RESEARCH ARTICLE

UniHAR: A unified hierarchical transformer with multi-scale temporal modeling for wearable human activity recognition

Memoona[1], Muhammad Haseeb Hashir[2], Sung Won Kim[3]*

1 Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, Republic of Korea, **2** Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, Republic of Korea, **3** School of Computer Science and Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, Republic of Korea

* swon@yu.ac.kr

Check for updates

OPEN ACCESS

Citation: Memoona, Hashir MH, Kim SW (2026) UniHAR: A unified hierarchical transformer with multi-scale temporal modeling for wearable human activity recognition. PLoS One 21(10): e0359774. https://doi.org/10.1371/journal.pone.0359774

Editor: Noman Naseer, Air University, PAKISTAN

Received: April 30, 2026

Accepted: September 16, 2026

Published: October 30, 2026

Copyright: © 2026 Memoona et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: The third-party datasets analyzed in this study are publicly available from the UCI Machine Learning Repository: UCI-HAR (https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones; DOI: https://doi.org/10.24432/C54S4K), OPPORTUNITY

Abstract

UniHAR is a lightweight hierarchical multi-scale transformer based on wearable sensor data that has to be able to model time well; be able to recognize human activity across different datasets; and do so in real-time. There are several problems with what currently exists. UniHAR uses four major components: universal channel harmonization; adaptive multi-scale patch embedding; scale-adaptive Fourier cross-attention; factorized temporal-channel attention. The components allow for the ability to use various sensors and enables the low-complexity multi-scale fusion of features. UniHAR achieved an accuracy of 96.35%; 94.45%; 96.44%; and 99.23% on UCI-HAR, Opportunity, Daphnet, and PAMAP2 datasets, while being very lightweight and having high-speed inference capabilities; therefore, UniHAR is suitable for real-time healthcare applications.

1 Introduction

Human activity recognition (HAR), is a method to automatically classify the type of activity that people are undertaking by analyzing data gathered from sensors through machine learning and signal processing [1,2]. The most common areas where this can be used are: medical monitoring [3–5], athletic analysis [6,7], fitness optimization [8–14], surveillance [15], and human-computer interaction [16]. There are two types of methods for HAR vision based and sensor based [16–19]; Both vision-based methods will gather information from optical streams [19], Sensor-based methods will utilize accelerometer, gyroscope and orientation sensors [19,20], and offer greater portability than vision systems [20].. Yet growing deep network complexity imposes prohibitive computational and memory costs in mobile and embedded settings [21,22], motivating lightweight architectures that preserve recognition accuracy [22].

Beyond this general efficiency concern, sensor-based HAR presents four domain-specific challenges that, taken together, motivate the design of UniHAR.

(https://archive.ics.uci.edu/dataset/226/opportunity+activity+recognition; DOI: https://doi.org/10.24432/C5M027), Daphnet Freezing of Gait (https://archive.ics.uci.edu/dataset/245/daphnet+freezing+of+gait; DOI: https://doi.org/10.24432/C56K78), and PAMAP2 Physical Activity Monitoring (https://archive.ics.uci.edu/dataset/231/pamap2+physical+activity+monitoring; DOI: https://doi.org/10.24432/C5NW2H). No new data were generated in this study. The source code used for the analysis is available in a public GitHub repository at https://github.com/memoonaqayyum/PLOS. In addition, all relevant details needed to reproduce the analysis are provided within the manuscript (Sections 3.8–3.10 and Tables 2–5).

Funding: This work was supported by the 2026 Yeungnam University Research Grant. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

First, HAR benchmarks differ widely in *sensor configuration*, so architectures built around a fixed channel-dependent weight matrix cannot be transferred across datasets without redesigning the input layer. Second, human motion is inherently *multi-scale*: fine-grained wrist or hand gestures coexist with coarse, whole-body postural transitions within the same activity window, so a model built around a single fixed temporal resolution systematically under-represents one class of motion at the expense of the other. Third, wearable signals such as gait cycles, pedaling cadence, and repetitive exercise are strongly *periodic*, a structure that standard time-domain self-attention does not exploit directly and must instead learn implicitly from data. Fourth, because HAR models are frequently intended for deployment on mobile or embedded devices, jointly modeling temporal dynamics and feature-channel dependencies must be done without incurring the quadratic cost of full self-attention, or the efficiency gains motivated above are lost in practice.

To address these four challenges jointly, we propose **UniHAR**, a hierarchical multi-scale Transformer built around four components, applied in sequence and each targeting one of the challenges described above: (i) **Universal Channel Harmonization (UCH)** projects heterogeneous sensor layouts into a unified, fixed-dimensional feature space using channel-count-agnostic per-channel descriptors, enabling cross-dataset generalization without dataset-specific redesign; (ii) **Adaptive Multi-Scale Patch Embedding (AMSPE)** encodes sequences at fine, medium, and coarse temporal resolutions whose patch sizes are tied to the physical sampling rate, so that fine- and coarse-grained motion are each represented at an appropriate scale rather than being forced into a single resolution; (iii) **Scale-Adaptive Fourier Cross-Attention (SAFCA)** combines FFT-based intra-scale mixing, which directly captures the periodic structure of wearable signals, with cross-scale attention that propagates context across the three temporal resolutions; and (iv) **Factorized Temporal-Channel Attention (FTCA)** factorizes attention across the temporal (token) dimension and the feature-channel (embedding) dimension, avoiding the quadratic cost of joint attention and yielding a form suitable for real-time, edge-deployed inference. Together, these components allow a single set of parameters to be trained once and applied across datasets with different channel counts, sampling rates, and activity taxonomies, without being re-initialized or re-trained per dataset. On UCI-HAR, Opportunity, Daphnet, and PAMAP2, UniHAR achieves 96.35%, 94.45%, 96.44%, and 99.23% accuracy within a 0.7–0.9M parameter footprint.

The contributions are:

- **UniHAR**: a hierarchical multi-scale Transformer for fine- and coarse-grained motion recognition, addressing the multi-scale nature of human activity.
- **Universal Channel Harmonization (UCH)**: a channel-agnostic projection enabling generalization across heterogeneous sensor configurations without architectural modification.
- **Efficient attention** via Scale-Adaptive Fourier Cross-Attention (SAFCA) and Factorized Temporal-Channel Attention (FTCA) for scalable multi-scale fusion at reduced complexity, addressing signal periodicity and the computational constraints of edge deployment.

- **State-of-the-art results** on four benchmarks with ablation and efficiency analysis.
- **Frequency-domain insights** showing Fourier processing is especially effective for periodic activities.

2 Related work

2.1 Traditional and deep learning approaches for wearable HAR

Early HAR based on sensors was primarily dependent on extracting hand-crafted statistical and frequency characteristics (e.g., mean, variance; FFT Coefficients; Zero-Cross Rate) from a fixed sliding window and classifying with traditional machine learning models (Decision Trees, SVM etc.) [23]. While these methods were relatively easy to interpret, they required domain knowledge for feature extraction and had poor generalization properties over sensor configurations.

The application of deep learning eliminated the need to manually design features by hand. Yang et al. [24] showed that a CNN operating directly on a multichannel raw signal time series could beat baseline systems engineered using predesigned features on the UCI-HAR database. Ordóñez and Roggen [25] employed both convolutional and bidirectional LSTM layers in a Deep Convolutional Long Short Term Memory (CNN-LSTM) model which learned joint spatial (inter sensor) and temporal information on the opportunity data set significantly increasing accuracy. The authors of Hammerla et al. [26], in their comparative study of deep, convolutional, and recurrent neural network models on four HAR benchmark data sets, showed that LSTM-based models were the most accurate consistently among all architectures tested. However, they also found that no model was best suited to all of the individual data sets; therefore, there is still an unmet need for universal (unified) design approaches. Subsequently, Guan et al. [27] expanded upon the use of recurrent models as ensembles as well as using residual formulations of these models which produced increased accuracy with reduced variability during model training. Recent InceptionTime-Style Models [28], which utilize Multi-Scale Convolutions Over Time-Series Data (to find patterns at varying levels of granularity), have been able to perform well by utilizing the various scales within one Convolution Block. Additionally, Ronald et al. [29], utilized iSPLInception (a variant of InceptionResNet) for performing strong results on the four benchmark datasets that were used in this study.

These advancements notwithstanding, there are fundamental limitations in both pure convolutional and Recurrent models. The first limitation is that using Local Receptive Fields with Fixed Length Kernels does not provide an efficient method of finding Long-Range Temporal Dependencies unless they are stacked very deeply. Secondly, each model type utilizes Dataset Specific Input Layers; therefore, if the Channel Count (C) changes among deployments, training or Structural Modifications would be required.

Beyond this general efficiency concern, a growing body of work targets the inference cost of deep HAR models directly rather than relying on generic complexity arguments [30]. proposed Rep-LOM, which fuses 1×1 convolutions with structural reparameterization and decoupled fully-connected (DFC) attention, collapsing a multi-branch training-time topology into a single-branch inference-time network to reduce FLOPs and parameters without sacrificing accuracy. Extending this direction [31], introduced Rep-MMB, which bridges mobile CNN and Transformer design to further improve the accuracy-efficiency trade-off of sensor-based HAR on resource-constrained hardware. Rather than statically shrinking the network [32], addressed inference cost through dynamic early-exit: a meta-learned sample-reweighting scheme aligns training-time loss weighting with the test-time behavior of multi-exit classifiers, letting easy activity samples exit early while harder samples are routed to deeper classifiers, which consistently improves the accuracy-cost trade-off over conventional early-exit training. These works illustrate that recent progress in efficient HAR increasingly relies on architectural reparameterization and adaptive inference rather than on hand-designed lightweight blocks alone; UniHAR's FTCA and SAFCA modules target the same computational bottleneck through a complementary sequence-factorization strategy rather than reparameterization or early exiting.

A parallel line of work sidesteps direct sequence modeling by first transforming the sensor time-series into a 2D image representation and then applying image-domain CNNs. These representations include time-dependent direction/position

images [33], cross-modal fusions of Markov transition fields, recurrence plots, and Gramian angular fields [34], relative position images derived from spatially distributed flexible sensor arrays [35], and wavelet scalograms combined with texture descriptors [36], with reported accuracies exceeding 99% on constrained gymnastics recognition using a dedicated 11-node sensor network [37]. These approaches show that engineered image transformations can substantially improve feature separability within a single, narrowly scoped activity domain. However, they are evaluated on task-specific rigs with fixed node counts and placements (e.g., an 11-node gymnastics network or a 16-channel lower-limb array) rather than the heterogeneous, cross-dataset channel counts addressed by UCH, and they rely on an explicit image-transformation preprocessing stage rather than operating on the raw multichannel sequence. UniHAR instead retains the sequence representation throughout, applying multi-scale tokenization and frequency-domain mixing directly to the raw signal without an intermediate image-transformation step.

2.2 Transformer models for time-series and HAR

The Transformer [38], based on multi-head self-attention, replaced recurrence with direct pairwise token interactions and became the dominant architecture in natural language processing. Adapting this mechanism to time-series data requires addressing the quadratic cost of attention over long sequences and the lack of positional inductive bias.

Zerveas et al. [39] introduced a Transformer-based framework for multivariate time-series representation learning, demonstrating that pre-training on masked inputs and fine-tuning on downstream tasks improves classification and regression performance substantially over supervised-only baselines. For long-sequence forecasting, Zhou et al. [40] proposed Informer, which replaces full self-attention with a sparse ProbSparse mechanism to achieve $O(N \log N)$ complexity, and Wu et al. [41] introduced Autoformer, which replaces attention with an auto-correlation mechanism over decomposed seasonal–trend components. Nie et al. [42] showed that segmenting time-series into non-overlapping patches before applying Transformer attention significantly reduces sequence length and improves long-range forecasting, a patching idea that directly motivates the multi-scale patch tokenization in AMSPE.

In the HAR domain specifically, Xu et al. [43] proposed LIMU-BERT, which pre-trains a BERT-style Transformer on raw IMU data with masked sensor modeling, achieving strong generalization across HAR benchmarks with limited labeled data. Tang et al. [44] proposed a multi-head self-attention module for activity recognition that weighs temporal segments adaptively, showing that attention-based pooling outperforms global average pooling for IMU sequences.

A parallel and more recent trend addresses HAR data scarcity and cross-dataset heterogeneity by adapting large pretrained Time-Series Foundation Models (TSFMs) rather than training HAR-specific backbones from scratch [45]. proposed DeepSenseMoE, a multi-scale convolutional Mixture-of-Experts module for parameter-efficient fine-tuning of general-purpose TSFMs on few-shot wearable activity recognition, unlocking general time-series representations for HAR despite severe annotation scarcity and sensor heterogeneity. In a related direction [46], introduced Sensor-Prompt Tuning, which aligns TSFMs with motion-sensor inputs through lightweight prompt-based adaptation for few-shot activity recognition, avoiding full fine-tuning of the foundation model. Both works reflect a paradigm shift toward reusing large-scale pretrained time-series representations for HAR, in contrast to UniHAR's from-scratch, channel-agnostic harmonization strategy (UCH), which addresses the same cross-dataset heterogeneity problem without relying on external large-scale pretraining.

Common to all of these works is a fixed-channel input assumption and the use of a single temporal resolution. UniHAR extends this line by operating simultaneously at three temporal scales and by introducing a channel-agnostic harmonization module that removes the fixed-channel constraint without dataset-specific redesign.

2.3 Multi-scale and hierarchical architectures

Human motion is inherently multi-scale: fine-grained wrist movements and coarse whole-body postural transitions coexist within a single activity window. Capturing this structure requires representations at multiple temporal granularities.

In computer vision, Liu et al. [47] introduced the Swin Transformer, which processes image patches in a hierarchical, shifted-window attention scheme, producing features at progressively coarser spatial resolutions. Fan et al. [48] extended this with Multiscale Vision Transformers (MViT), which pool spatial and channel dimensions across stages to model visual content at multiple scales. These hierarchical designs demonstrate that scale-aware representation learning improves recognition performance on structured signals.

For time-series, InceptionTime [28] applies parallel convolutions with kernels of different lengths before concatenating the resulting feature maps, capturing temporal patterns at multiple granularities within each layer. However, this approach applies fixed kernel sizes independent of the sampling rate and does not perform cross-scale information fusion across encoder layers.

UniHAR generalizes multi-scale design to wearable HAR by: (i) adapting patch sizes to the physical sampling rate rather than fixing them as hyperparameters, (ii) explicitly fusing information across scales at every encoder block via SAFCA and ISIE, and (iii) learning scale-specific attention weights at classification time rather than concatenating all scale outputs uniformly.

2.4 Frequency-domain methods for sequence modeling

Wearable sensor signals exhibit strong periodic structure—gait cycles, pedaling cadence, repetitive exercise—that is naturally represented in the frequency domain. Several recent works have incorporated Fourier processing into sequence models to exploit this structure.

Lee et al. [49] replaced multi-head self-attention in standard Transformers with unlearned Fourier transforms applied over both the sequence and embedding dimensions, achieving near-BERT-level accuracy on NLP benchmarks at a fraction of the computational cost. Zhou et al. [50] proposed FEDformer, which applies Fourier and wavelet transforms to decompose time-series into frequency components before attention, improving long-term forecasting while reducing complexity. Wu et al. [41] also leverage frequency-domain structure through their Auto-Correlation mechanism, which identifies recurrent dependencies by computing the FFT of the time-delay similarity function.

These works primarily target forecasting and NLP, not multi-sensor HAR, and do not combine frequency-domain intra-scale mixing with cross-scale attention. SAFCA in UniHAR unifies both: FFT-based token mixing captures periodic patterns within each scale, while cross-scale attention propagates context across temporal resolutions in a single efficient block.

2.5 Efficient attention mechanisms for long sequences

Standard self-attention scales as $O(N^2)$ in sequence length, motivating approximations for long-sequence inputs. Katharopoulos et al. [51] reformulated dot-product attention using feature-map kernel approximations, showing that the cumulative key-value product can be maintained in linear time and constant memory, yielding $O(N)$ complexity. This linear attention formulation is adopted in UniHAR's FTCA temporal attention sub-layer to keep per-token costs tractable.

2.6 Cross-dataset generalization in HAR

Cross-dataset generalization has received comparatively little attention in HAR. Most published models are evaluated on a single benchmark and assume fixed channel counts at training and test time. Wilson et al. [52] studied domain adaptation between HAR datasets with different sensor placements, demonstrating that naive fine-tuning degrades significantly without explicit alignment mechanisms. Haresamudram et al. [53] proposed contrastive predictive coding on unlabeled IMU sequences to learn transferable representations, reducing the gap between source and target distributions. Nevertheless, these approaches still require the input channel count to be identical between domains, or resort to channel selection/alignment steps that discard sensor channels not present in the target.

UniHAR addresses cross-dataset generalization at the architecture level through UCH, which learns a shared per-channel scorer that maps variable-length sensor inputs to a fixed-dimensional representation without dataset-specific re-training. To our knowledge, UniHAR is the first wearable HAR architecture designed to jointly address multi-scale temporal modeling, heterogeneous channel harmonization, and efficient edge deployment within a single unified framework.

3 Methodology

This section presents *UniHAR*, a single end-to-end trainable framework for wearable sensor-based HAR that operates across datasets with varying channel counts, sampling rates, and activity taxonomies without dataset-specific backbone modifications. The pipeline proceeds through four stages: (i) Universal Channel Harmonization (UCH), (ii) Adaptive Multi-Scale Patch Embedding (AMSPE), (iii) a Hierarchical Transformer Encoder (SAFCA–ISIE–FTCA blocks), and (iv) Multi-Scale Aggregation with Adaptive Classification. The overall architecture is illustrated in Fig 1.

3.1 Problem formulation

Let $\mathbf{X} \in \mathbb{R}^{T \times C}$ be a windowed wearable sensor segment with T time-steps, C channels, and label $y \in \{1, ..., K\}$. Given $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$, subject-independent splits evaluate cross-subject generalization. UniHAR must simultaneously accommodate three sources of inter-dataset heterogeneity: (i) temporal (T, f_s), (ii) sensory (C, ranging 9/113/9/18 across our benchmarks), and (iii) task (K, binary to 18-class). The objective is to learn

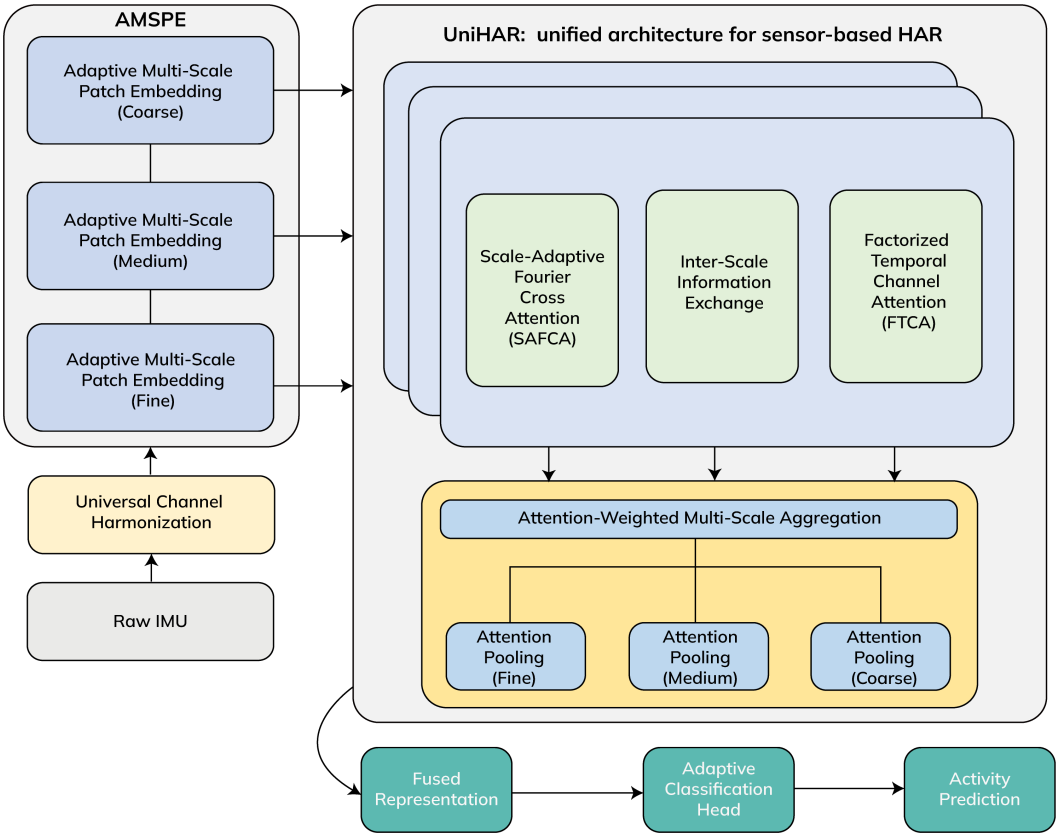


Fig 1. Overall architecture of the proposed framework.

https://doi.org/10.1371/journal.pone.0359774.g001

$$f_\theta : \mathbb{R}^{T\times C} \longrightarrow \mathbb{R}^K \tag{1}$$

whose parameters θ are not re-initialized or re-trained per dataset.

3.2 Universal channel harmonization (UCH)

Directly sharing a weight matrix $\mathbf{W} \in \mathbb{R}^{C\times d}$ across datasets with different channel counts C is impossible without modification. UCH resolves this by replacing the channel-count-dependent weight with a *shared per-channel scorer* that operates on a fixed-size descriptor computed independently for each channel, regardless of how many channels C the input contains.

3.2.1 Per-channel descriptor extraction. For each channel $c \in \{1, ..., C\}$, a fixed-length descriptor $\psi_c \in \mathbb{R}^{d_\psi}$ is computed from the channel's time-series $\mathbf{x}_c \in \mathbb{R}^T$ using a compact set of statistics that are well-defined for any T:

$$\psi_c = \left[\mu_c,\ \sigma_c,\ \gamma_{1,c},\ \gamma_{2,c},\ \mathbf{p}_c^\top\right]^\top \in \mathbb{R}^{d_\psi}, \tag{2}$$

where μ_c, σ_c, $\gamma_{1,c}$, $\gamma_{2,c}$ are the mean, standard deviation, skewness, and excess kurtosis of $\mathbf{x}_c$, and $\mathbf{p}_c \in \mathbb{R}^{d_\psi-4}$ are the top-(d_ψ–4) normalized spectral power bins of $\mathbf{x}_c$. We set d_ψ=16 in all experiments. We adopt statistical descriptors rather than a learned encoder to ensure robustness across heterogeneous sensor modalities without introducing dataset-specific feature extractors; the statistics are well-defined for any sensor type and require no pre-training. The descriptor is channel-count agnostic: the same d_ψ-dimensional vector is produced for any channel in any dataset.

3.2.2 Shared group-assignment scorer. A two-layer MLP with parameters shared across all datasets and all channels maps each descriptor to a soft group-membership vector:

$$\mathbf{a}_c = \mathrm{softmax}\big(\mathrm{MLP}_g(\psi_c)\big) \in \mathbb{R}^M, \tag{3}$$

where $\mathrm{MLP}_g : \mathbb{R}^{d_\psi} \to \mathbb{R}^M$ has fixed input and output dimensions regardless of C. Stacking all C channel assignments gives $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_C]^\top \in \mathbb{R}^{C\times M}$. Because the same MLP weights are used for every channel in every dataset, UCH shares all parameters across datasets and adapts only through the input descriptors, which vary with the actual sensor data.

3.2.3 Soft-weighted group projection. Each group embedding is a soft weighted sum over all channels:

$$\hat{\mathbf{X}}_m = \mathbf{X}\,\mathrm{diag}(\mathbf{A}_{:,m}) \in \mathbb{R}^{T\times C}, \quad m = 1, ..., M, \tag{4}$$

followed by a 1×1 Conv1D projection to d/M features:

$$\mathbf{H}_m = \mathrm{Conv1D}\left(\hat{\mathbf{X}}_m;\ \mathrm{out}=d/M\right) \in \mathbb{R}^{T\times d/M}. \tag{5}$$

3.2.4 Fusion. All group embeddings are concatenated and linearly fused:

$$\mathbf{X}_u = \mathrm{LayerNorm}\big(\mathbf{W}_f\,\mathrm{Concat}[\mathbf{H}_1, ..., \mathbf{H}_M]\big) \in \mathbb{R}^{T\times d}, \tag{6}$$

with $\mathbf{W}_f \in \mathbb{R}^{d\times d}$. The only parameter that changes shape with C is the Conv1D kernel in Eq. (5), which operates on the weighted input $\hat{\mathbf{X}}_m \in \mathbb{R}^{T\times C}$ but projects along the channel dimension independently per time-step. In practice, we implement this as a shared 1×1 convolution whose C-dimensional input is zero-padded to the maximum C seen during training; channels beyond the training maximum are assigned near-zero group weights by the scorer, making the padding effectively inactive. We acknowledge that this padding strategy imposes a soft upper bound on supported channel counts during training; supporting truly unbounded C at test time remains an open direction.

3.3 Adaptive multi-scale patch embedding (AMSPE)

Direct self-attention over raw time-steps costs $O(T^2)$ and treats all temporal granularities equally. AMSPE constructs three token sequences whose patch sizes are tied to the physical sampling rate, ensuring each scale corresponds to a meaningful motion time-scale.

3.3.1 Target-duration patch computation. For a given sampling rate f_s, the patch size for scale $s \in \{f, m, c\}$ is determined by a target patch duration τ_s in seconds:

$$p_f = \max\left(4,\ \lfloor \tau_f f_s \rfloor\right), \qquad p_m = \max\left(8,\ \lfloor \tau_m f_s \rfloor\right), \qquad p_c = \max\left(16,\ \lfloor \tau_c f_s \rfloor\right), \tag{7}$$

with τ_f=0.15 s, τ_m=0.35 s, τ_c=0.70 s. For example, at f_s=50 Hz: p_f=7, p_m=17, p_c=35 time-steps, corresponding to approximately 140 ms, 340 ms, and 700 ms motion windows. Strides are $\text{str}_s = \max(\lfloor p_s/2 \rfloor, 2)$ (50% overlap). The lower bounds in Eq. (7) prevent degenerate patches at very low sampling rates.

3.3.2 Token generation and encoding. For each scale s, a Conv1D layer produces $N_s = \lfloor (T - p_s)/\text{str}_s \rfloor + 1$ tokens:

$$\mathbf{P}^s = \text{Conv1D}\left(\mathbf{X}_u;\ \text{kernel}=p_s,\ \text{stride}=\text{str}_s,\ \text{out}=d\right) \in \mathbb{R}^{N_s \times d}. \tag{8}$$

Learnable positional embeddings $\mathbf{PE}^s \in \mathbb{R}^{N_s \times d}$, per-scale embeddings $\mathbf{SE}^s \in \mathbb{R}^{1 \times d}$ (broadcast), and a global temporal summary $\mathbf{TE} \in \mathbb{R}^{1 \times d}$ (mean of $\mathbf{X}_u$ projected to d) are summed:

$$\mathbf{Z}^s = \mathbf{P}^s + \mathbf{PE}^s + \mathbf{SE}^s + \mathbf{TE}, \quad s \in \{f, m, c\}, \tag{9}$$

giving $N_f > N_m > N_c$ tokens, all of dimension d.

3.4 Hierarchical transformer encoder

The encoder comprises $L = 4$ identical blocks, each processing $\{\mathbf{Z}^f, \mathbf{Z}^m, \mathbf{Z}^c\}$ through SAFCA, ISIE, and FTCA sub-layers with residual connections and LayerNorm after each sub-layer.

3.4.1 Scale-adaptive fourier cross-attention (SAFCA). Step 1: Fourier Intra-Scale Mixing. A 1D FFT is applied along the token dimension (N_s tokens, each a row vector in $\mathbb{R}^d$), independently per feature channel:

$$\mathbf{F}^s = \mathcal{F}_{N_s}\left(\mathbf{Z}^s\right) \in \mathbb{C}^{N_s \times d}. \tag{10}$$

Learnable complex-valued spectral filters $\mathbf{\Phi}^s \in \mathbb{C}^{N_s \times d}$ modulate each frequency–feature entry element-wise:

$$\tilde{\mathbf{F}}^s = \mathbf{\Phi}^s \odot \mathbf{F}^s. \tag{11}$$

The inverse FFT is applied and the real part retained before any cross-scale operation (so all subsequent Q, K, V are formed from real-valued signals):

$$\tilde{\mathbf{Z}}^s = \text{Re}\left(\mathcal{F}_{N_s}^{-1}\left(\tilde{\mathbf{F}}^s\right)\right) \in \mathbb{R}^{N_s \times d}. \tag{12}$$

The pipeline order is: $\mathbf{Z}^s \xrightarrow{\mathcal{F}} \mathbf{F}^s \xrightarrow{\mathbf{\Phi}^s} \tilde{\mathbf{F}}^s \xrightarrow{\mathcal{F}^{-1}} \tilde{\mathbf{Z}}^s \rightarrow$ cross-scale attention.

Step 2: Cross-Scale Attention. Queries, keys, and values are formed from the Fourier-mixed outputs $\tilde{\mathbf{Z}}^s$. Because N_f, N_m, N_c differ, keys and values from source scale s' are pooled to the query length before attention:

$$\mathbf{K}^{s \leftarrow s'} = \text{Pool}_{N_s}\left(\tilde{\mathbf{Z}}^{s'}\mathbf{W}_K\right), \quad \mathbf{V}^{s \leftarrow s'} = \text{Pool}_{N_s}\left(\tilde{\mathbf{Z}}^{s'}\mathbf{W}_V\right), \tag{13}$$

$$\mathbf{C}^{s\leftarrow s'} = \mathrm{softmax}\left(\frac{\tilde{\mathbf{Z}}^{s}\mathbf{W}_Q\left(\mathbf{K}^{s\leftarrow s'}\right)^{\top}}{\sqrt{d_h}}\right)\mathbf{V}^{s\leftarrow s'}, \tag{14}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ and $d_h = d/H$.

Step 3: Gated Fusion. Following the row-vector convention ($\mathbf{Z}^s \in \mathbb{R}^{N_s\times d}$, tokens as rows), the gate is computed as:

$$\mathbf{G}^s = \sigma\left(\mathbf{Z}^s\mathbf{W}_g^s\right), \quad \mathbf{W}_g^s \in \mathbb{R}^{d\times d}, \tag{15}$$

yielding $\mathbf{G}^s \in \mathbb{R}^{N_s\times d}$. The final SAFCA output is:

$$\mathbf{Z}^s_{\mathrm{SAFCA}} = \mathrm{LayerNorm}\left(\mathbf{G}^s \odot \tilde{\mathbf{Z}}^s + (1-\mathbf{G}^s) \odot \tfrac{1}{2}\textstyle\sum_{s'\neq s}\mathbf{C}^{s\leftarrow s'} + \mathbf{Z}^s\right). \tag{16}$$

All terms are $\mathbb{R}^{N_s\times d}$; element-wise operations are dimensionally consistent.

Fig 2 shows the schematic illustration of the proposed Scale-Adaptive Fourier Cross-Attention module.

3.4.2 Inter-scale information exchange (ISIE). SAFCA injects cross-scale context through attention but leaves each scale normalized independently, allowing representations to drift. ISIE enforces explicit alignment through directed message passing in the order coarse→medium→fine.

This direction is chosen for the following reason: the coarse-scale stream summarizes the longest time context and thus carries the most stable global activity state. Propagating this global context down to finer scales grounds local motion features in the overall activity label before fine-grained temporal modeling begins. The reverse direction (fine→coarse) would risk injecting local noise into the global representation before sufficient context has been established; bidirectional schemes could be explored in future work. Only adjacent scales exchange messages, as non-adjacent scales differ too greatly in temporal resolution for a single linear projection to provide meaningful alignment.

For each adjacent pair (s, s–1) where s–1 is finer:

$$\mathbf{m}^{s\to s-1} = \mathrm{Pool}_{N_{s-1}}\left(\mathbf{Z}^s_{\mathrm{SAFCA}}\right)\mathbf{W}^s_m \in \mathbb{R}^{N_{s-1}\times d}, \tag{17}$$

$$\mathbf{Z}^{s-1}_{\mathrm{ISIE}} = \mathrm{LayerNorm}\left(\mathrm{FFN}\left(\mathbf{Z}^{s-1}_{\mathrm{SAFCA}} + \mathbf{m}^{s\to s-1}\right) + \mathbf{Z}^{s-1}_{\mathrm{SAFCA}}\right). \tag{18}$$

The coarsest stream is passed through unchanged. Fig 3 shows the schematic illustration of the proposed module.

3.4.3 Factorized temporal–channel attention (FTCA). After ISIE, each scale's token sequence $\mathbf{Z}^s_{\mathrm{ISIE}} \in \mathbb{R}^{N_s\times d}$ must model temporal dependencies across the N_s token positions and feature-channel dependencies across the d embedding dimensions. FTCA factorizes these into two sequential operators to avoid the $O(N_s^2 d)$ cost of joint attention.

Temporal Attention. A linear-attention operator is applied along the token dimension using the feature-map approximation $\phi(\mathbf{x}) = \mathrm{elu}(\mathbf{x}) + 1$:

$$\mathrm{LinAttn}(\mathbf{Q},\mathbf{K},\mathbf{V})_i = \frac{\phi(\mathbf{q}_i)^{\top}\left(\sum_j \phi(\mathbf{k}_j)\mathbf{v}_j^{\top}\right)}{\phi(\mathbf{q}_i)^{\top}\sum_j \phi(\mathbf{k}_j)}, \tag{19}$$

where $\mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{d_h}$ and the cumulative sum $\sum_j \phi(\mathbf{k}_j)\mathbf{v}_j^{\top}$ can be maintained in $O(d_h^2)$ memory for causal or $O(N_s d_h^2)$ for full attention, reducing the N_s-dependent cost from $O(N_s^2 d)$ to $O(N_s d)$:

$$\mathbf{Z}^s_T = \mathrm{LayerNorm}\left(\mathrm{LinAttn}_T\left(\mathbf{Z}^s_{\mathrm{ISIE}}\right) + \mathbf{Z}^s_{\mathrm{ISIE}}\right). \tag{20}$$

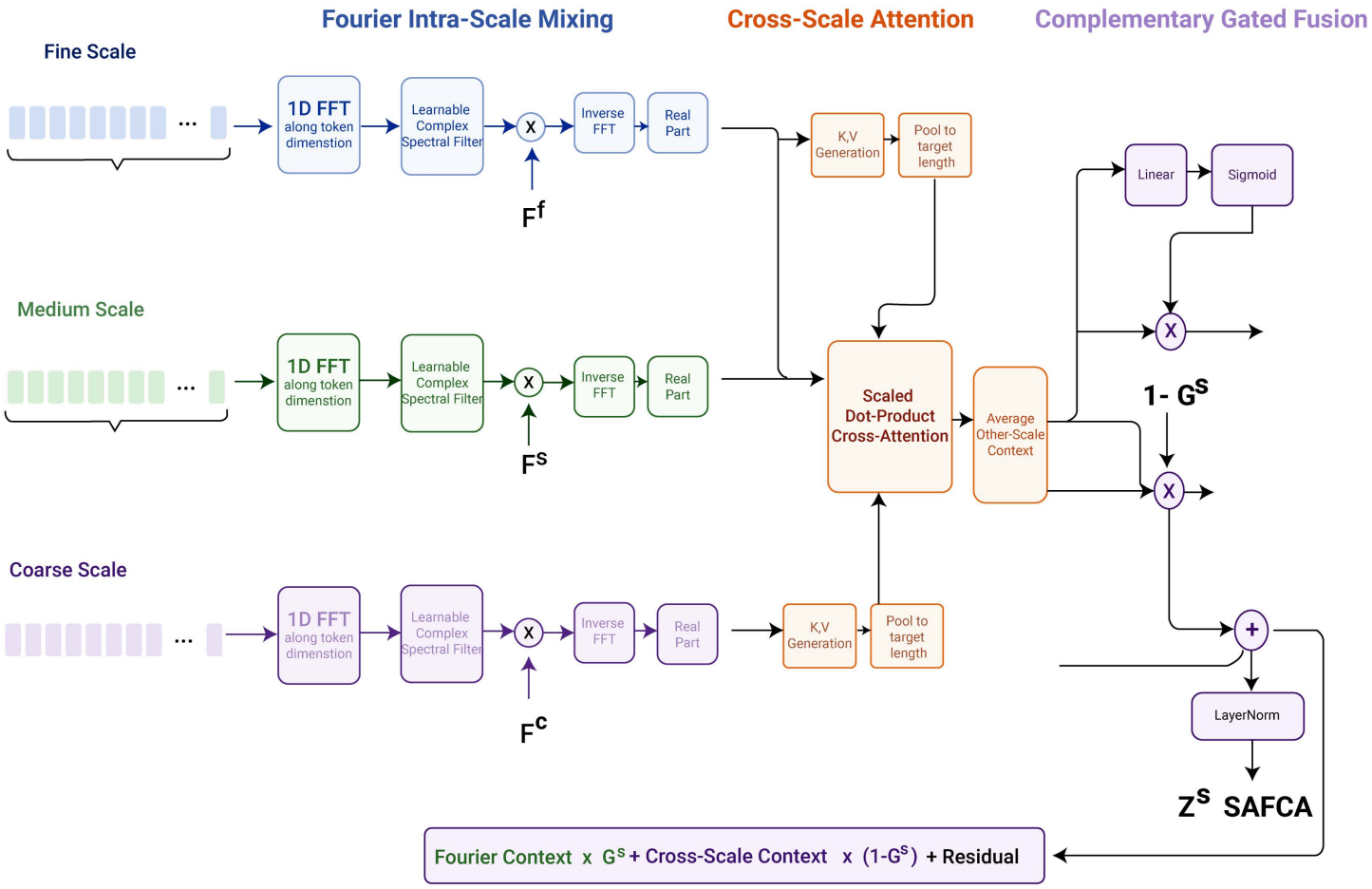


Fig 2. Schematic illustration of the proposed Scale-Adaptive Fourier Cross-Attention (SAFCA) module.

https://doi.org/10.1371/journal.pone.0359774.g002

Channel Attention. To capture dependencies across the d embedding dimensions, we reshape $\mathbf{Z}_T^s$ so that features become the sequence dimension. Concretely, we transpose to treat each of the d feature channels as a "token" of length N_s:

$$\hat{\mathbf{Z}}^s = \left(\mathbf{Z}_T^s\right)^\top \in \mathbb{R}^{d \times N_s}, \tag{21}$$

apply multi-head self-attention with H heads of size N_s/H along the d token dimension:

$$\bar{\mathbf{Z}}^s = \mathrm{MHA}_d\left(\hat{\mathbf{Z}}^s\right) \in \mathbb{R}^{d \times N_s}, \tag{22}$$

then transpose back:

$$\mathbf{Z}_C^s = \mathrm{LayerNorm}\left(\left(\bar{\mathbf{Z}}^s\right)^\top + \mathbf{Z}_T^s\right) \in \mathbb{R}^{N_s \times d}. \tag{23}$$

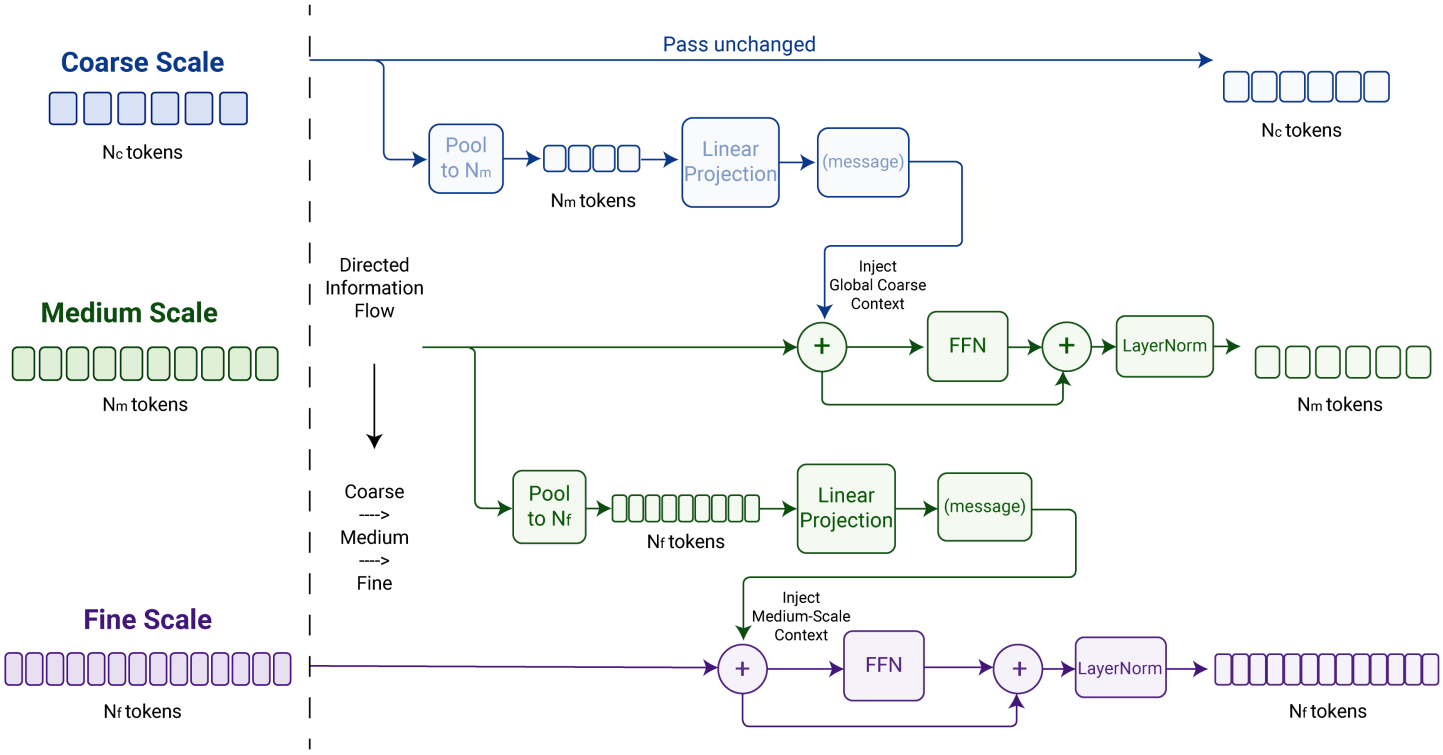


Fig 3. Schematic illustration of the proposed Inter-Scale Information Exchange (ISIE) module with coarse-to-fine message passing.

https://doi.org/10.1371/journal.pone.0359774.g003

FFN Refinement.

$$\mathbf{Z}^s_{\text{out}} = \text{LayerNorm}\left(\text{FFN}\left(\mathbf{Z}^s_C\right) + \mathbf{Z}^s_C\right). \tag{24}$$

Fig 4 shows the schematic illustration of the proposed Factorized Temporal–Channel Attention module.

3.5 Multi-scale aggregation and classification

Intra-Scale Attention Pooling. Each token sequence is collapsed to a fixed vector:

$$\boldsymbol{\alpha}^s = \text{softmax}\left(\frac{\mathbf{Z}^s_{\text{out}}\,\mathbf{w}_\alpha}{\sqrt{d}}\right) \in \mathbb{R}^{N_s}, \quad \mathbf{z}^s = \left(\mathbf{Z}^s_{\text{out}}\right)^\top \boldsymbol{\alpha}^s \in \mathbb{R}^d. \tag{25}$$

Inter-Scale Fusion. Learnable scale weights are computed from the concatenated pooled vectors:

$$\boldsymbol{\beta} = \text{softmax}\left(\mathbf{W}_\beta\,\text{Concat}[\mathbf{z}^f, \mathbf{z}^m, \mathbf{z}^c]\right) \in \mathbb{R}^3, \quad \mathbf{W}_\beta \in \mathbb{R}^{3\times 3d}, \tag{26}$$

$$\mathbf{z} = \beta_f \mathbf{z}^f + \beta_m \mathbf{z}^m + \beta_c \mathbf{z}^c \in \mathbb{R}^d. \tag{27}$$

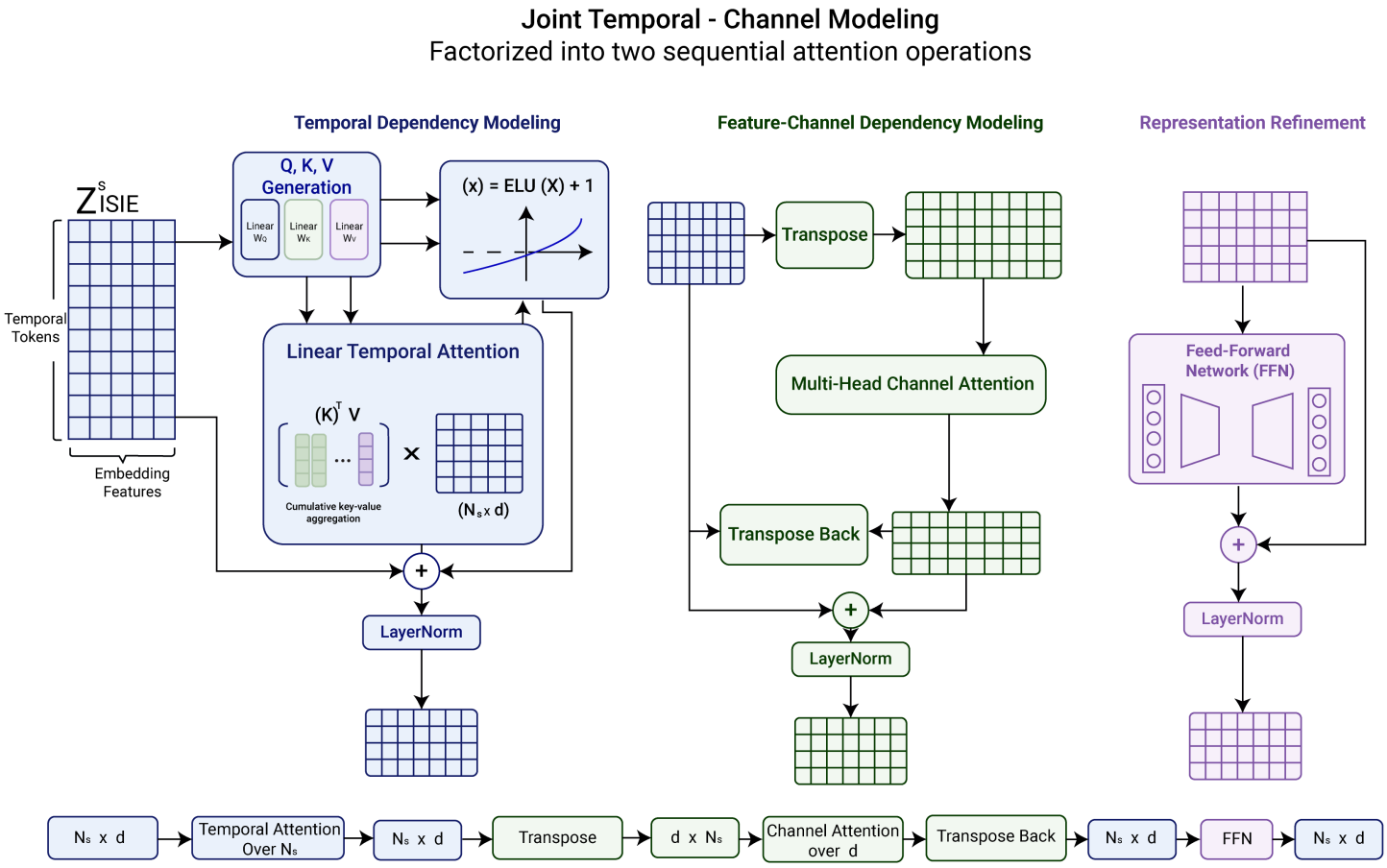


Fig 4. Schematic illustration of the proposed Factorized Temporal–Channel Attention (FTCA) module.

https://doi.org/10.1371/journal.pone.0359774.g004

Classification Head. A 3-layer MLP produces activity logits:

$$\hat{\mathbf{y}} = \mathbf{W}_3\,\mathrm{GELU}\big(\mathbf{W}_2\,\mathrm{GELU}\big(\mathbf{W}_1\,\mathbf{z}\big)\big) \in \mathbb{R}^K. \tag{28}$$

Softmax is applied for multi-class tasks; sigmoid for binary.

3.6 Training strategy

Primary Loss. Label-smoothed cross-entropy with ℓ_2 regularization:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{k=1}^{K} \tilde{y}_k \log \hat{y}_k, \quad \tilde{y}_k = (1-\epsilon)\, y_k + \frac{\epsilon}{K},\ \epsilon=0.1. \tag{29}$$

Per-class frequency weights $w_k \propto 1/n_k$ are applied on imbalanced datasets.

Scale-Utilization Regularizer. A key design assumption is that the three scales contribute complementarily to the final prediction. If the learned scale weights β concentrate on a single scale, the multi-scale design degenerates. We prevent this with an entropy-maximizing regularizer over β:

$$\mathcal{L}_{\mathrm{SU}} = -H(\beta) = \sum_{s\in\{f,m,c\}} \beta_s \log \beta_s. \tag{30}$$

Minimizing $\mathcal{L}_{SU}$ (i.e., maximizing entropy) encourages balanced utilization across scales rather than enforcing strict uniformity. Some datasets may genuinely benefit from one scale more than others; the task loss $\mathcal{L}_{CE}$ retains that flexibility by driving each scale's representation to be discriminative, so the two objectives together encourage *all three scales to be individually informative* rather than collapsed onto a single dominant scale.

Total Objective.

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{SU} + \lambda_2 \|\theta\|_2^2, \tag{31}$$

with λ_1=0.1 and λ_2=10^{-4}.

Optimization. Batch size (64) and 100 epochs with AdamW is used (β_1=0.9, β_2=0.999, lr 10^{-3}) with cosine annealing and warm restarts. In transformer blocks Dropout is applied 0.1 and in the classifier it is applied 0.2.

3.7 Model complexity analysis

Complexity Derivations. Per encoder block and per scale the asymptotic costs are derived as Let $N = \max(N_f, N_m, N_c)$ denote the maximum token count.

- **UCH:** Per-channel descriptor extraction (Eq. 2) computes spectral power bins via FFT, costing $\mathcal{O}(T \log T)$ per channel and $\mathcal{O}(CT \log T)$ across all C channels. The shared MLP scorer (Eq. 3) costs $\mathcal{O}(Cd_\psi)$, independent of T. The soft-weighted group projection and Conv1D projection (Eqs. 4–5) map each of the M groups from C to d/M channels at every timestep, costing $\mathcal{O}(TCd)$ in total. The fusion step (Eq. 6) applies a $d \times d$ linear projection at each timestep, costing $\mathcal{O}(Td^2)$. Since $d \geq C$ across all benchmarks used here ($C \leq 113 \leq d = 128$), the $\mathcal{O}(TCd)$ and $\mathcal{O}(Td^2)$ terms are comparable in order to AMSPE's $\mathcal{O}(Td^2)$ cost below, rather than negligible.
- **AMSPE:** this module employs Conv1D with kernel p_s and stride $\text{str}_s = p_s/2$ with input $T \times d$ and output $N_s \times d$ where $N_s \approx 2T/p_s$. Cost = $N_s \cdot p_s \cdot d \cdot d = (2T/p_s) \cdot p_s \cdot d^2 = O(Td^2)$, to embedding dimension d. This assumes a dense projection and because d is fixed across all the datasets, this reduce to $O(T)$.
- **SAFCA Fourier:** As this module applies a Fourier-based mixing instead of standard self-attention so it apply 1D FFT over N_s tokens, and d channels processing and $O(N_s \log N_s \cdot d)$. Spectral filter application: $O(N_s d)$ to filter them together, the total computational cost becomes $O(Nd \log N)$.
- **ISIE:** This module has mainly two operations Pooling $O(N\,d)$ and linear projection $O(N\,d^2)$. So the total cost, which is linear and cheap, is $O(N\,d^2)$.
- **FTCA Temporal:** This module avoid computing the full matrix so the linear attention costs $O(N\,d)$ (the cumulative sum is $O(d^2)$ memory and $O(N\,d^2)$ estimate, but with $N \gg d$ controlled by $O(N\,d)$ per head). the total complexity is $O(N\,d)$.
- **FTCA Channel:** This module selects across channels instead of across time, so it transposes to $d \times N_s$, Multi-head attention on d tokens of size N_s/H: $O(d^2 \cdot N_s/H \cdot H) = O(d^2 N_s)$. so the final complexity is $O(d^2\,N)$.

The above derivations are summed up in Table 1.

The complete forward pass is explained in Algorithm 1.

Algorithm 1 UniHAR Forward Pass

Require: Sensor window $\mathbf{X} \in \mathbb{R}^{T \times C}$, sampling rate f_s
Ensure: Activity logits $\hat{\mathbf{y}} \in \mathbb{R}^K$
// **Stage 1: Universal Channel Harmonization**
for c=1 to C **do**
  $\psi_c \leftarrow [\mu_c, \sigma_c, \gamma_{1,c}, \gamma_{2,c}, \mathbf{p}_c^\top]^\top$ (Eq. 2)
  $\mathbf{a}_c \leftarrow \text{softmax}(\text{MLP}_g(\psi_c))$ (Eq. 3)

Table 1. Asymptotic computational complexity per encoder block per scale. N: max token count; d: embedding dim; T: time-steps; C: channel count; p_s: patch size.

Module	Dominant operation	Cost
UCH	Feature encoding	$O(CT \log T + TCd + Td^2)$
AMSPE	Conv1D patching	$O(Td^2)$
SAFCA (Fourier mix)	FFT + spectral filter	$O(Nd \log N)$
SAFCA (Cross-attn)	Dot-product attention	$O(N^2 d)$
ISIE	Linear proj. + pooling	$O(Nd^2)$
FTCA (Temporal)	Linear attention	$O(Nd)$
FTCA (Channel)	MHA over d dimension	$O(d^2 N)$

https://doi.org/10.1371/journal.pone.0359774.t001

```
  end for
  A ← [a_1, ..., a_C]^⊤
  for m=1 to M do
    H_m ← Conv1D(X diag(A_:,m), d/M)                    (Eqs. 4–5)
  end for
  X_u ← LayerNorm(W_f Concat[H_1, ..., H_M])            (Eq. 6)
  // Stage 2: Adaptive Multi-Scale Patch Embedding
  {p_s, str_s}_s ← ComputeScales(T, f_s)                (Eq. 7)
  for s ∈ {f, m, c} do
    Z^s ← Conv1D(X_u; p_s, str_s) + PE^s + SE^s + TE    (Eqs. 8–9)
  end for
  // Stage 3: Hierarchical Transformer Encoder
  for ℓ = 1 to L do
    {Z^s} ← SAFCA(Z^f, Z^m, Z^c)                        (Eqs. 10–16)
    {Z^s} ← ISIE(Z^f, Z^m, Z^c)                         (Eqs. 17–18)
    {Z^s} ← FTCA(Z^f, Z^m, Z^c)                         (Eqs. 20–24)
  end for
  // Stage 4: Aggregation and Classification
  for s ∈ {f, m, c} do
    z^s ← AttnPool(Z^s)                                 (Eq. 25)
  end for
  z ← ScaleFuse(z^f, z^m, z^c)                          (Eqs. 26–27)
  ŷ ← MLP(z)                                            (Eq. 28)
  return ŷ
```

3.8 Experimental configuration

Before presenting the experimental results, we summarize the dataset-dependent input configuration and the shared UniHAR architecture and training hyperparameters. UniHAR uses the same backbone configuration across all four datasets. Only the input window length, sampling rate, number of input channels, and number of output classes vary according to the dataset. The Adaptive Multi-Scale Patch Embedding (AMSPE) patch sizes are calculated from the dataset sampling rate using the target durations $\tau_f = 0.15$ s, $\tau_m = 0.35$ s, and $\tau_c = 0.70$ s. Table 2 presents the dataset-dependent settings, while Table 3 summarizes the shared network and optimization settings.

The patch sizes in Table 2 are obtained as

$$\begin{aligned} p_f &= \max\left(4, \lfloor 0.15 f_s \rfloor\right), \\ p_m &= \max\left(8, \lfloor 0.35 f_s \rfloor\right), \\ p_c &= \max\left(16, \lfloor 0.70 f_s \rfloor\right). \end{aligned} \tag{32}$$

Table 2. Dataset-dependent input and AMSPE configurations. Here, f_s denotes the sampling rate, T is the number of samples in an input window, C is the post-preprocessing input-channel count, K is the number of activity classes, and N_f, N_m, and N_c are the numbers of fine-, medium-, and coarse-scale tokens, respectively. The target patch durations are fixed to $(\tau_f, \tau_m, \tau_c) = (0.15, 0.35, 0.70)$ s for all datasets.

Dataset	f_s (Hz)	T	C	K	Window (s)	$p_f/p_m/p_c$	$N_f/N_m/N_c$	Overlap
UCI-HAR	50	128	9	6	2.56	7/17/35	41/14/6	50%
Opportunity	30	90	113	18	3.00	4/10/21	44/17/7	50%
Daphnet	64	192	9	2	3.00	9/22/44	46/16/7	50%
PAMAP2	100	256	18	12	2.56	15/35/70	35/14/6	50%

https://doi.org/10.1371/journal.pone.0359774.t002

The scale-specific stride is defined as

$$\mathrm{str}_s = \max\left(\left\lfloor \frac{p_s}{2} \right\rfloor, 2\right). \tag{33}$$

The resulting number of tokens at scale s is

$$N_s = \left\lfloor \frac{T - p_s}{\mathrm{str}_s} \right\rfloor + 1. \tag{34}$$

All architecture-related settings, including d_{model}, M, L, n_h, and d_{FFN}, are held constant across datasets. The UCH group-assignment scorer uses a two-layer architecture of $16 \to 32 \to 4$, where the four output values represent the soft channel-group assignments. The unified feature dimension is $d_{\text{model}} = 128$, and each of the four UCH groups produces a 32-dimensional feature representation.

The hierarchical encoder contains four identical blocks and uses eight attention heads, resulting in a per-head dimension of $d_h = d_{\text{model}}/n_h = 16$. The feed-forward network expands the 128-dimensional representation to 512 dimensions before projecting it back to 128 dimensions. The classification head subsequently maps the aggregated representation through dimensions $128 \to 256 \to 128 \to K$.

Dataset-specific output layers are used only to accommodate the corresponding number of activity classes K. No changes are made to the UCH, AMSPE, SAFCA, ISIE, or FTCA backbone when moving between datasets.

3.9 Data preprocessing

A standardized pre-treatment was applied to the data from the UCI-HAR, Opportunity, Daphne, and PAMAP2, which included a process of channel selection, the elimination of samples with invalid or incorrect values, and the conversion of data from one channel to another, followed by windowed blocks. All these data were kept at their original sampling rate because AMSPE determines the size of the patches from the physical sampling frequency. To avoid the loss of information, independent data samples were generated before the calculation of pre-treatment statistics, and the normalization parameters were only estimated from the training set.

For UCI-HAR [54], nine accelerometer and gyroscope channels were retained. The 50-Hz signals were represented using 2.56-s windows of 128 samples with 50% overlap. The original six activity classes and official subject-independent split were preserved.

For Opportunity [55], timestamps, annotation fields, and non-sensor metadata were removed, while 113 sensor channels and 18 activity classes were retained. The 30-Hz signals were segmented into 3-s windows of 90 samples with a stride of 45 samples.

Table 3. Shared UniHAR architecture and training configuration used across the evaluated datasets.

Component	Hyperparameter	Setting
Input/UCH	Maximum supported channel count C_{max}	113
	Per-channel descriptor dimension d_{ψ}	16
	UCH group count M	4
	Group-assignment MLP	$16 \rightarrow 32 \rightarrow 4$
	Unified embedding dimension d_{model}	128
	Per-group projection dimension d_{model}/M	32
AMSPE	Number of temporal scales	3
	Target patch durations	0.15/0.35/0.70 s
	Patch stride	$\max(\lfloor p_s/2 \rfloor, 2)$
	Patch-embedding output dimension	128
Hierarchical encoder	Number of encoder blocks L	4
	Number of attention heads n_h	8
	Per-head dimension d_h	16
	Feed-forward dimension d_{FFN}	512
	Feed-forward expansion ratio	$4\times$
	Transformer dropout	0.1
	Normalization	Layer normalization
	Residual connections	Applied after each sub-layer
SAFCA	Fourier-transform dimension	Temporal-token dimension
	Number of temporal scales exchanged	3
	Cross-scale fusion	Learnable gated fusion
ISIE	Message-passing direction	Coarse $\rightarrow$ medium $\rightarrow$ fine
	Scale connections	Adjacent scales only
FTCA	Temporal-attention mechanism	Linear attention with $\phi(x) = \mathrm{ELU}(x) + 1$
	Number of channel-attention heads	8
	Attention implementation dimension	128
	Per-head attention dimension	16
Classifier	Number of fully connected layers	3
	Hidden-layer dimensions	$128 \rightarrow 256 \rightarrow 128 \rightarrow K$
	Activation function	GELU
	Classifier dropout	0.2
Optimization	Optimizer	AdamW
	Initial learning rate	10^{-3}
	Adam coefficients	$\beta_1 = 0.9$, $\beta_2 = 0.999$
	Learning-rate scheduler	Cosine annealing with warm restarts
	Batch size	64
	Maximum number of training epochs	100
	Early-stopping patience	15 epochs
	Label-smoothing coefficient	0.1
	Number of independent runs	5 random seeds
	Class weighting	Inverse-frequency weighting for imbalanced datasets

https://doi.org/10.1371/journal.pone.0359774.t003

For Daphnet [56], nine accelerometer channels from the ankle, thigh, and trunk were used. The 64-Hz recordings were divided into 3-s windows of 192 samples with 50% overlap. The two activity labels were mapped to contiguous class indices.

For PAMAP2 [57], tri-axial accelerometer and gyroscope signals from the hand, chest, and ankle were selected, producing 18 channels. Samples with undefined activity identifier zero were removed. The 100-Hz signals were segmented into 2.56-s windows of 256 samples with a stride of 128 samples.

PAMAP2 missing values were processed independently for each subject, recording, and sensor channel before window segmentation. Invalid and infinite values were first converted to `NaN`. Internal gaps were filled using linear interpolation:

$$\tilde{x}_{t,c} = x_{t_a,c} + \frac{t - t_a}{t_b - t_a}\left(x_{t_b,c} - x_{t_a,c}\right), \tag{35}$$

where t_a and t_b are the nearest valid samples before and after the missing interval. Leading and trailing missing values were replaced by the nearest valid observation using backward and forward filling, respectively. Windows containing remaining non-finite values or channels that were entirely unavailable were discarded.

Finally, each channel was standardized using training-set statistics:

$$z_{t,c} = \frac{x_{t,c} - \mu_c^{\text{train}}}{\sigma_c^{\text{train}} + \varepsilon}, \qquad \varepsilon = 10^{-8}. \tag{36}$$

The same statistics were applied to the validation and test sets. For sample-level annotations, each window received its majority activity label, with the center-sample label used to resolve ties. Windows were constructed independently within each subject and recording and retained their dataset-specific dimensions $T \times C$ for subsequent processing by UCH.

3.10 Data partitioning and model selection

A subject-independent partitioning was used to measure generalization to unseen participants. All the windows corresponding to a given subject were assigned to either the training, validation or test set, so that no subject appeared in more than one part. The partitioning was carried out before normalization and the generation of the individual windows to prevent any leakage of information. The same partitioning was used for all unrelated subjects and ablation experiments. For UCI-HAR, the official partitioning was used: 21 subjects formed the development set, and the nine official test subjects the test set. Four subjects in the development set were set aside for testing, yielding 17 training, 4 testing, and 9 test subjects. For OPP, subjects S1 and S2 were used for training, S3 for validation, and S4 only for testing. For Daphnet, the ten participants were divided into seven training, one validation and two test subjects. For PAMAP, six training, one testing, and two test subjects. The Daphnet and PAMAP subjects were assigned by drawing lots with a known random seed of 42. Thus, the test set consisted of valid windows from subjects kept separate for testing rather than windows randomly selected from those seen during training.

The checkpoint with the lowest validation loss was selected. When validation losses were equal, the checkpoint with the higher validation macro-F1 was selected. When two checkpoints had the same averaged loss in the test set, the one with the highest macro-F1 score was chosen. The selected weights were only reinstated at the end of the training and were then run only once on the test set. The test set was not used for hyperparameter tuning, stopping criteria, checkpoint selection, or normalization of metrics.

4 Results

The comprehensive evalution is performed on four widely used HAR benchmarks: UCI-HAR [54], Opportunity [55], Daphnet Freezing of Gait [56], and PAMAP2 [57]. These datasets consist of smartphone- and IMU-based sensing of human activities, with coarse-grained, varying activity levels. These datasets correspond to challenging scenarios such as pathological motion and severe class imbalance. The subject-independent evaluation is performed on all datasets.

4.1 Baseline models and efficiency evaluation

To ensure a fair comparison, all competing models were reimplemented and evaluated under the same experimental conditions as UniHAR. These models included CNN [24], CNN–LSTM [25], vanilla LSTM (vLSTM), stacked LSTM (sLSTM), and bidirectional LSTM (BiLSTM) [26],Inception-GRU [58], iSPLInception [29], DKInception [59] and Rep-LOM [30]. Their architectures were reproduced according to the corresponding publications, while dataset-specific classification layers were adjusted only to match the number of activity classes.

All models used the same preprocessed sensor windows, subject-independent training, validation, and test partitions, channel normalization, class weights, batch size, optimizer settings, early-stopping criterion, and checkpoint-selection procedure. Each model was trained from scratch, and the checkpoint with the lowest validation loss was evaluated once on the held-out test set. Therefore, the results in Tables 2–Table 5 were obtained using identical data partitions and evaluation protocols rather than being copied from the original publications.

In addition to accuracy and macro-F1 score, computational efficiency was evaluated using the number of trainable parameters, multiply–accumulate operations (MACs), inference latency, and throughput. Parameters and MACs were calculated for one dataset-specific input window. Inference latency was measured with batch size one in evaluation mode after 100 warm-up iterations, followed by 1000 timed forward passes. Throughput was measured with batch size and reported as the number of sensor windows processed per second. Timing excluded data loading and preprocessing and was repeated ten times, with the average result reported. Tables 6 and Table 7 show the computational efficiency metrics.

4.2 Per-dataset results

Our model achieves the highest F1-score and accuracy on every dataset; validation curves, confusion matrices, and ROC curves are shown in Figs 4–15 and Tables 7–11 report the results across all four benchmark datasets.

UCI-HAR. The training and validation curves Fig 5 of UCIHAR dataset show stable convergence and display a good generalization during training.

Our model, as shown in the confusion matrix Fig 6 successfully discriminates between conceptually similar activities.

Class separability is excellently confirmed by the ROC curves Fig 7 across all six activities. UniHAR achieve an accuracy of 96.35% and an F1-score of 96% in the UCI-HAR dataset.

Opportunity. UniHAR achieves an accuracy of 94.45% and an F1-score of 94%. The training and validation curves Fig 8 indicate stable convergence and effective generalization during training.

Table 4. Dataset-specific preprocessing configurations.

Dataset	f_s (Hz)	Duration (s)	T	C	K
UCI-HAR	50	2.56	128	9	6
Opportunity	30	3.00	90	113	18
Daphnet	64	3.00	192	9	2
PAMAP2	100	2.56	256	18	12

https://doi.org/10.1371/journal.pone.0359774.t004

Table 5. Subject-independent data partitioning used for each dataset.

Dataset	Training subjects	Validation subjects	Test subjects
UCI-HAR	17	4	9
Opportunity	2	1	1
Daphnet	7	1	2
PAMAP2	6	1	2

https://doi.org/10.1371/journal.pone.0359774.t005

Table 6. Computational efficiency on Daphnet and PAMAP2. Parameters are reported in millions and MACs per input window.

Method	Daphnet (192 × 9)				PAMAP2 (256 × 18)			
	Params (M)	MACs (M)	Latency (ms)	Throughput (win/s)	Params (M)	MACs (M)	Latency (ms)	Throughput (win/s)
CNN [24]	0.629	21.5	0.94	1024	0.837	35.8	1.22	823
CNN–LSTM [25]	4.026	102.7	4.81	206	4.029	138.3	6.25	161
vLSTM [26]	0.084	16.2	2.02	489	0.099	24.5	2.83	351
sLSTM [26]	0.263	50.3	3.42	293	0.267	67.3	4.60	212
BiLSTM [26]	0.167	31.9	3.24	315	0.196	49.2	4.40	227
iSPLInception [29]	1.327	46.8	1.92	517	1.339	65.2	2.38	416
Inception–GRU [58]	1.606	62.4	2.24	446	1.648	82.7	2.86	350
Rep-LOM [30]	0.354	15.9	1.21	826	0.428	23.8	1.52	658
DKInception [59]	0.397	16.4	1.49	691	0.471	25.2	1.79	558
UniHAR (Ours)	**0.789**	**25.7**	**3.34**	**298**	**0.797**	**24.6**	**3.55**	**281**

https://doi.org/10.1371/journal.pone.0359774.t006

Table 7. Computational efficiency on UCI-HAR and Opportunity. Parameters are reported in millions and MACs per input window. Latency and throughput represent batch-one inference on an NVIDIA A100.

Method	UCI-HAR (128 × 9)				Opportunity (90 × 113)			
	Params (M)	MACs (M)	Latency (ms)	Throughput (win/s)	Params (M)	MACs (M)	Latency (ms)	Throughput (win/s)
CNN [24]	0.424	14.8	0.72	1389	0.311	18.6	0.82	1220
CNN–LSTM [25]	3.502	72.4	3.65	274	3.444	68.1	3.42	292
vLSTM [26]	0.084	10.8	1.42	704	0.120	10.8	1.31	763
sLSTM [26]	0.263	33.7	2.31	433	0.272	24.4	2.08	481
BiLSTM [26]	0.168	21.5	2.18	459	0.239	21.5	1.98	505
iSPLInception [29]	1.327	32.6	1.56	641	1.355	38.9	1.68	595
Inception–GRU [58]	1.612	41.7	1.88	532	1.684	47.9	2.03	493
Rep-LOM [30]	0.356	10.6	0.96	1042	0.521	15.2	1.08	926
DKInception [59]	0.398	11.8	1.13	885	0.611	17.6	1.28	781
UniHAR (Ours)	**0.782**	**22.9**	**2.74**	**365**	**0.805**	**27.8**	**3.12**	**321**

https://doi.org/10.1371/journal.pone.0359774.t007

The confusion matrix Fig 9 shows improved discrimination between closely related activity categories, while the ROC curves Fig 10 further confirm strong class separability across all 18 mid-level classes.

Daphnet. UniHAR attains an accuracy of 96.44% and an F1-score of 96%. The training and validation curves Fig 11 indicate stable convergence and good generalization.

The confusion matrix Fig 12 demonstrates robust detection of the clinically critical “frozen” class under severe class imbalance, while the ROC curves Fig 13 show that all classes achieve an AUC greater than 0.95.

PAMAP2. UniHAR reaches an overall classification accuracy of 99.23% and an overall F1 score of 99%. All of these values are better than those reached by UniHAR’s baseline models. The training curve for UniHAR is shown in Fig 14 to be convergent.

The confusion matrix for UniHAR is shown in Fig 15 as well as being convergent, and shows that UniHAR was able to recognize each of the twelve classes with a good degree of accuracy.

Fig 16 confirms that UniHAR was able to produce good separations among the data points, since it has an area under the receiver operating characteristic curve greater than 0.985 for each class.

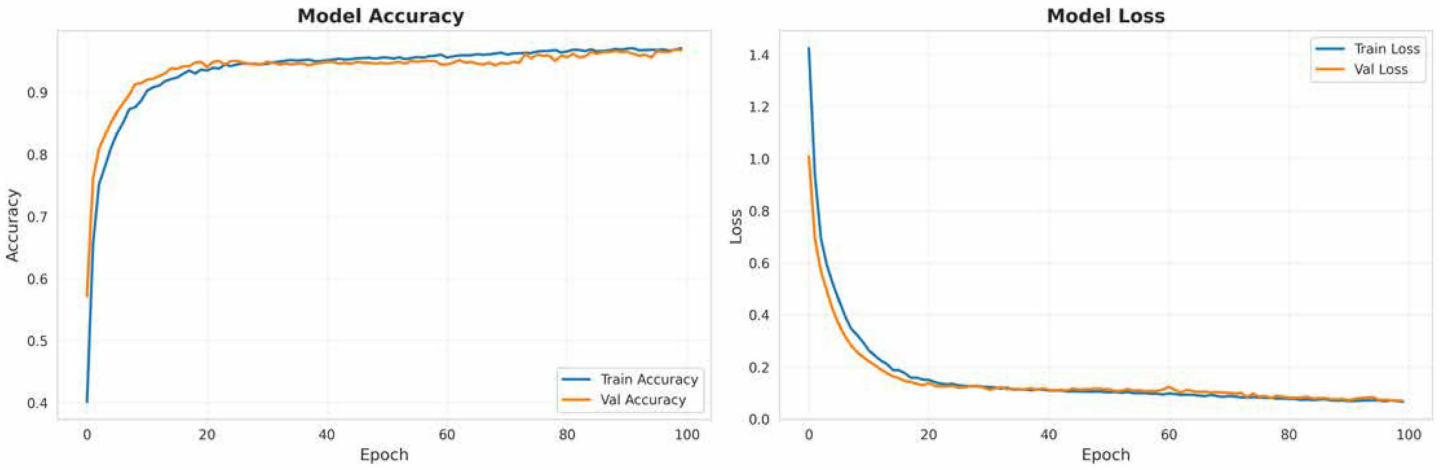


Fig 5. UniHAR results on the UCI-HAR dataset: training and validation curves illustrating the convergence behavior of the proposed UniHAR model.

https://doi.org/10.1371/journal.pone.0359774.g005

UniHAR also demonstrates practical inference efficiency, requiring approximately 5 ms per input window on a desktop CPU.

4.3 Zero-shot cross-dataset generalization results

The zero-shot cross-validation experiments evaluated UniHAR on six shared activities between UCI-HAR and PAMAP-2. No domain-independent samples were used for training, verification, normalization, hyperparameter estimation, and fine-tuning. The results, based on five independent source-domain training runs, are summarized in Table 12.

For UCI-HAR→PAMAP2 transfer, From an accuracy of 73.7%, a macro-F1 score of 69.9%, and a balanced accuracy of 71.1%. the average descriptor grouped by the method of common descriptors gives an advantage over disabling additional channels and averaging their representations. Compared with common channel selection, constant padding and averaging, UCH improved 13.7, 10.6, and 7.8 percentage points, respectively.

For PAMAP2→UCI-HAR transfer, In the classification test, the unique cross-classifier performed remarkably, yielding an accuracy of 79.8% and a macro-F1 score of 77.6%. In this test, the unique cross-classifier outperformed the nearest competing technique, channel-mean aggregation, by a significant margin of 6.5 percentage points in macro-F1. thus indicating a greater tendency toward the $6 \rightarrow 18$ and $18 \rightarrow 6$ channel transitions.

Transfer from PAMAP2 to UCI-HAR yielded an F1 macro score 7.7 percentage points higher than the reverse. One explanation is that PAMAP2 contains measurements from the hand, chest, and ankle and thus provides a wider source-referenced spatial information base than the single-sensor UCI-HAR configuration.

4.3.1 Source-to-target generalization gap. The generalization gap was calculated as

$$\Delta_{\text{gen}} = F1_{\text{source}} - F1_{\text{target}}. \tag{37}$$

Compared to the different channel-handling strategies, the UCH method had the lowest relative generalization gap, with a reduction of 26.2 percentage points for UCI-HAR→PAMAP2 and 21.3 percentage points for PAMAP2→UCI-HAR. These results suggest that the UCH method more effectively transfers the source-based performance when transferring to a different channel configuration. Table 13 show the source-to-target generalization scores

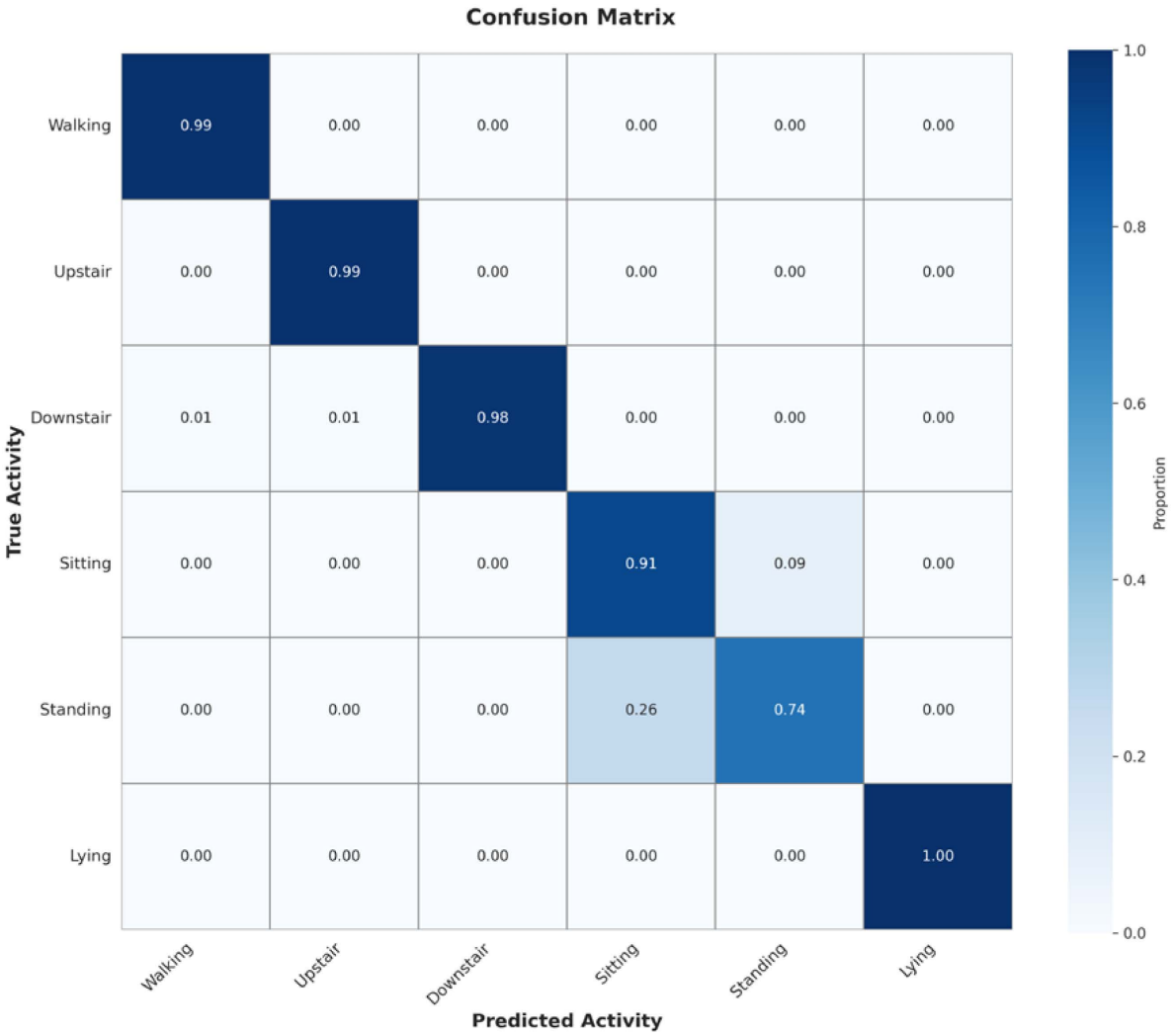


Fig 6. UniHAR results on the UCI-HAR dataset: confusion matrix shows the class-wise classification performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g006

4.3.2 Class-wise transfer performance. Walking was expected to yield the best transfer performance, with F1 scores of 82.9% and 89.4% for UCI-HAR→PAMAP2 and PAMAP2→UCI-HAR, respectively. The main classification errors were expected between standing and sitting, and between climbing and descending stairs, as these two sets of actions had similar temporal characteristics and because they were sensitive to differences in positioning and orientation of the sensors. The standing posture was expected to be the most difficult, yielding F1 scores of 59.4% and 69.5% in the two transfer directions. Table 14 show the class-wise zero shot f1 scores.

4.3.3 Representation alignment. The channel homogenization reduced the MMD from 0.437 to 0.215, while the channel aggregation reduced it from 0.437 to 0.322, which is less efficient than the learned channel grouping. Table 15 shows the discrepancy averaged across the two transfer directions.

4.3.4 Robustness to channel removal. At a channel-removal ratio of 25%, UCH lost 4.2 percentage points, whereas fixed padding and channel-mean aggregation lost 6.7 and 4.9 percentage points, respectively. When half of the sensor channels were unavailable, UCH retained a macro-F1 score of 63.2%, corresponding to a reduction of 10.6 percentage

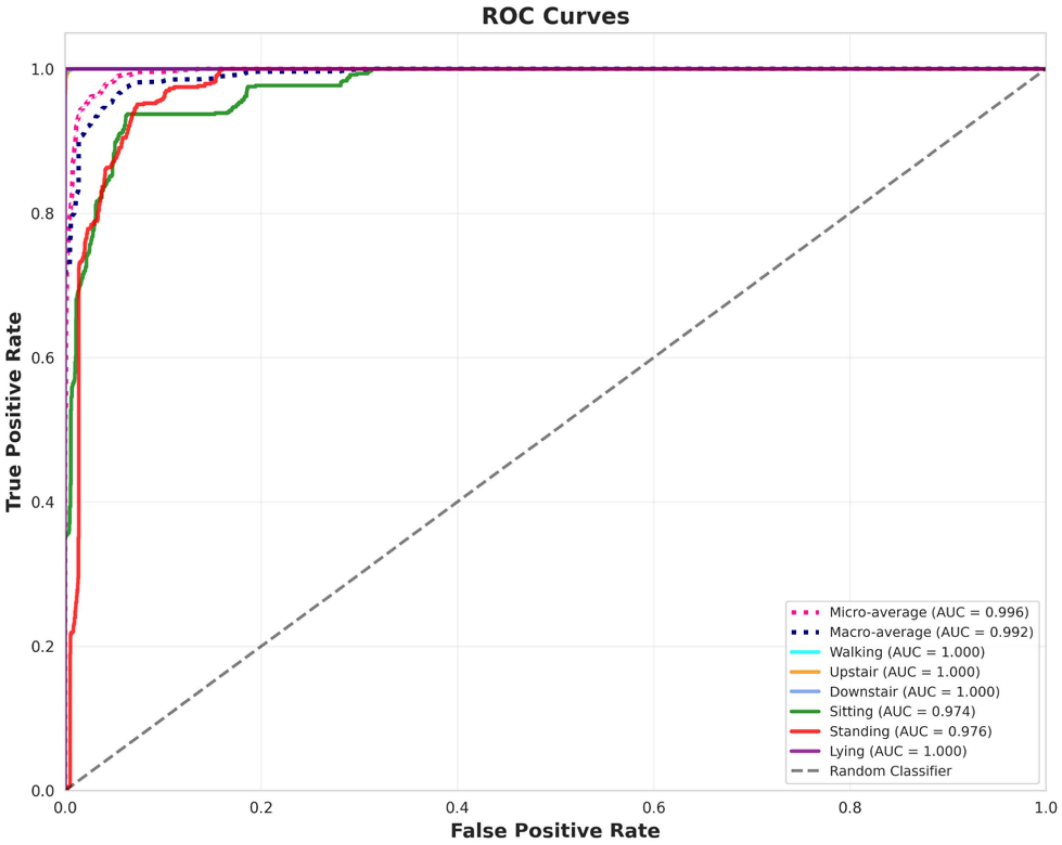


Fig 7. UniHAR results on the UCI-HAR dataset: (ROC) curves show the class-wise discrimination performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g007

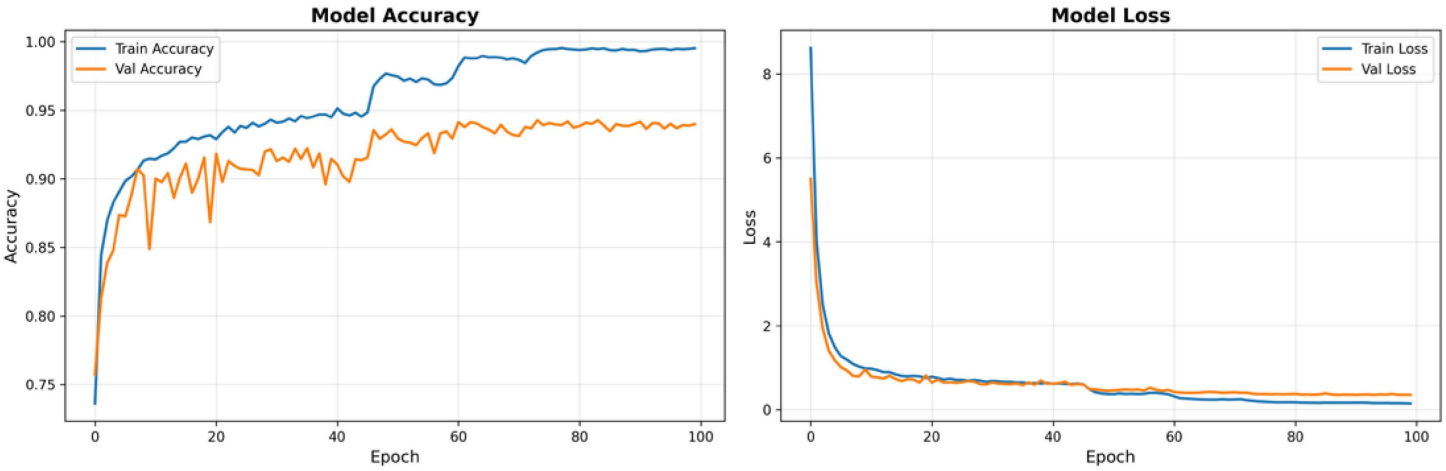


Fig 8. UniHAR results on the Opportunity dataset: training and validation curves illustrating the convergence behavior of the proposed Uni-HAR model.

https://doi.org/10.1371/journal.pone.0359774.g008

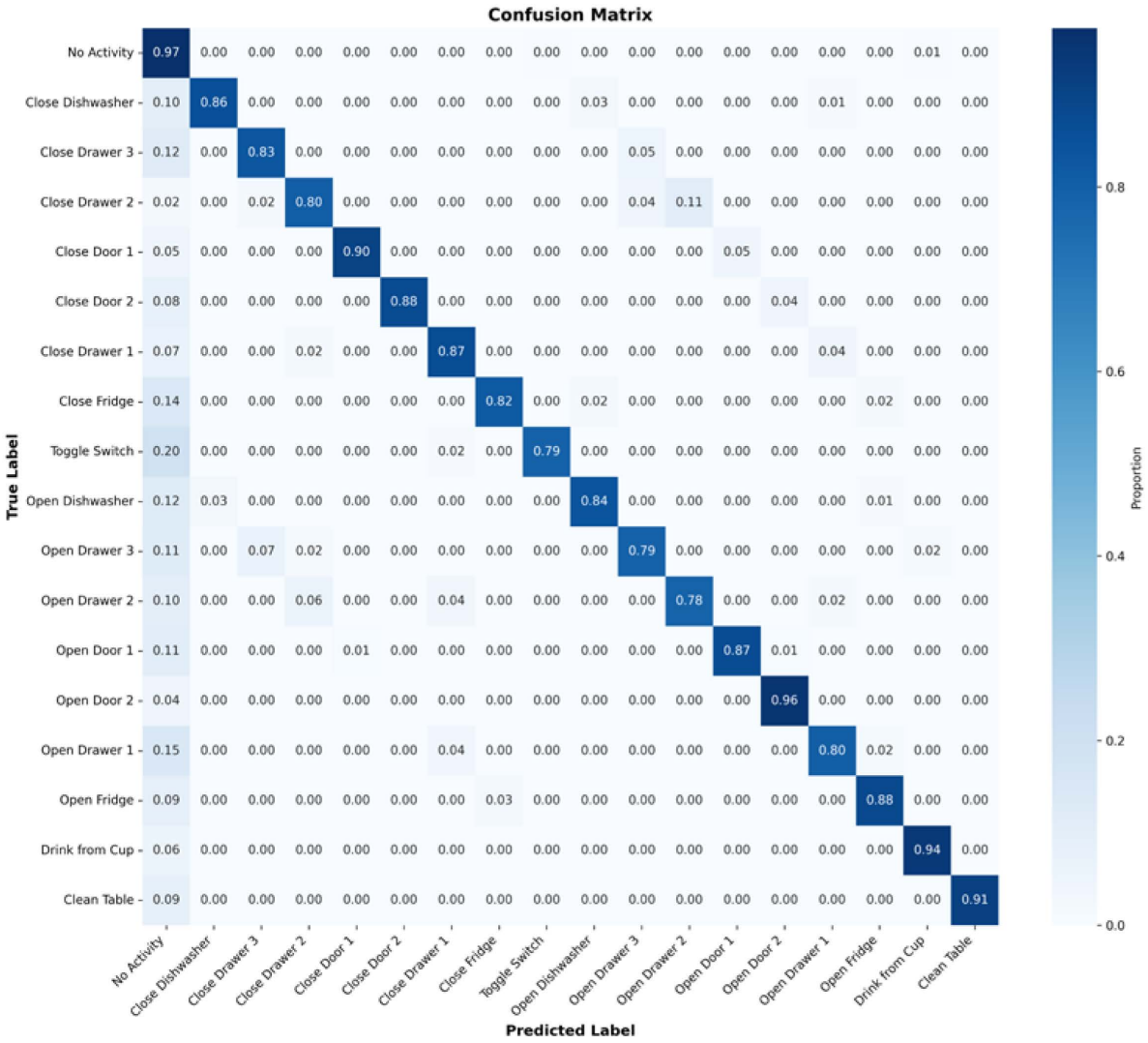


Fig 9. UniHAR results on the Opportunity dataset: confusion matrix shows the class-wise classification performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g009

points from its complete-channel performance. Table 16 shows the target-domain macro-F1, averaged across both transfer directions, under random channel removal.

4.3.5 UCH internal ablation. Uniform channel-mean pooling produced a target-domain macro-F1 score of 66.60%. Incorporating only statistical descriptors increased macro-F1 to 70.05%, while using only spectral descriptors resulted in 69.35%. Random channel grouping achieved 68.40%, showing that independent channel processing alone would not explain the complete UCH improvement.

Full UCH achieved the highest target-domain macro-F1 of 73.75% and the lowest MMD of 0.215. Compared with mean pooling, it introduced approximately 0.04 million additional parameters and 0.34 ms of additional inference latency, while remaining within the sub-5 ms inference range reported for the original UniHAR architecture. Table 17 shows the internal ablation results.

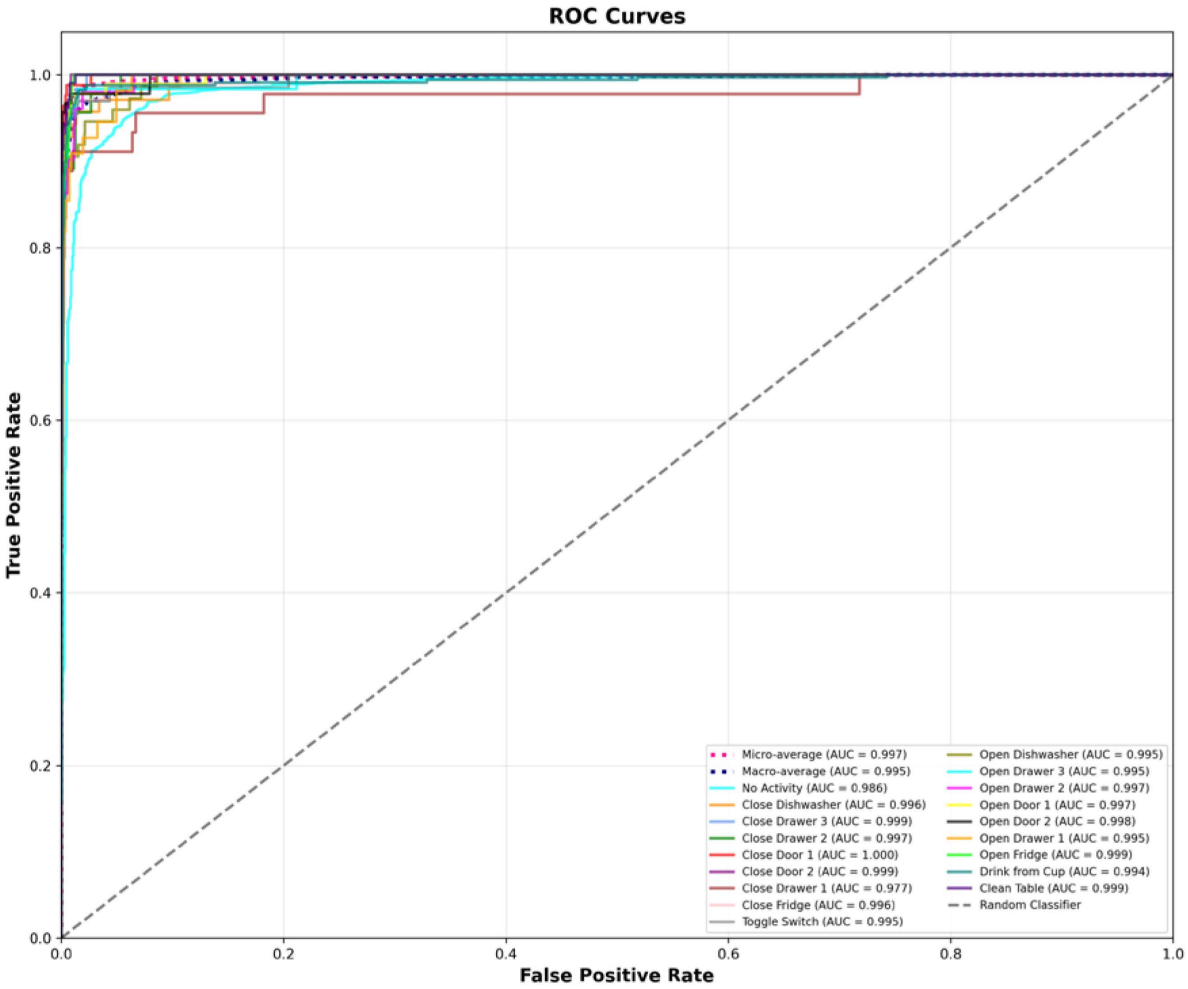


Fig 10. UniHAR results on the Opportunity dataset: (ROC) curves show the class-wise discrimination performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g010

4.3.6 Effect of the number of latent groups. A single latent group effectively behaved as weighted global channel pooling and achieved a macro-F1 score of 66.60%. Increasing the group count to two improved macro-F1 to 71.00%, while four groups achieved the best performance of 73.75%. Increasing the number of groups to eight slightly reduced performance to 72.40%, suggesting that excessive grouping fragments the available channel information. Table 18 shows the effect of the latent group count on target-domain performance.

4.3.7 Statistical significance. Using direction-averaged results from five matched random seeds, full UCH improved target-domain macro-F1 by 7.15 percentage points over channel-mean aggregation. A one-sided exact Wilcoxon signed-rank test and a bootstrap 95% confidence interval were used to assess the statistical significance and uncertainty of this improvement.

5 Discussion

Errors were not random but systematic. On UCI-HAR, 26% of standing samples were misclassified as sitting (Fig 6). This is because of the very low variability of both postures in the IMU data; the coarse-scale AMSPE stream is unable to resolve the ambiguity. On the Opportunity, errors are confined to the fine-grained and low-frequency m-values (e.g.,

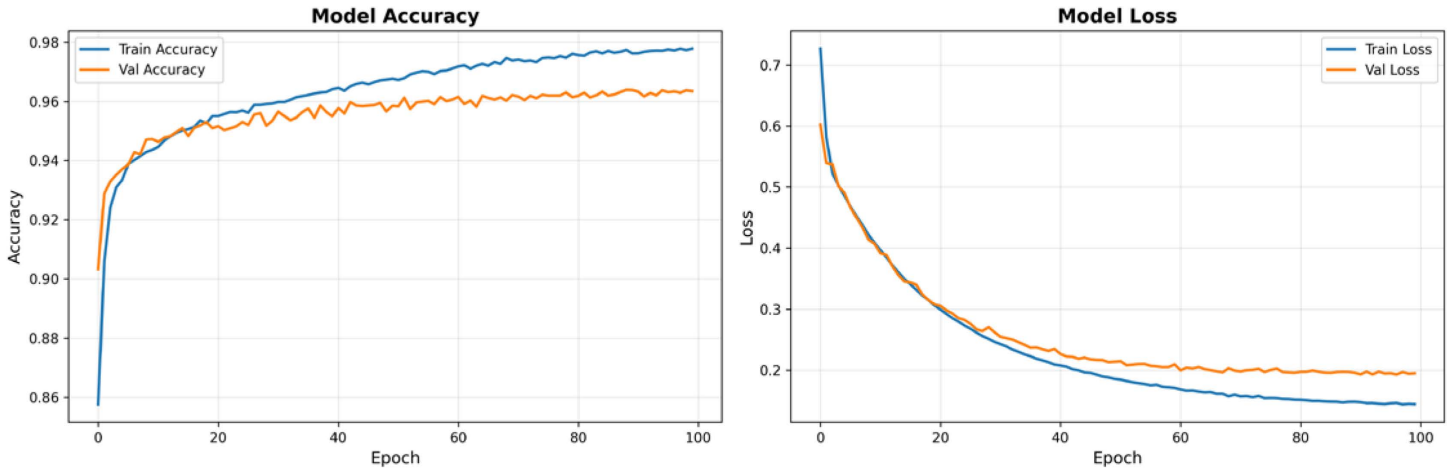


Fig 11. UniHAR results on the Daphnet dataset training and validation curves illustrating the convergence behavior of the proposed UniHAR model.

https://doi.org/10.1371/journal.pone.0359774.g011

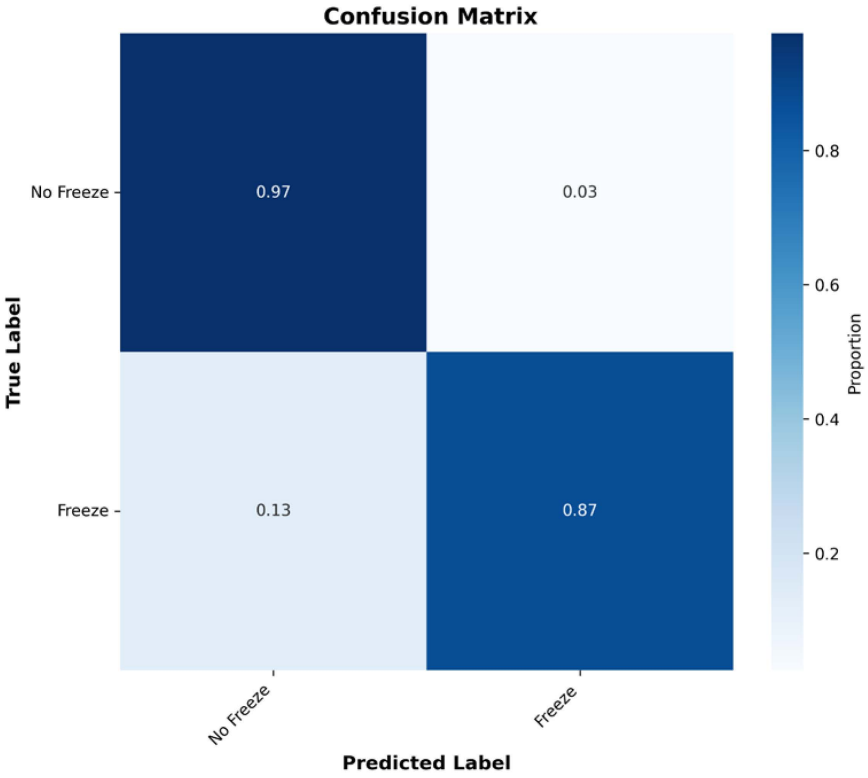


Fig 12. UniHAR results on the Daphnet dataset: confusion matrix shows the class-wise classification performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g012

drawer-open/close, Fig 9). As noted in Table 19, the FTCA approach cost 12.1 points of accuracy on the Opportunity (94.45% → 82.34%) compared to 4.87 and 2.34 points on UCI-HAR and PAMAP2, which is not tested here since all four test datasets are densely recorded. Indicating that channel-dependency modeling, not multi-scale temporal modeling, is

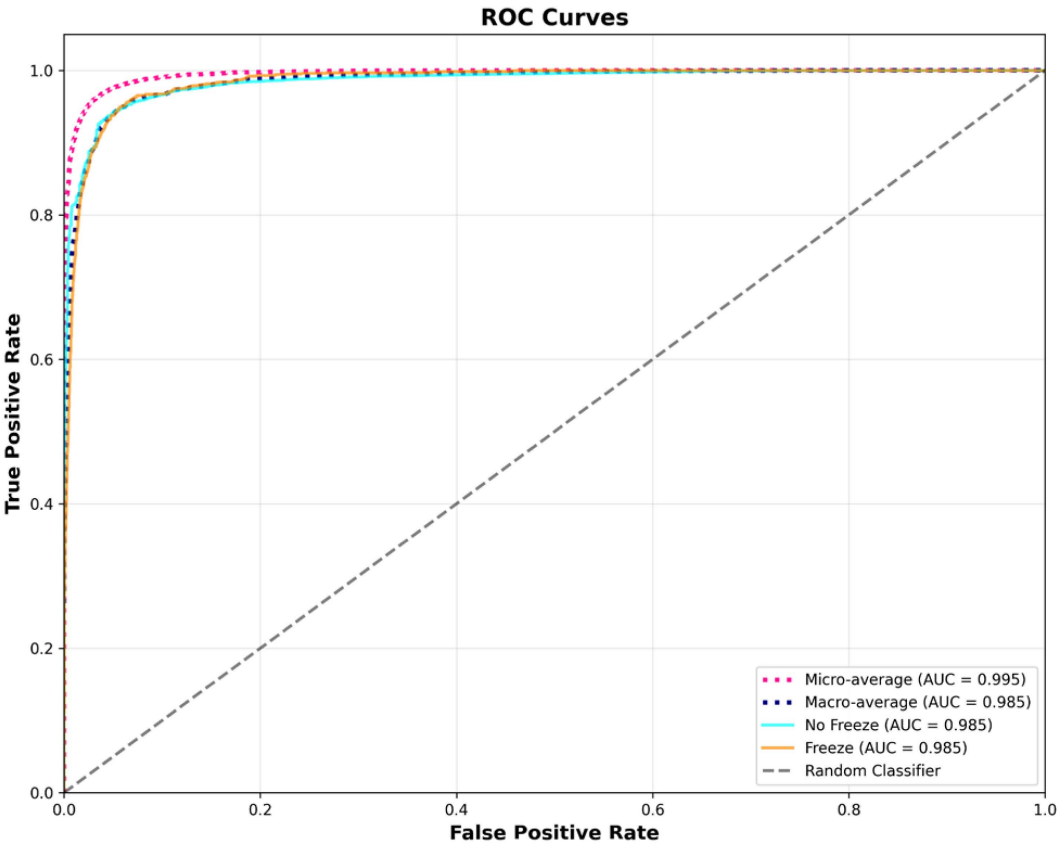


Fig 13. UniHAR results on the Daphnet dataset: (ROC) curves show the class-wise discrimination performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g013

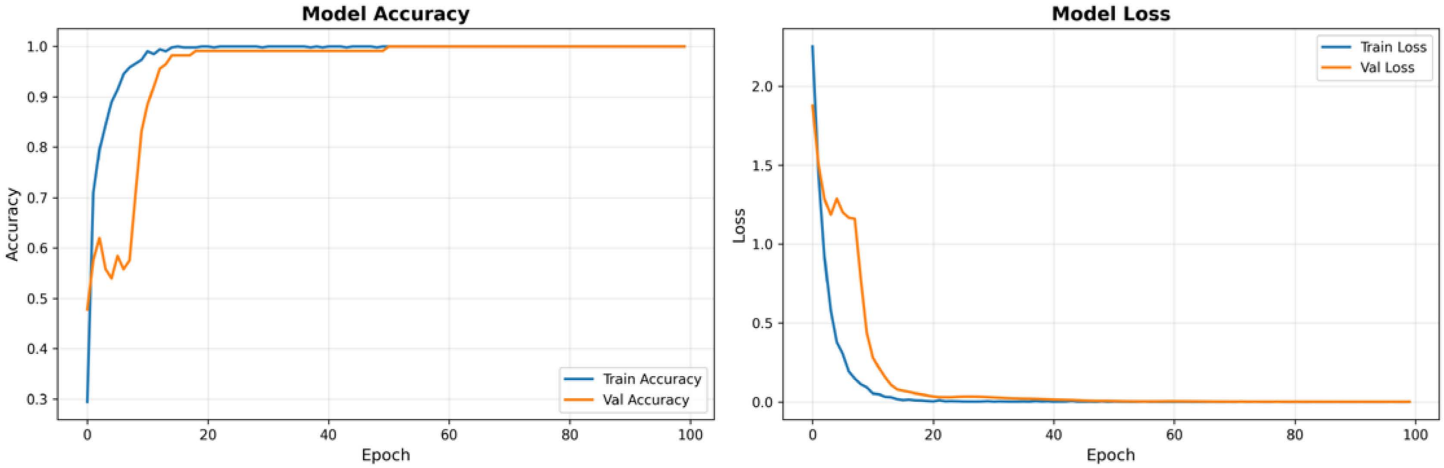


Fig 14. UniHAR results on the PAMAP2 dataset: training and validation curves illustrating the convergence behavior of the proposed UniHAR model.

https://doi.org/10.1371/journal.pone.0359774.g014

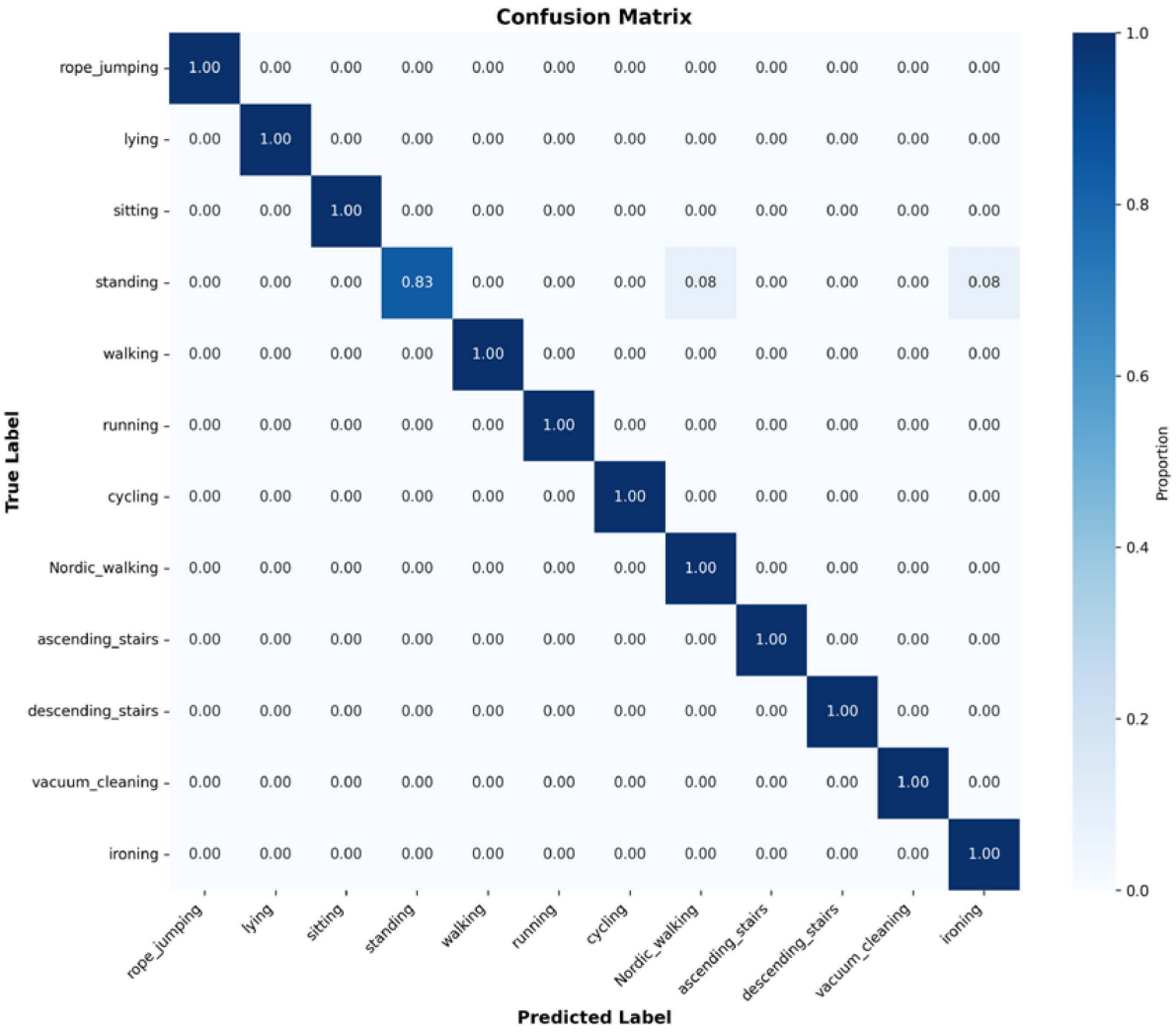


Fig 15. UniHAR results on the PAMAP2 dataset: confusion matrix shows the class-wise classification performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g015

Table 8. Performance comparison on the UCI-HAR dataset.

Method	Acc. (%)	Loss	F1 (%)
CNN	93.66	0.2993	93
CNN-LSTM	92.44	0.3864	92
vLSTM	90.67	0.3344	91
sLSTM	91.62	0.4559	91
BiLSTM	93.51	0.2829	93
Inception-GRU	95.41	0.1627	95
Rep-LOM	95.97	0.1564	95
iSPLInception	95.09	0.1761	95
DKInception	95.70	0.1625	96
UniHAR (Ours)	**96.35**	**0.1520**	**96**

https://doi.org/10.1371/journal.pone.0359774.t008

Table 9. Performance comparison on the Opportunity dataset.

Method	Acc. (%)	Loss	F1 (%)
CNN	78.72	0.8149	77
CNN-LSTM	81.00	0.5979	81
vLSTM	77.66	0.7905	78
sLSTM	81.70	0.6261	81
BiLSTM	79.83	0.6442	80
Inception-GRU	86.03	0.4989	86
Rep-LOM	88.74	0.5026	88
iSPLInception	88.14	0.4790	88
DKInception	87.48	0.5198	87
UniHAR (Ours)	**94.45**	**0.1850**	**94**

https://doi.org/10.1371/journal.pone.0359774.t009

Table 10. Performance comparison on the Daphnet dataset.

Method	Acc. (%)	Loss	F1 (%)
CNN	91.89	0.2847	92
CNN-LSTM	93.11	0.2658	93
vLSTM	93.02	0.2479	93
sLSTM	90.24	0.2821	90
BiLSTM	92.19	0.2485	92
Inception-GRU	94.27	0.2231	94
Rep-LOM	93.33	0.2168	93
iSPLInception	93.52	0.2271	94
DKInception	94.00	0.2286	94
UniHAR (Ours)	**96.44**	**0.2080**	**96**

https://doi.org/10.1371/journal.pone.0359774.t010

Table 11. Performance comparison on the PAMAP2 dataset.

Method	Acc. (%)	Loss	F1 (%)
CNN	86.13	0.6732	86
CNN-LSTM	87.29	0.7041	86
vLSTM	86.21	0.5938	87
sLSTM	86.12	0.6835	86
BiLSTM	86.84	0.5498	87
Inception-GRU	88.91	0.4711	88
Rep-LOM	91.74	0.4227	91
iSPLInception	89.09	0.4322	89
DKInception	89.72	0.4872	90
UniHAR (Ours)	**99.23**	**0.1117**	**99**

https://doi.org/10.1371/journal.pone.0359774.t011

disproportionately load-bearing on high-channel-count datasets ($C = 113$), and that the model's robustness margin there is narrower than headline accuracy suggests.

Architecturally, UCH's channel-count generalization relies on a soft upper bound: channels beyond C_{max}=113 are zero-padded and suppressed only insofar as the group-assignment scorer learns near-zero weights for them (Section 3.2.4), which is an approximation rather than a guarantee and is untested here since all four benchmarks satisfy $C \leq 113$.

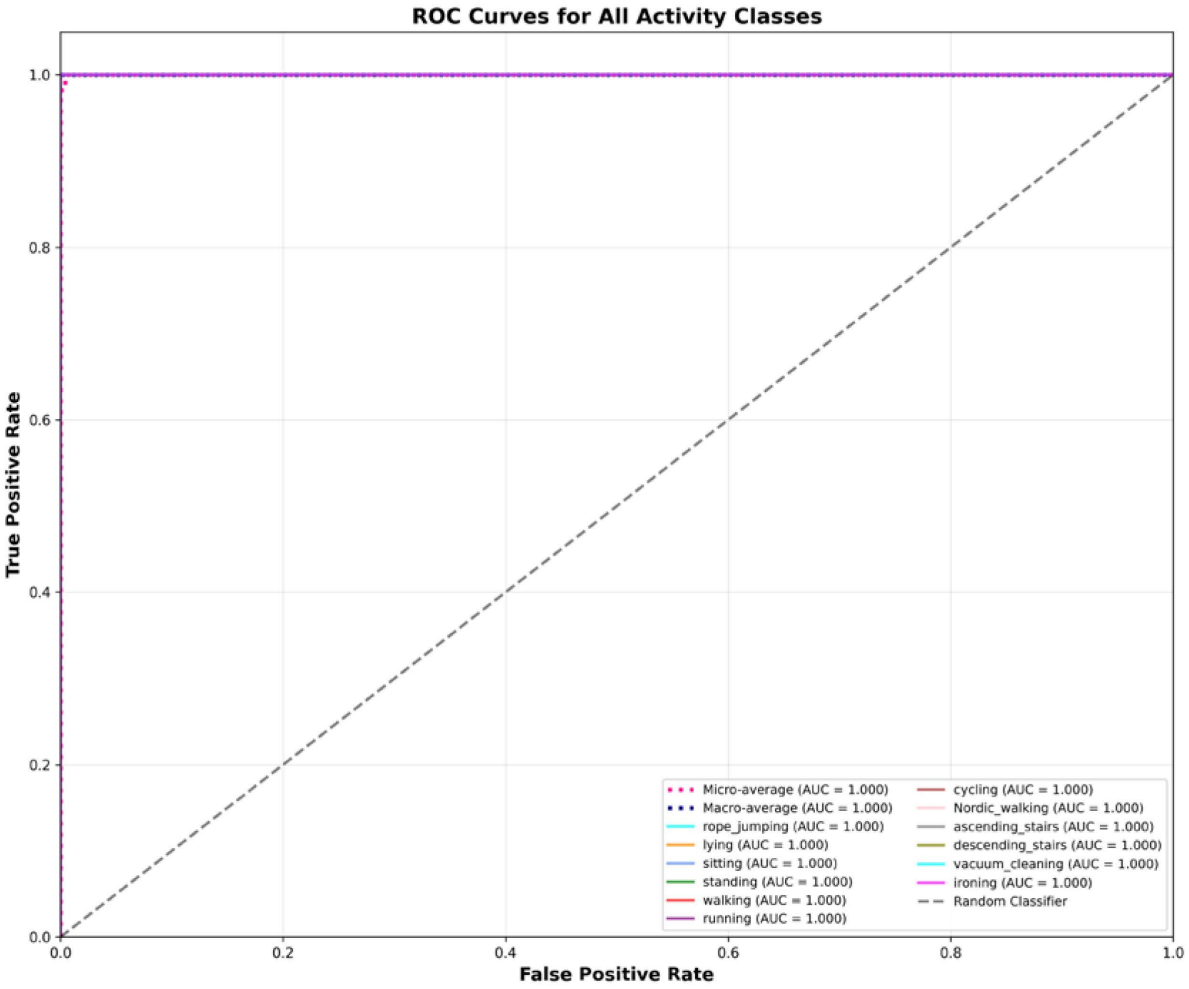


Fig 16. UniHAR results on the PAMAP2 dataset: (ROC) curves show the class-wise discrimination performance of the proposed model.

https://doi.org/10.1371/journal.pone.0359774.g016

Table 12. Zero-shot cross-dataset generalization performance.

Source	Target	Input Method	Accuracy (%)	Macro-F1 (%)	Weighted-F1 (%)	Balanced Acc. (%)
UCI-HAR	PAMAP2	Common-channel selection	59.8	56.2	57.3	55.1
		Fixed padding	63.1	59.3	62.2	60.3
		Channel-mean aggregation	66.0	62.1	64.7	65.1
		UCH	**73.7**	**69.9**	**72.8**	**71.1**
PAMAP2	UCI-HAR	Common-channel selection	65.5	61.3	63.7	62.1
		Fixed padding	69.2	66.8	68.3	67.9
		Channel-mean aggregation	72.8	71.1	74.0	72.7
		UCH	**79.8**	**77.6**	**78.3**	**78.4**

https://doi.org/10.1371/journal.pone.0359774.t012

ISIE's coarse→medium→fine message-passing direction (Section 3.4.2) is fixed a priori rather than learned or data-adaptive; a bidirectional scheme was proposed but not implemented, and this rigidity may partly explain the static-posture errors above, where coarse-scale context should dominate but cannot be reweighted per activity.

Table 13. Source-domain and zero-shot target-domain macro-F1 scores.

Source	Target	Method	Source F1 (%)	Target F1 (%)	Δ_{gen}
UCI-HAR	PAMAP2	Common-channel selection	93.3	56.2	37.1
		Fixed padding	93.9	59.3	34.6
		Channel-mean aggregation	95.7	62.1	33.6
		UCH	**96.1**	**69.9**	**26.2**
PAMAP2	UCI-HAR	Common-channel selection	93.9	61.3	32.6
		Fixed padding	97.1	66.8	30.3
		Channel-mean aggregation	96.9	71.1	25.8
		UCH	**98.9**	**77.6**	**21.3**

https://doi.org/10.1371/journal.pone.0359774.t013

Table 14. Class-wise zero-shot F1 scores for UCH.

Activity	UCI-HAR→PAMAP2	PAMAP2→UCI-HAR
Walking	82.9	89.4
Walking upstairs	68.6	75.7
Walking downstairs	66.2	73.6
Sitting	62.3	71.3
Standing	59.4	69.5
Lying	80.0	86.1
Macro average	**69.9**	**77.6**

https://doi.org/10.1371/journal.pone.0359774.t014

Table 15. Source–target representation discrepancy averaged across the two transfer directions. Lower MMD values indicate better alignment.

Representation method	MMD
Before harmonization	0.437
Channel-mean aggregation	0.322
UCH	**0.215**

https://doi.org/10.1371/journal.pone.0359774.t015

Table 16. Target-domain macro-F1 averaged across both transfer directions under random channel removal.

Removed channels	Fixed padding	Mean aggregation	UCH
0%	63.1	66.6	**73.8**
10%	60.8	65.1	**71.8**
25%	56.4	61.7	**69.6**
50%	47.9	53.8	**63.2**
75%	35.6	41.7	**51.4**

https://doi.org/10.1371/journal.pone.0359774.t016

On efficiency, UniHAR's inference latency (2.74 ms on UCI-HAR, batch size one, Table 7) exceeds eight of nine baselines, including CNN (0.72 ms) and DKInception (1.13 ms), and its parameter count (0.782M) exceeds several lightweight baselines by 2–9×; the efficiency claim is therefore a favorable accuracy–cost trade-off rather than outright superiority.

On generalization, the zero-shot cross-dataset experiments (Section 4.3) show UCH narrows but does not close the source-to-target gap: target-domain macro-F1 remains 22–26 points below source-domain performance (Table 13), and this evaluation covers only a six-class harmonized subset shared between UCI-HAR and PAMAP2 rather than either

Table 17. Internal ablation results averaged across the two cross-dataset transfer directions.

Variant	Source F1 (%)	Target F1 (%)	Balanced Acc. (%)	MMD	Parameters (M)	Latency (ms)
Channel-mean pooling	96.30	66.60	68.90	0.322	0.80	3.58
UCH-statistics	96.80	70.05	70.75	0.257	0.83	3.76
UCH-spectrum	96.55	69.35	70.05	0.274	0.83	3.81
UCH-random groups	96.30	68.40	69.10	0.298	0.82	3.73
Full UCH	**97.50**	**73.75**	**74.75**	**0.215**	0.84	3.92

https://doi.org/10.1371/journal.pone.0359774.t017

Table 18. Effect of the latent group count on target-domain performance.

M	Macro-F1 (%)	MMD	Parameters (M)
1	66.60	0.322	0.80
2	71.00	0.252	0.82
4	**73.75**	**0.215**	0.84
8	72.40	0.226	0.88

https://doi.org/10.1371/journal.pone.0359774.t018

Table 19. Accuracy (%) of component-wise ablation variants across datasets.

Configuration	UCI-HAR	Opp.	Daph.	PAMAP2
Full UniHAR	**96.35**	**94.45**	**96.44**	**99.23**
w/o UCH	93.89	87.62	92.31	95.47
w/o AMSPE	92.24	85.89	91.78	97.71
w/o SAFCA	92.12	91.58	90.89	97.85
w/o ISIE	96.21	92.87	96.38	98.18
w/o FTCA	91.48	82.34	92.12	96.89
w/o Fourier mixing	94.17	93.24	91.56	98.94

https://doi.org/10.1371/journal.pone.0359774.t019

dataset's full taxonomy. All four evaluation datasets are wearable-IMU-based; extension to other HAR modalities (WiFi CSI, mmWave radar, vision-based skeletons) is untested, and channel harmonization across modalities with different physical semantics may require descriptor design beyond the statistical/spectral features used in UCH (Section 3.2.1).

The results of these experiments suggest several concrete extensions. Adaptive or learned channel-count bounds could replace the static zero-padding. Similarly, a bi-directional AMSPE scheme could refine the static-posture error.

5.1 Ablation study

To determine how much the performance of the UniHAR system depends on the various constituents of the system, a series of systematic ablation experiments were carried out on the system, using the four available data. These studies showed a different impact on the different combinations of sensors, each of which was used in a different way for each data set. The ablation results of the individual components of the UniHAR system, tested against all data sets, are shown in Table 19. This table demonstrates how the removal of each individual module affects the overall performance of the UniHAR system.

6 Conclusion

In this study, a hierarchical multi-scale Transformer, UniHAR, is proposed for sensor-based human activity recognition. The model is suitable for edge deployment to address the heterogeneous sensor configurations and multi-scale temporal

modeling. Integration of multiple modules, including Adaptive Multi-Scale Patch Embedding, Universal Channel Harmonization, Factorized Temporal–Channel Attention and Fourier-based cross-scale attention. On four benchmark datasets our model achieves state-of-the-art performance with only sub-5ms inference latency on a desktop CPU and a lightweight architecture. Complementary contribution of each module is proved by an ablation study. In future work, our focus will be on imbalance-aware learning, adaptive module selection, and cross-dataset generalization to further improve the health care assistive applications.

Author contributions

Conceptualization: Memoona.

Data curation: Memoona.

Formal analysis: Memoona, Muhammad Haseeb Hashir.

Funding acquisition: Sung Won Kim.

Methodology: Memoona.

Project administration: Muhammad Haseeb Hashir.

Resources: Muhammad Haseeb Hashir.

Software: Memoona.

Supervision: Sung Won Kim.

Validation: Memoona.

Visualization: Memoona.

Writing – original draft: Memoona.

Writing – review & editing: Memoona.

References

1. Beddiar DR, Nini B, Sabokrou M, Hadid A. Vision-based human activity recognition: a survey. Multimed Tools Appl. 2020;79(41–42):30509–55. https://doi.org/10.1007/s11042-020-09004-3
2. Taylor W, Shah SA, Dashtipour K, Zahid A, Abbasi QH, Imran MA. An Intelligent Non-Invasive Real-Time Human Activity Recognition System for Next-Generation Healthcare. Sensors (Basel). 2020;20(9):2653. https://doi.org/10.3390/s20092653 PMID: 32384716
3. Serpush F, Menhaj MB, Masoumi B, Karasfi B. Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System. Comput Intell Neurosci. 2022;2022:1391906. https://doi.org/10.1155/2022/1391906 PMID: 35251142
4. Mukherjee A, Bose A, Chaudhuri DP, Kumar A, Chatterjee A, Ray SK, et al. Edge-based Human Activity Recognition System for Smart Healthcare. J Inst Eng India Ser B. 2021;103(3):809–15. https://doi.org/10.1007/s40031-021-00663-w
5. Hsu Y-L, Yang S-C, Chang H-C, Lai H-C. Human Daily and Sport Activity Recognition Using a Wearable Inertial Sensor Network. IEEE Access. 2018;6:31715–28. https://doi.org/10.1109/access.2018.2839766
6. Zhuang Z, Xue Y. Sport-Related Human Activity Detection and Recognition Using a Smartwatch. Sensors (Basel). 2019;19(22):5001. https://doi.org/10.3390/s19225001 PMID: 31744127
7. Koskimäki H, Siirtola P, Röning J. MyoGym: Introducing an open gym dataset for activity recognition collected using the Myo armband. In: Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2017 ACM International Symposium on Wearable Computers. 2017. p. 537–46.
8. Koskimäki H, Siirtola P. Recognizing unseen gym activities from streaming data—accelerometer vs. electromyogram. In: Distributed Computing and Artificial Intelligence, 13th International Conference. Springer; 2016. p. 195–202.
9. Vazan M, Masoumi FS, Ou R, Rawassizadeh R. Augmenting vision-based human pose estimation with rotation matrix. ArXiv preprint arXiv:2310.06068. 2023.
10. Fu B, Kirchbuchner F, Kuijper A. Performing Realistic Workout Activity Recognition on Consumer Smartphones. Technologies. 2020;8(4):65. https://doi.org/10.3390/technologies8040065

11. Soro A, Brunner G, Tanner S, Wattenhofer R. Recognition and Repetition Counting for ComplexPhysical Exercises with Deep Learning. Sensors (Basel). 2019;19(3):714. https://doi.org/10.3390/s19030714 PMID: 30744158
12. Morris D, Saponas TS, Guillory A, Kelner I. RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2014. p. 3225–34.
13. Sunil A, Sheth MH, Shreyas E, et al. Usual and unusual human activity recognition in video using deep learning and artificial intelligence for security applications. In: 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE; 2021. p. 1–6.
14. Malibari AA, Alzahrani JS, Qahmash A, Maray M, Alghamdi M, Alshahrani R, et al. Quantum Water Strider Algorithm with Hybrid-Deep-Learning-Based Activity Recognition for Human–Computer Interaction. Appl Sci. 2022;12(14):6848. https://doi.org/10.3390/app12146848
15. Minh Dang L, Min K, Wang H, Jalil Piran Md, Hee Lee C, Moon H. Sensor-based and vision-based human activity recognition: A comprehensive survey. Pattern Recognit. 2020;108:107561. https://doi.org/10.1016/j.patcog.2020.107561
16. Almusawi A, Ali AH. Efficient human activity recognition system using long short-term memory. In: International Conference of Reliable Information and Communication Technology. Springer; 2021. p. 73–83.
17. Islam MM, Nooruddin S, Karray F, Muhammad G. Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges, and future prospects. Comput Biol Med. 2022;149:106060. https://doi.org/10.1016/j.compbiomed.2022.106060 PMID: 36084382
18. Wang X, Shang J. Human Activity Recognition Based on Two-Channel Residual–GRU–ECA Module with Two Types of Sensors. Electronics. 2023;12(7):1622. https://doi.org/10.3390/electronics12071622
19. Zebin T, Scully PJ, Ozanyan KB. Human activity recognition with inertial sensors using a deep learning approach. In: 2016 IEEE Sensors. IEEE; 2016. p. 1–3.
20. Lattanzi E, Contoli C, Freschi V. Do we need early exit networks in human activity recognition? Eng Appl Artif Intell. 2023;121:106035. https://doi.org/10.1016/j.engappai.2023.106035
21. Mishra R, Gupta H. Transforming large-size to lightweight deep neural networks for IoT applications. ACM Comput Surveys. 2023;55(11):1–35.
22. Han Y, Huang G, Song S, Yang L, Wang H, Wang Y. Dynamic Neural Networks: A Survey. IEEE Trans Pattern Anal Mach Intell. 2022;44(11):7436–56. https://doi.org/10.1109/TPAMI.2021.3117837 PMID: 34613907
23. Bulling A, Blanke U, Schiele B. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput Surv. 2014;46(3):1–33. https://doi.org/10.1145/2499621
24. Yang J, Nguyen MN, San PP, Li X, Krishnaswamy S. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In: Proc. 24th Int. Joint Conf. Artificial Intelligence (IJCAI). 2015. p. 3995–4001.
25. Ordóñez FJ, Roggen D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors (Basel). 2016;16(1):115. https://doi.org/10.3390/s16010115 PMID: 26797612
26. Hammerla NY, Halloran S, Plötz T. Deep, convolutional, and recurrent models for human activity recognition using wearable sensors. In: Proc. 25th Int. Joint Conf. Artificial Intelligence (IJCAI). 2016. p. 1533–40.
27. Guan Y, Plötz T. Ensembles of Deep LSTM Learners for Activity Recognition using Wearables. Proc ACM Interact Mob Wearable Ubiquitous Technol. 2017;1(2):1–28. https://doi.org/10.1145/3090076
28. Ismail FH, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, et al. InceptionTime: Finding AlexNet for Time Series Classification. Data Min Knowl Discov. 2020;34(6):1936–62. https://doi.org/10.1007/s10618-020-00710-y
29. Ronald M, Poulose A, Han DS. iSPLInception: An Inception-ResNet Deep Learning Architecture for Human Activity Recognition. IEEE Access. 2021;9:68985–9001. https://doi.org/10.1109/access.2021.3078184
30. Liu J, Zhang L, Liu X, Yao M, Wang L, Wu H, et al. Rep-LOM: Multibranch Linear Operation Module With DFC Attention for Sensor-Based Human Activity Recognition. IEEE Sensors J. 2025;25(12):22291–306. https://doi.org/10.1109/jsen.2025.3565646
31. Fu Z, Zhang L, Huang W, Cheng D, Wu H, Song A. Learning Sensor Sample-Reweighting for Dynamic Early-Exit Activity Recognition Via Meta Learning. IEEE J Biomed Health Inform. 2025;29(6):4081–94. https://doi.org/10.1109/JBHI.2024.3524252 PMID: 40030777
32. Liu J, Zhang L, Liu X, Chen G, Fu Z, Huang W, et al. Rep-MMB: Bridging Mobile CNN and Transformer for Sensor-Based Human Activity Recognition. IEEE Internet Things J. 2026;13(13):28866–82. https://doi.org/10.1109/jiot.2026.3685081
33. Lian C, Li W, Zhou D, Kang Y, Sun T, Lyu X, et al. TDI-TFFNet: Infusing time dependent images and two-stream feature fusion network for gymnastic activity recognition. Pattern Recognit. 2026;172:112726. https://doi.org/10.1016/j.patcog.2025.112726
34. Zhao Y, Shao J, Lin X, Sun T, Li J, Lian C, et al. CIR-DFENet: Incorporating cross-modal image representation and dual-stream feature enhanced network for activity recognition. Expert Syst Appl. 2025;266:125912. https://doi.org/10.1016/j.eswa.2024.125912
35. Lian C, Li WJ, Kang Y, Li W, Zhou D, Zhan Z, et al. Enhanced human lower-limb motion recognition using flexible sensor array and relative position image. Pattern Recognit. 2026;171:112142. https://doi.org/10.1016/j.patcog.2025.112142
36. Lian C, Zhao Y, Sun T, Shao J, Liu Y, Fu C, et al. Incorporating image representation and texture feature for sensor-based gymnastics activity recognition. Knowl-Based Syst. 2025;311:113076. https://doi.org/10.1016/j.knosys.2025.113076

37. Zhao Y, Meng X, Ren X, Lian C, Sun J, Sha X. Quantitative evaluation of gymnastics based on multiple MEMS sensors. IEEE Sensors J. 2021;21(21):24531–9. https://doi.org/10.1109/JSEN.2021.3114758
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN. Attention Is All You Need. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 30. 2017.
39. Zerveas G, Jayaraman S, Patel D, Bhamidipaty A, Eickhoff C. A Transformer-based Framework for Multivariate Time Series Representation Learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD). 2021. p. 2114–24.
40. Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. AAAI. 2021;35(12):11106–15. https://doi.org/10.1609/aaai.v35i12.17325
41. Wu H, Xu J, Wang J, Long M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34. 2021. p. 22419–30.
42. Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A Time Series is Worth 64 Words: Long-Term Forecasting with Transformers. In: Proc. 11th Int. Conf. Learning Representations (ICLR). 2023.
43. Xu H, Zhou P, Tan R, Li M, Shen G. LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications. In: Proc. 19th ACM Conf. Embedded Networked Sensor Systems (SenSys). 2021. p. 220–33.
44. Tang Y, Tran H, Keupp M, Keupp M, Sax E, Schubert F. Triple Cross-Domain Attention on Human Activity Recognition Using Wearable Sensors. In: Proc. 18th IEEE Int. Conf. Ubiquitous Intelligence and Computing (UIC). 2021. https://doi.org/10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Martec53992.2021.00065
45. Fu Z, Cheng D, Zhang L, Huang W, Chen Z, Wu H. DeepSenseMoE: Harnessing Power of Time Series Foundation Models for Few-Shot Human Activity Recognition. AAAI. 2026;40(1):292–9. https://doi.org/10.1609/aaai.v40i1.36990
46. Liu X, Cheng D, Fu Z, Zhang L, Wu H, Song A. Sensor-Prompt Tuning: Aligning Time Series Foundational Models With Motion Sensors for Few-Shot Activity Recognition. IEEE Trans Mobile Comput. 2026;25(7):11227–43. https://doi.org/10.1109/tmc.2026.3664424
47. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021. p. 9992–10002.
48. Fan H, Xiong B, Mangalam K, Li Y, Yan Z, Malik J, et al. Multiscale Vision Transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021. p. 6804–15.
49. Lee-Thorp J, Ainslie J, Eckstein I, Ontanon S. FNet: Mixing Tokens with Fourier Transforms. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). 2022. p. 4296–313.
50. Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In: Proceedings of the 39th International Conference on Machine Learning (ICML). 2022. p. 27268–86.
51. Katharopoulos A, Vyas A, Pappas N, Fleuret F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In: Proc. 37th Int. Conf. Machine Learning (ICML). 2020. p. 5156–65.
52. Wilson G, Cook DJ. Resampling Strongly Imbalanced and Heterogeneous Data for Activity Recognition. vol. 65. 2020. p. 101173. https://doi.org/10.1016/j.pmcj.2020.101173
53. Haresamudram H, Essa I, Plötz T. Contrastive Predictive Coding for Human Activity Recognition. Proc ACM Interact Mob Wearable Ubiquitous Technol. 2021;5(2):1–26. https://doi.org/10.1145/3463506
54. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL. A public domain dataset for human activity recognition using smartphones. In: Proceedings of the European Symposium on Artificial Neural Networks (ESANN). 2013. https://doi.org/10.24432/C54S4K
55. Roggen D, Calatroni A, Rossi M, Holleczek T, Forster K, Troster G, et al. Collecting complex activity datasets in highly rich networked sensor environments. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS). 2010. p. 233–40.
56. Bächlin M, Plotnik M, Roggen D, Giladi N, Hausdorff JM, Tröster G. A wearable system to assist walking of Parkinson s disease patients. Methods Inf Med. 2010;49(1):88–95. https://doi.org/10.3414/ME09-02-0003 PMID: 20011807
57. Reiss A, Stricker D. Introducing a new benchmarked dataset for activity monitoring. In: Proceedings of the 16th International Symposium on Wearable Computers. Dataset: PAMAP2 Physical Activity Monitoring, UCI Machine Learning Repository (ID 231). 2012. p. 108–9. https://doi.org/10.24432/C5NW2H
58. Mim TR, Amatullah M, Afreen S, Yousuf MA, Uddin S, Alyami SA, et al. GRU-INC: An inception-attention based approach using GRU for human activity recognition. Expert Syst Appl. 2023;216:119419. https://doi.org/10.1016/j.eswa.2022.119419
59. Yu X, Al-Qaness MAA. Human Activity Recognition Using Deep Residual Convolutional Network Based on Wearable Sensors. IEEE J Biomed Health Inform. 2025;29(3):1950–8. https://doi.org/10.1109/JBHI.2024.3510860 PMID: 40030576