to the parameters of the probabilistic circuit.

DeepProbLog [6], [7] is a programming language that combines neural networks with probabilistic logic. It extends ProbLog by introducing Neural Annotated Disjunctions (nADs). Differently from ADs, in nADs the probabilities of the categorical distribution are the output layer of a neural network $f(\boldsymbol{x}, \boldsymbol{\theta})$. There are no restrictions on the form of $f(\cdot)$ as long as it outputs a categorical distribution over K classes, *e.g.*, using a *softmax* activation function at the network output. Each nAD is thus associated to a specific neural network $f(\cdot)$

For each nAD, DeepPropLog computes the gradient of the loss w.r.t. the output of the associated $f(\boldsymbol{x}, \boldsymbol{\theta})$. Standard backpropagation algorithms use the gradient to train the parameters $\boldsymbol{\theta}$. Such a computation leverages the differentiability of the ProbLog program, the computational machinery of which can be expressed over the associated probabilistic circuits. For further details, the interested reader is referred to [7] and for applications of DeepProbLog to analogous tasks such as complex event processing, to [19].

B. Domain-Dependent Rule from Different Modality

To operate using a neuro-symbolic approach, we must use some declarative knowledge to describe the activities we plan to classify. We assume that every target activity can be defined by combining basic movements; these "atomic" movements should be sufficiently easy to identify. For instance, *running* mandates a rapid alternate motion of the legs accompanied by broad arms' movement; conversely, *clapping* can be defined by the repetitive motion of the forearms. Building on this assumption, we can combine a limited subset of basic movements into a potentially large set of target activities.

In the following, we describe the process of extracting indicators of simple movements from the video dataset introduced in Section II-B. The idea is to compute a vector of scalar values (one for each limb) that characterise the *amount of motion* of the limbs in a given time window. Such values will serve as numerical features that can be combined and used to describe more complex target human activities.

A brief analysis of the video dataset revealed that we have access to the (x, y) coordinates of 17 key points in the camera viewport for each video frame. Specifically, 12 key points correspond to the person's shoulders, elbows, hands, hips, knees, and feet (one for each side, *cf.*, Fig. 2). Given the location of the joints and using basic trigonometry, we can precisely locate the limbs' position and orientation in the 2D frame. We consider $L = 8$ limb segments: two upper arms, lower arms, upper legs, and lower legs. Each segment is defined by two joints, a parent joint p (*e.g.*, the shoulder for the upper arm limb) and a child joint c (*e.g.*, the elbow for the upper arm). Figure 4 shows that if a parent joint p has coordinates (x_p, y_p) and a child joint c has coordinates (x_c, y_c), then the limb segment vector draws an angle $\alpha = \arctan\left(\frac{y_c - y_p}{x_c - x_p}\right)$ which changes at every frame depending on its motion.

We index the different limb angles with α_l, where l can range from 1 to L. To quantify the motion of each limb, we consider a sliding time window of duration $T = 3$ seconds, matching the time window already applied on the CSI data processed by the VAEs. Since the video was recorded at 30 fps, every time window contains 90 samples of the angles α_l. Then,

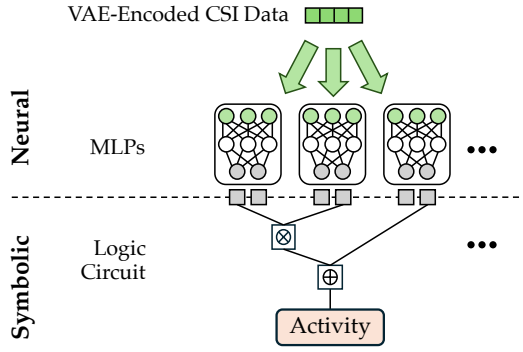


Figure 7: Architecture of DeepProbHAR. The neural part extracts *concepts* from the compressed CSI, while a logic circuit combines such *concepts* to derive the target activity.

for each time window t, we compute the dynamic range of every angle α_l. The result is a feature vector $\boldsymbol{\delta}(t)$ with L elements describing the amount of motion of each limb during a time window. Let us assume that a limb was not moving during the time window t (*e.g.*, one leg when the candidate is clapping); then, the associated α_l is barely changing, and $\delta_l(t)$ will be minimal. On the other hand, if we look at the same leg when the person is walking, we will see that α_l describes broader trajectories, and $\delta_l(t)$ will be much more significant.

The computation of the dynamic range of the angles α_l provides a quick estimation of each limb's motion amount. It is a simple way to discriminate the target activities in our dataset. In Fig. 5, we show how just the feature δ_l (where l is the right lower leg) is enough to separate three families of activities. The first family involves broad movements of the lower legs (*walking* and *running*); the second one involves modest movements of the lower legs (*jumping* and *squatting*); finally, the last family of activities consists of no movement of the leg at all (*waving*, *clapping*, and *wiping*). Arguably, the limbs' motion can be estimated using more sophisticated approaches, *e.g.*, involving a Fourier analysis of the angles α_l to identify periodic patterns or taking into account the perspective effects. However, we argue that taking the dynamic range is a reasonably simple and effective solution.

The video dataset analysis and the features extracted from the limbs' motion resulted in the decision tree shown in Fig. 6. This rule-based classification combines six *concepts* extracted from the dataset that can take binary values. At every decision node, one single value of the $\boldsymbol{\delta}$ vector is compared against a threshold that determines whether the corresponding limb is moving. Note that, in general, we are not bound to consider just binary decisions; indeed, according to what we already observed in Fig. 5, the first node at the top of the tree in Fig. 6 could have had three outputs. However, we forced the model to have only binary decisions to simplify the implementation of our neuro-symbolic architecture. In this way, the same feature is tested twice against two different thresholds (*cf.*, *Right lower leg moves* and *Right lower leg moves a lot* in Fig. 6).

C. DeepProbHAR: A Neuro-Symbolic Architecture for Human Activity Recognition Using Wi-Fi Data

In Section III-B, we have defined the declarative knowledge necessary to identify the different target activities and a simple way to extract the information from the video data. This section introduces DeepProbHAR, the first neuro-symbolic system designed explicitly for HAR applications using Wi-Fi sensing data.

A schematic overview of the proposed architecture is shown in Fig. 7. The system takes in input the compressed CSI data encoded by one of the VAEs introduced in Section II-C. This implies that we can define several architectures depending on the VAE used to pre-process the dataset. We consider four single-antenna architectures No-Fused-x (x ranging from 1 to 4) and two architectures fusing the data from multiple antennas, namely Early-Fusing and Delayed-Fusing (*cf.*, Section II-C). All the DeepProbHAR architectures employ six small MLPs to extract binary symbols from the input CSI data, which are combined using logic rules to estimate the target activity. Every MLP contains only two hidden layers with 8 neurons each, activated using a ReLU function, and a binary output layer with a SoftMax activation function.

The DeepProbLog code used to combine the neural networks' output with the logic part of the architecture is listed in Fig. 8a. The DeepProbLog program defines the six MLPs such that each MLP should correspond to a different decision node in Fig. 6. Then, a list of predicates implements the rules described in the decision tree constructed starting from the video data analysis. Figure 8b shows an example of the logic circuit derived from the DeepProbLog code, where the grey rectangles correspond to the probabilistic facts identified by the neural networks `net1` and `net5` and the red rectangle corresponds to the query defined by the formula on line 11 of the code listing. The white box with the $\otimes$ symbol represents the logical operator `AND` applied to its children.

IV. Experimental Results

We now evaluate the classification performance of all the DeepProbHAR models on the selected public dataset, which comprises both the CSI data and anonymised video recordings [3]. First, we certify that the rules extracted from the video dataset yield good accuracy in estimating the target activities (Section IV-A). Then, we measure the classification accuracy of the DeepProbHAR models (Section IV-B). Since DeepProbHAR leverages declarative knowledge gathered for a different modality (the video data feed), we expect such a reference to be the upper limit that DeepProbHAR can reach. Finally, we compare the results obtained with the neuro-symbolic architecture with those obtained by more traditional approaches based on neural networks[3].

A. Validation of Declarative Knowledge

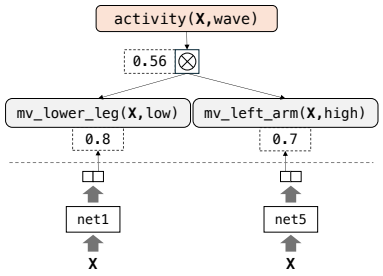
To ensure that the declarative knowledge gathered from the video data is enough to produce sensible guesses about

[3]Code available: https://github.com/marcocominelli/csi-vae/tree/fusion2024

```
nn(net1, [X], Y, [yes, no]) :: mv_lower_leg(X,Y).
nn(net2, [X], Y, [yes, no]) :: mv_lower_leg_alot(X,Y).
nn(net3, [X], Y, [yes, no]) :: mv_right_arm(X,Y).
nn(net4, [X], Y, [yes, no]) :: mv_upper_leg(X,Y).
nn(net5, [X], Y, [yes, no]) :: mv_left_arm(X,Y).
nn(net6, [X], Y, [yes, no]) :: mv_forearm(X,Y).

activity(X,walk) :- mv_lower_leg(X,yes), mv_lower_leg_alot(X,yes), mv_right_arm(X,no).
activity(X,run)  :- mv_lower_leg(X,yes), mv_lower_leg_alot(X,yes), mv_right_arm(X,yes).
activity(X,squat):- mv_lower_leg(X,yes), mv_lower_leg_alot(X,no), mv_upper_leg(X,yes).
activity(X,jump) :- mv_lower_leg(X,yes), mv_lower_leg_alot(X,no), mv_upper_leg(X,no).
activity(X,wave) :- mv_lower_leg(X,no), mv_left_arm(X,yes).
activity(X,clap) :- mv_lower_leg(X,no), mv_left_arm(X,no), mv_forearm(X,yes).
activity(X,wipe) :- mv_lower_leg(X,no), mv_left_arm(X,no), mv_forearm(X,no).
```

(a) The DeepProbLog program.



(b) Logic circuit for the query `activity(X,wave)`.

Figure 8: (a) DeepProbLog code implementing the neuro-symbolic architecture and (b) representation of the logic circuit for the query `activity(X,wave)`. Notice we drop the left/right distinction when features are uniquely defined in the code.

Table I: Accuracy and average precision, recall and F1 score of DeepProbHAR with different data fusion strategies. The accuracy of the extracted declarative knowledge tested over the video feed is provided as *Video reference* and is the upper limit of DeepProbHAR's performance.

Architecture	Accuracy	Precision	Recall	F1
Video reference	*0.98*	*0.99*	*0.98*	*0.98*
No-Fused-1	0.50	0.49	0.51	0.50
No-Fused-2	0.62	0.62	0.63	0.62
No-Fused-3	0.53	0.53	0.51	0.52
No-Fused-4	0.76	0.76	0.77	0.77
Early-Fusing	0.84	0.84	0.85	0.84
Delayed-Fusing	**0.95**	**0.95**	**0.95**	**0.95**

Table II: Comparison with state-of-the-art non-neuro-symbolic approaches using a single MLP trained on the corresponding dataset of each different architecture.

Architecture	DeepProbHAR	Small MLP	Large MLP
No-Fused-1	0.50	0.59	0.62
No-Fused-2	0.62	0.74	0.76
No-Fused-3	0.53	0.58	0.59
No-Fused-4	0.76	0.78	0.80
Early-Fusing	0.84	0.88	0.89
Delayed-Fusing	**0.95**	**0.94**	**0.98**

the target activity, we implemented the rule-based classifier introduced in Fig. 6, using manually fine-tuned thresholds on the angles' features $\boldsymbol{\delta}$) extracted directly from the video data. Such a classifier operating over the video feed achieves *an accuracy of 98%* over the seven target activities. Interestingly, classification errors happen between the activities *walk* and *run*. Arguably, these are the most challenging activities to discriminate, even for a human viewer, mainly because of the indoor experimental setting.

B. Performance of DeepProbHAR

To evaluate the performance of the different architectures, we partition every compressed CSI dataset into a training and a testing set with an 80/20 split. In the following, all the models are trained with a learning rate of 0.001 for 20 epochs.

Table I summarises the main results of the DeepProbHAR architectures for all the fusion strategies considered in [2] (*cf.*, Section III-C). Similarly to [2]'s results, the Delayed-Fusing fusion strategy yields the best results. The model's accuracy that combines the data of different antennas, each processed by a separate VAE, closely approaches the accuracy of the reference classifier trained on video data. However, as highlighted by the confusion matrixes of DeepProbHAR for the various data fusion techniques (Figure 9), classification errors are not limited to the classes *walk* and *run*.

In Table II, we compare the results with two state-of-the-art non-neuro-symbolic architectures derived from the work in [2]. These architectures use the same VAE and one single MLP substituting the entire neuro-symbolic architecture. In the neuro-symbolic architecture, each of the six MLPs learning a separate feature has 130 parameters (226 for the Delayed-Fusing approach). If we try to approximate the neuro-symbolic architecture using a single *small MLP* (2 hidden layers, 8 neurons each), the resulting model has 175 parameters (271 for the Delayed-Fusing approach). Arguably, since the neuro-symbolic architecture features six MLPs, it would be interesting to consider a *large MLP* (2 hidden layers, 22 neurons each) whose number of parameters closely matches the one of all the neuro-symbolic MLPs. The results in Table II reveal that the neuro-symbolic architectures perform worse than the single MLPs when considering just one antenna, but their accuracy becomes similar when fusing the data from multiple antennas. We also highlight that the models in [2] were evaluated on different activities, so the corresponding MLPs have been trained from scratch in this work.

V. Discussion

Table III (top) shows the results of the six MLPs that have been trained in DeepProbHAR, one for each feature as illustrated in Figure 6. For the sake of comparison, we also trained independently six MLP over a finely labelled dataset of simple movements — it is worth mentioning that labelling such a dataset could be extremely costly in less controlled settings — so that each MLP was optimised to classify only one of the relevant features, see Table III (bottom).

We wish to point out three aspects. First, the performance of each of the DeepProbHAR's MLP trained on sparse data (top of the table) appears to be close to the optimised MLPs trained over the finely-labelled dataset.

Table III: Classification accuracy of the various DeepProbHAR's MLPs compared to specialised MLPs trained on a finely labelled dataset of simple movements.

DeepProbHAR's MLPs							
Architecture	**MLP 1 (right lower leg)**	**MLP 2 (right lower leg #2)**	**MLP 3 (right arm)**	**MLP 4 (left upper leg)**	**MLP 5 (left arm)**	**MLP 6 (left forearm)**	**Overall Accuracy**
No-Fused-1	0.80	0.79	0.65	0.84	0.67	0.84	0.50
No-Fused-2	0.82	0.74	0.97	0.98	0.89	0.93	0.62
No-Fused-3	0.86	0.66	0.72	0.70	0.88	0.97	0.53
No-Fused-4	0.87	**0.98**	0.92	0.87	0.88	**1.00**	0.76
Early-Fusing	0.94	**0.98**	0.85	0.99	0.88	0.99	0.84
Delayed-Fusing	**0.99**	**0.98**	**0.96**	**1.00**	**0.96**	**1.00**	**0.95**
MLPs trained independently on finely labelled dataset of simple movements							
Architecture	**MLP 1 (right lower leg)**	**MLP 2 (right lower leg #2)**	**MLP 3 (right arm)**	**MLP 4 (left upper leg)**	**MLP 5 (left arm)**	**MLP 6 (left forearm)**	**Overall Accuracy**
No-Fused-1	0.83	0.81	0.68	0.84	0.85	0.90	0.59
No-Fused-2	0.87	0.82	0.97	0.99	0.90	0.93	0.70
No-Fused-3	0.89	0.68	0.74	0.70	0.90	0.99	0.58
No-Fused-4	0.90	0.98	0.92	0.88	0.88	**1.00**	0.79
Early-Fusing	0.96	0.98	0.87	**1.00**	0.89	**1.00**	0.86
Delayed-Fusing	**1.00**	**0.99**	**0.98**	**1.00**	**0.98**	**1.00**	**0.98**

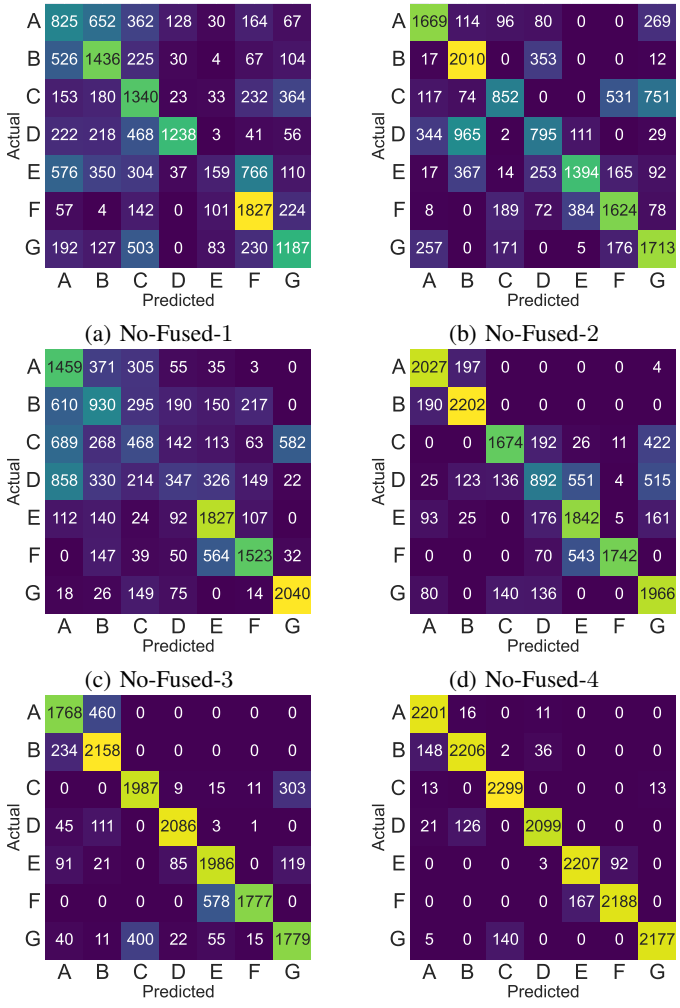
Secondly, the overall accuracy (last column) is lower than each of the accuracies for all the MLPs. This is due to the independence assumptions underlying the training of all such neural networks and their subsequent usage for classification, whether via DeepProbHAR (which relies on the same strong independence assumptions of ProbLog [18], [20][4]) or by a deterministic classifier that follows the decision tree in Fig. 6. Indeed, given a neural network $f(\cdot)$, its accuracy can be seen as the probability of $f(\cdot)$ to return the correct answer for a given input. In Fig. 6, for instance, we see that classifying *running* and *walking* relies on three of the classifiers whose accuracies are available in Table III: **MLP 1** that tells if the right lower leg moves; **MLP 2** that tells if the right lower leg moves a lot; and **MLP 3** that tells if the right upper arm moves. Under independence assumptions, the probability of correct classification of *running* and *walking* is the product of the probability that each of the three classifiers returns a correct answer. The average of the probabilities of correct classification for each of the activities as the product of the probabilities of correct classifications for the MLPs used for such a classification according to Figure 6 amounts to the same overall accuracy as computed in the last column of Table III.

Finally, we observe that some MLPs (*e.g.*, **MLP 4**) reach 1.00 accuracy. From Figure 6, such a MLP is responsible for distinguishing between *squatting* and *jumping*. A quick inspection of the confusion matrixes (Figure 9) reveals that such accuracy is the product of perfect split among those two classes over the test set, indicating that it can be relatively easy for an MLP to separate the remaining two activities.

VI. Conclusion

We introduced DeepProbHAR, a novel neuro-symbolic fusion approach to HAR, and provided initial evidence that

[4]For a more comprehensive discussion on the role of probabilistic dependencies among variables in probabilistic circuits — including those derived from ProbLog — we refer the interested reader to [21].



(a) No-Fused-1 (b) No-Fused-2

(c) No-Fused-3 (d) No-Fused-4

(e) Early-Fusing (f) Delayed-Fusing

Figure 9: Confusion matrixes of DeepProbHAR for the different fusion strategies. Activities are labelled as: A) Walk; B) Run; C) Squat; D) Jump; E) Wave; F) Clap; G) Wipe.

simple movements, such as leg or arm movements, which are integral to human activities like running or walking, can be discerned using Wi-Fi signals. Leveraging declarative knowledge of several activities extracted from a video feed, DeepProbHAR achieves results comparable to the state-of-the-art in CSI-based HAR. Moreover, as a by-product of the learning process, DeepProbHAR generates specialised classifiers for simple movements whose accuracy is on par with that of models trained on finely labelled datasets with a much higher cost. However, we expect that as the complexity of the events to be detected increases, the neuro-symbolic approach can even outperform only-neural techniques [22].

In future work, we will examine the efficacy of discerning simple movements when categorising unseen activities, *e.g.*, *parkour*. We shall also evaluate whether the inductive bias provided by the symbolic part enables learning with smaller training dataset sizes w.r.t. other state-of-the-art HAR models. Second, we intend to utilise the declarative knowledge of DeepProbHAR to explain the latent space of the VAEs employed as input. This will help us better comprehend the underlying structure and distribution of the data, potentially leading to more precise and efficient models. Third, we will consider an Evidential Deep Learning (EDL) [23], [24] approach to enhance robustness against out-of-distribution data, thereby improving the generalisability and reliability of our models, also across different indoor locations.

Acknowledgments

This work was partially supported by the European Office of Aerospace Research & Development (EOARD) under award number FA8655-22-1-7017 and by the US DEVCOM Army Research Laboratory (ARL) under Cooperative Agreements #W911NF2220243 and #W911NF1720196. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States government.

This work was conducted while M. Cominelli was affiliated with the DII, University of Brescia, Italy.

The authors thank Dr. Marc Roig Vilamala for his help in debugging part of the DeepProbLog code used in this work.

While preparing this work, the authors used GPT-3.5 and 4.0 to improve readability and language. After using them, the authors reviewed and edited the content as needed, and they take full responsibility for the publication's content.

References

[1] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, and K. Chetty, "Passive WiFi radar for human sensing using a stand-alone access point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, 2020.

[2] M. Cominelli, F. Gringoli, L. M. Kaplan, M. B. Srivastava, and F. Cerutti, "Accurate passive radar via an uncertainty-aware fusion of Wi-Fi sensing data," in *26th International Conference on Information Fusion (FUSION)*, 2023.

[3] M. Cominelli, F. Gringoli, and F. Restuccia, "Exposing the CSI: A systematic investigation of CSI-based Wi-Fi sensing capabilities and limitations," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023, pp. 81–90.

[4] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR2014*, 2014.

[5] A. Sheth, K. Roy, and M. Gaur, "Neurosymbolic artificial intelligence (why, what, and how)," *IEEE Intelligent Systems*, vol. 38, no. 3, 2023.

[6] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "DeepProbLog: Neural probabilistic logic programming," *Advances in neural information processing systems*, vol. 31, 2018.

[7] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt, "Neural probabilistic logic programming in DeepProbLog," *Artificial Intelligence*, vol. 298, 2021.

[8] H. Storf, M. Becker, and M. Riedl, "Rule-based activity recognition framework: Challenges, technique and learning," in *3rd International Conference on Pervasive Computing Technologies for Healthcare*. IEEE, 2009, pp. 1–7.

[9] P. Theekakul, S. Thiemjarus, E. Nantajeewarawat, T. Supnithi, and K. Hirota, "A rule-based approach to activity recognition," in *Proceedings of the 5th International Conference on Knowledge, Information, and Creativity Support Systems*. Springer-Verlag, 2010, pp. 204–215.

[10] M. Atzmueller, N. Hayat, M. Trojahn, and D. Kroll, "Explicative human activity recognition using adaptive association rule-based classification," in *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*. IEEE, 2018, pp. 1–6.

[11] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "SHARP: Environment and person independent activity recognition with commodity IEEE 802.11 access points," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2022.

[12] N. Bahadori, J. Ashdown, and F. Restuccia, "ReWiS: Reliable Wi-Fi sensing through few-shot multi-antenna multi-receiver CSI learning," in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022, pp. 50–59.

[13] J. Liu, H. Mu, A. Vakil, R. Ewing, X. Shen, E. Blasch, and J. Li, "Human occupancy detection via passive cognitive radio," *Sensors*, vol. 20, no. 15, 2020.

[14] F. Gringoli, M. Cominelli, A. Blanco, and J. Widmer, "AX-CSI: Enabling CSI extraction on commercial 802.11ax Wi-Fi platforms," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, 2021, p. 46–53.

[15] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7745–7754.

[16] S. Ding, Z. Chen, T. Zheng, and J. Luo, "RF-Net: A unified meta-learning framework for RF-enabled one-shot human activity recognition," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys)*, 2020, p. 517–530.

[17] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: A probabilistic prolog and its application in link discovery," in *IJCAI 2007, Proceedings of the 20th international joint conference on artificial intelligence*, 2007.

[18] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt, "Inference and learning in probabilistic logic programs using weighted boolean formulas," *Theory and Practice of Logic Programming*, vol. 15, no. 3, pp. 358–401, 2015.

[19] M. Roig Vilamala, T. Xing, H. Taylor, L. Garcia, M. Srivastava, L. Kaplan, A. Preece, A. Kimmig, and F. Cerutti, "DeepProbCEP: A neuro-symbolic approach for complex event processing in adversarial settings," *Expert Systems with Applications*, vol. 215, p. 119376, 2023.

[20] L. De Raedt, A. Dries, I. Thon, G. Van den Broeck, and M. Verbeke, "Inducing probabilistic relational rules from probabilistic examples," in *IJCAI*, 2015, pp. 1835–1843.

[21] F. Cerutti, L. M. Kaplan, A. Kimmig, and M. Şensoy, "Handling epistemic and aleatory uncertainties in probabilistic circuits," *Machine Learning*, pp. 1–43, 2022.

[22] L. Han and M. B. Srivastava, "An empirical evaluation of neural and neuro-symbolic approaches to real-time multimodal complex event detection," *arXiv preprint arXiv:2402.11403*, 2024.

[23] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential Deep Learning to quantify classification uncertainty," in *NeurIPS*, Jun. 2018.

[24] M. Sensoy, L. Kaplan, F. Cerutti, and M. Saleki, "Uncertainty-aware deep classifiers using generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 5620–5627.