Practically Adopting Human Activity Recognition

Huatao Xu[1], Pengfei Zhou[2], Rui Tan[1], Mo Li[1,3]
[1]Nanyang Technological University, [2]University of Pittsburgh,
[3]Hong Kong University of Science and Technology
Email:{huatao001, tanrui}@ntu.edu.sg, pengfeizhou@pitt.edu, lim@cse.ust.hk

ABSTRACT

Existing inertial measurement unit (IMU) based human activity recognition (HAR) approaches still face a major challenge when adopted across users in practice. The severe heterogeneity in IMU data significantly undermines model generalizability in wild adoption. This paper presents UniHAR, a universal HAR framework for mobile devices. To address the challenge of data heterogeneity, we thoroughly study augmenting data with the physics of the IMU sensing process and present a novel adoption of data augmentations for exploiting both unlabeled and labeled data. We consider two application scenarios of UniHAR, which can further integrate federated learning and adversarial training for improved generalization. UniHAR is fully prototyped on the mobile platform and introduces low overhead to mobile devices. Extensive experiments demonstrate its superior performance in adapting HAR models across four open datasets.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → *Machine learning approaches.*

KEYWORDS

Mobile sensing, human activity recognition, IMU, physics-informed data augmentation

ACM Reference Format:
Huatao Xu, Pengfei Zhou, Rui Tan, and Mo Li. 2023. Practically Adopting Human Activity Recognition. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3570361.3613299

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3613299

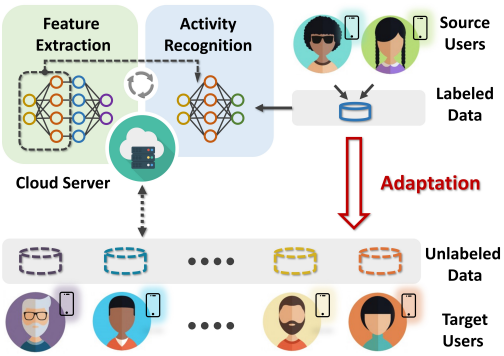


Figure 1: A universal HAR scenario.

1 INTRODUCTION

Human activity recognition (HAR) has played a critical role in considerable real-world applications. Existing studies [8, 9, 15, 25, 29, 40, 64] have explored the possibility of pervasive HAR sensing with inertial measurement unit (IMU) sensors in commodity smart devices. A significant challenge arises when most existing approaches are adopted at scale, i.e., the data heterogeneity caused by real-world diversities (e.g., different devices and usage patterns) leads to degraded performance when HAR models are applied across different user groups. The straightforward solution in addressing such heterogeneity is to collect a substantial amount of labeled data from each of the numerous users, which, however, is prohibitive in practice due to its high overhead.

This paper is motivated by an essential question: *can we have a universal framework that supports applying HAR models across different user groups of real-world diversities, and with realistic adoption overhead?* As depicted in Figure 1, we consider an HAR application scenario where a large number of mobile users have their own data locally collected. The IMU data of the *target users* are implicitly collected during their daily lives and thus unlabeled. The only labeled data are shared from a small group of participating users (*source users*), which are of small size and may be biased in terms of users, usage patterns, devices, or environments. The raw data transmissions from the *target users* to the cloud server are highly undesirable due to the prohibitive processing overhead for the cloud server as well as the related privacy concerns. The

objective is to transfer HAR models from the *source users* to *target users* with realistic adoption overhead.

We find existing works poor in the HAR scenario envisioned in Figure 1. Conventional supervised learning models [15, 25, 27, 64] assume collected labeled dataset is general and thus suffer from severe performance degradation in practice. Recent self-supervised learning works [11, 12, 40, 52, 62], including those aiming at building foundation models for IMU sensing, e.g., TPN [40] and LIMU-BERT [62], however, may still overlook the data heterogeneity and overfit to specific user domains. Some domain adaptation works [4, 17, 37, 65] consider certain aspects of diversity but require fully labeled data from source users, which still underperform when source domain labels are limited. We notice that most existing efforts are focused on directly learning common features among raw data, with the implied assumption that data across different domains already share similar distributions. However, this assumption does not hold when the sensor data collected from different user groups are highly heterogeneous. As a result, most existing approaches fail to achieve satisfactory performance in practical adoptions at scale.

This paper explores the data augmentation perspective to combat data heterogeneity by incorporating physical knowledge. Most existing IMU data augmentation approaches are directly borrowed from other application domains (e.g., images or text processing [45, 46, 60]) without considering and exploiting the physics of inertial sensing, which can lead to harmful results when improperly adopted. We thoroughly study a variety of IMU data augmentation methods and classify them into three categories based on their relations with underlying physical processes: *complete* - which fully aligns with physics, *approximate* - which captures underlying physics but with approximate formulations, and *flaky* - which is not supported by the physical process and may undermine data distribution. The data augmentation with physical priors does not introduce extra labeling overhead and would generalize data distributions. We refer to this technique as *Physics-Informed Data Augmentation*, as opposed to the conventional data plane approaches that disregard underlying physical processes.

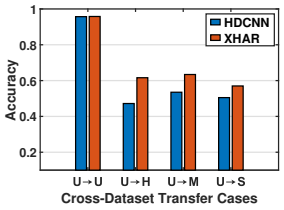
By applying the carefully designed data augmentation approaches, this paper presents UniHAR, a universal HAR framework that extracts generalizable activity-related representations from heterogeneous IMU data. UniHAR comprises two stages as shown in Figure 1 - i) self-supervised learning for feature extraction with massive unlabeled data from all users, and ii) supervised training for activity recognition with limited labeled data from the source users. Catering to the nature of different augmentation methods, UniHAR only applies complete data augmentation during the feature extraction stage to align data distributions from various user groups. On the other hand, both complete and approximate data augmentations are applied during the supervised training stage to increase data diversity for better generalization.

In practical applications, UniHAR is a configurable framework that can adapt to two scenarios, i.e., *data-decentralized* and *data-centralized* scenarios. In the data-decentralized scenario where raw data transmission is not encouraged, as illustrated in Figure 1, UniHAR integrates self-supervised and federated learning techniques to train a generalized feature extraction model. UniHAR then constructs an activity recognition model using limited but augmented labeled data. The recognition model is distributed to all users for activity inference without additional training. In the data-centralized scenario, where raw data transmissions from target users are possible, UniHAR can further leverage adversarial training techniques for improved performance.

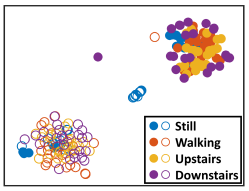
For experiment evaluation, different from previous works [11, 12, 15, 27, 33, 40, 47, 52, 62–64], UniHAR is fully prototyped on the mobile platform. The client is deployable on Android devices, which supports real-time model training and inference with locally collected data. We conduct extensive experiments with four open datasets by transferring models across datasets, i.e., the activity recognition models are trained with activity labels from only one dataset and then applied to the other three datasets without activity labels. To the best of our knowledge, such a level of heterogeneity involved in the experiment settings has not been investigated in existing studies. The results show UniHAR achieves an average HAR accuracy of 78.5% as compared to <62% achieved by extending any existing solutions as alternatives. When the raw data transmissions are allowed in the data-centralized scenario, UniHAR can achieve 82.5% average accuracy as compared to <72% achieved with state-of-the-art solutions. The key contributions of this paper are summarized as follows:

- We consider a practical and challenging HAR scenario, where models trained from a small group of source users are adopted across massive target users with realistic adoption overhead.
- We present a thorough and comprehensive analysis of IMU data augmentation methodology and characterize *physics-informed data augmentation* based on the underlying physics of IMU sensing.
- We identify a novel approach that organically integrates different data augmentation methods into a self-supervised learning framework to address data heterogeneity.
- We fully prototype UniHAR on the standard mobile platform and evaluate its generalization with practical experiment settings across different datasets. The source codes are publicly available [1].

[1] https://dapowan.github.io/wands_unihar/



(a) Across dataset HAR.



(b) Data visualization.

Figure 2: Impact of real-world diversities. (a) The performance of domain adaptation models transferred from the UCI (U) to other datasets (H,M,S) in Table 2. (b) Solid dots and circles represent samples of the UCI and MotionSense datasets, respectively.

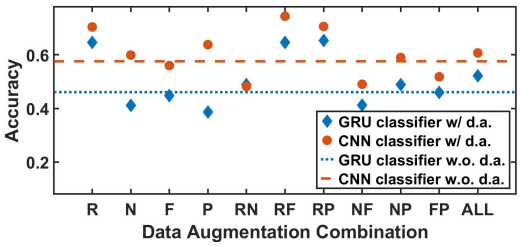


Figure 3: Performance of different data augmentation methods. The "R", "N", "F" and "P" denote rotation, noising, flipping, and permutation. Two letters represent the combination of two methods and "ALL" is the combination of all methods.

2 MOTIVATION

2.1 Data Heterogeneity

The different users, devices, placements, and environments cause data diversity for body-worn IMU-based HAR applications [16, 50, 61, 62]. Most existing works [11, 12, 15, 27, 40, 47, 52, 62–64] overlook the heterogeneity problem and would underperform in practice. Only a few domain adaptation-based works [4, 17, 37, 65] aim at mitigating the impact of certain aspects of diversity. To investigate how those approaches perform with such data heterogeneity, we adopt HDCNN [17] and XHAR [65] to distinguish activity types and examine their performance across datasets. We choose two open datasets (i.e., UCI [38] and MotionSense [31]) with details provided in Section 7, which are collected with different user groups, placements, devices, and environments. The two models are trained to transfer from the UCI dataset to the other three datasets. The detailed settings are provided in Section 7.1. As shown in Figure 2(a), the two models can handle the diversity in the original dataset and achieve nearly 100% classification accuracy. However, when applied across datasets, they suffer from significant performance degradation. Similarly, ASTTL as reported in [37] only yields an average accuracy of 66.3% when transferred across datasets. In summary, there exists a gap in addressing the data heterogeneity when adopting HAR in practice.

To investigate why the models do not perform well in the experiment, we select the common activity types of the two datasets and visualize the raw IMU reading with t-distributed Stochastic Neighbor Embedding (t-SNE) [56] in 2D space. The raw data have the same sampling rate and window size. Figure 2(b) clearly suggests that the IMU data of the same activity type are totally mismatched between the two datasets. Existing works [17, 65] may fail to handle the significant data distribution gap and thus cannot achieve satisfactory performance in cross-dataset HAR.

2.2 IMU Data Augmentation

Being targeted to improving the diversity and size of training data to prevent overfitting, data augmentation has been a commonly employed technique in various application domains [45, 46, 59, 60]. Prior studies [4, 36, 40, 52, 55, 57] directly borrow such a technique and apply many augmentation methods on IMU data for improved performance, e.g., adding random noise, rotation, flipping, etc. However, it remains unclear how effective these methods from other application domains are in handling IMU data heterogeneity. To this end, we apply some classical data augmentation methods to augment data from the UCI dataset as an example. We then train two widely adopted deep learning classifiers, i.e., the GRU [62] and CNN [63] classifiers, with the augmented data and test their performance on the MotionSense dataset. Our results presented in Figure 3 indicate that many data augmentation methods do not improve the cross-dataset HAR performance of the two classifiers. Some methods, such as noising (N) and flipping (F), even negatively impact the end performance. These findings show that many IMU data augmentation methods may not effectively increase IMU data diversity and prevent trained models from overfitting.

As our experimental results suggest, although data augmentation for images or text has been well-established [45, 46, 59, 60], a blind adoption of those may not work for IMU data. This is because IMU sensor readings are observations of the underlying physical states of device movement, e.g., the device orientation. Conventional data augmentation like flipping does not consider IMU sensing physics and directly applying it to IMU data may generate readings that do not adhere to underlying sensing principles. Such unconstrained data generation may lead to biased or even wrong data distributions and as a result degrade the performance of trained models. It remains a challenge to design effective IMU data augmentation approaches and appropriately adopt them to improve model performance.

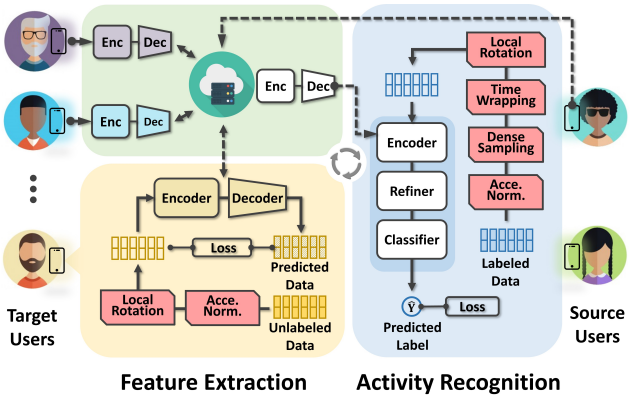


Figure 4: UniHAR overview.

The virtual IMU technique [19, 20] aims at converting videos of human activity into virtual streams of IMU measurement to augment the training data, which follows legitimate physical processes. Virtual IMU, however, requires additional sensing information, including activity videos and the on-body position of the device, to reconstruct the physical states of devices and thus generate virtual IMU data. It cannot be generally applied when the additional camera sensing modality is not available.

2.3 Experiment Setting

We also notice that the experiment settings of most existing studies [4, 5, 11–13, 15, 17, 27, 32, 41, 42, 47, 50, 52, 52, 62–65] are still not practical - the proposed HAR models are primarily evaluated with single IMU datasets. Although some works [4, 40, 41, 52, 62] employ multiple datasets, models are evaluated on individual datasets separately and no cross-dataset evaluation is presented. The data from single datasets can be highly biased in various ways, e.g., different users and devices, on-body positions, and environments. As suggested by the results in Figure 2(a), the evaluation results with a single dataset may not generalize across variable datasets, which however is essential for practical adoptions. In this paper, all identified approaches are systematically evaluated by transferring from one dataset to multiple other datasets in order to investigate their generalizability, which to the best of our knowledge is the first time.

3 UNIHAR OVERVIEW

3.1 Problem Definition

We consider an HAR framework consisting of a cloud server and a number of clients (users). As shown in Figure 1, each client has a local IMU dataset collected by single or multiple mobile devices. The cloud server has some initial datasets shared by a small group of clients (the source users in Figure 1), which are defined as *source domain* $\mathcal{D}_S = \{X_i^s\}_{i=1}^{n_s}$. The local datasets of other clients (the target users) are defined as the *target domain* $\mathcal{D}_T = \{X_i^t\}_{i=1}^{n_t}$. The X_i^s or $X_i^t \in \mathbb{R}^{F\times M}$ represents one IMU sample, where F is the number of sensor features and M is the number of IMU readings. Only a small fraction of $\mathcal{D}_S$ is annotated with activity labels, denoted as $\mathcal{D}_L = \{X_i^l, y_i^l\}_{i=1}^{n_l}$. The $\mathcal{D}_L$ may be biased to a limited number of combinations of {*device, placement, user, environment*}. Mobile clients can communicate with the cloud server and exchange necessary information (e.g., trained models). The objective of the framework is to achieve high activity recognition accuracy for the clients in the *target domain.*

3.2 Overview

As depicted in Figure 4, UniHAR has two training stages:

■ **Feature Extraction**. All local unlabeled datasets are first augmented to align the distributions of heterogeneous data from various clients. To construct a generalized feature extractor (i.e., the *encoder*), the cloud server collaborates with all mobile clients to exploit massive augmented unlabeled data. The encoder and decoder are trained on clients individually, which learn the high-level features using self-supervised learning techniques. The cloud server combines local models and obtains a generalized model. In a nutshell, the whole process aims at solving the following problem:

$$w^* = \arg\min_{w} \ell_r(w; \mathcal{D}_S, \mathcal{D}_T), \tag{1}$$

where ℓ_r denotes the loss function and w denotes the weights of the *encoder* and *decoder.*

■ **Activity Recognition**. Based on the generalized encoder, the server then adopts a small amount of labeled data from source users and trains an activity recognition model. Data augmentation is also integrated to enrich the diversity of labeled data and narrow the distribution gap between the source and target domains. The activity recognizer, including *encoder*, *refiner*, and *classifier*, jointly learn to recognize activity types of labeled IMU data. The training process can be represented by

$$c^* = \arg\min_{c} \ell_c(w^*, c; \mathcal{D}_L), \tag{2}$$

where ℓ_c denotes the loss function and c represents the weights of the activity recognizer. After the server dispatches the recognizer, each client utilizes it to classify activities without additional training.

4 PHYSICS-INFORMED DATA AUGMENTATION

To mitigate the data heterogeneity, UniHAR enriches the IMU data diversity based on physical knowledge and assists the learning of generalizable features.

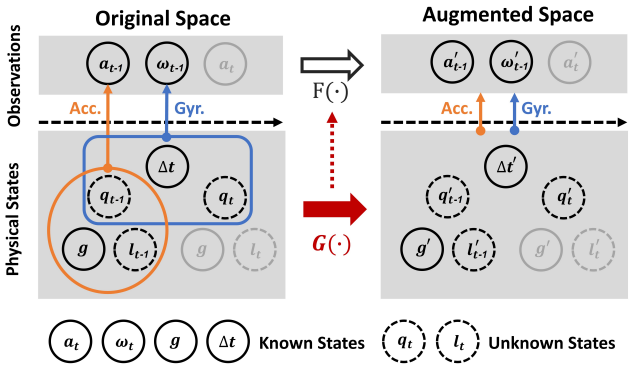


Figure 5: Physics-informed IMU data augmentation.

4.1 Physical Sensing Models

UniHAR augments both accelerometer and gyroscope sensor readings, which provide more modality information for HAR [33, 62]. The measured acceleration is defined as

$$\boldsymbol{a} = S_a(\boldsymbol{q}, \boldsymbol{l}, \boldsymbol{g}) = \boldsymbol{q}^* \otimes (\boldsymbol{l} + \boldsymbol{g}) \otimes \boldsymbol{q}, \tag{3}$$

where $\boldsymbol{l}$ and $\boldsymbol{g}$ denote the acceleration caused by the movement of the device and gravity in the global frame, respectively. The unit quaternion $\boldsymbol{q}$ represents the orientation of the device which is the rotation from the global frame to the local (body) frame. The $\boldsymbol{q}^*$ is the conjugation of $\boldsymbol{q}$ and $\otimes$ is the Hamilton product. The acceleration reading $\boldsymbol{a}$ is a rotated vector of the addition of $\boldsymbol{l}$ and $\boldsymbol{g}$ in the local frame. The gyroscope measures the angular velocity $\boldsymbol{\omega}$ that can be used to derive the change of orientation $\boldsymbol{q}$ by the formula[2] as follows:

$$\boldsymbol{q_t} = \boldsymbol{q_{t-1}} + \frac{\Delta t}{2} \boldsymbol{q_{t-1}} \otimes \boldsymbol{\omega_t}, \tag{4}$$

where Δt is a small value, e.g., 0.01s, denoting the sampling interval between $\boldsymbol{q_t}$ and $\boldsymbol{q_{t-1}}$. By transforming Equation (4), the sensing model of angular velocity is

$$\boldsymbol{\omega_t} = S_\omega(\boldsymbol{q}, \Delta t) = \frac{2}{\Delta t} \boldsymbol{q}^*_{t-1} \otimes (\boldsymbol{q_t} - \boldsymbol{q_{t-1}}). \tag{5}$$

4.2 Data Augmentation Model

In this paper, we propose a general model for IMU data augmentation as indicated in Figure 5. The $\boldsymbol{q}$, $\boldsymbol{l}$, $\boldsymbol{g}$, and Δt are underlying physical states of the device, and sensor readings $\boldsymbol{a}$ and $\boldsymbol{\omega}$ are observations from the physical states:

$$(\boldsymbol{q}, \boldsymbol{l}, \boldsymbol{g}) \xrightarrow{S_a} \boldsymbol{a}, \ (\boldsymbol{q}, \Delta t) \xrightarrow{S_\omega} \boldsymbol{\omega}, \tag{6}$$

where S_a and S_ω indicate the accelerometer and gyroscope sensing models, respectively. In practice, the $\boldsymbol{q}$ and $\boldsymbol{l}$ are typically unknown (dashed circles in Figure 5) while other physical states and observations are known (solid circles in Figure 5). A data augmentation is a mapping $F(\cdot)$ that transforms observations from the original space to the augmented space:

$$(\boldsymbol{a}, \boldsymbol{\omega}) \xrightarrow{F} (\boldsymbol{a'}, \boldsymbol{\omega'}), \tag{7}$$

We introduce the concept of *physical embedding* $G(\cdot)$ to align the mapping $F(\cdot)$ between observations with the underlying physical principles, which is defined as:

Definition 1. *$G(\cdot)$ is a physical embedding of $F(\cdot)$ if $G(\cdot)$ transforms physical states by*

$$(\boldsymbol{q}, \boldsymbol{l}, \boldsymbol{g}, \Delta t) \xrightarrow{G} (\boldsymbol{q'}, \boldsymbol{l'}, \boldsymbol{g'}, \Delta t'), \tag{8}$$

such that the observations from the transformed physical states equal the augmented observations of $F(\cdot)$:

$$(\boldsymbol{q'}, \boldsymbol{l'}, \boldsymbol{g'}) \xrightarrow{S_a} \boldsymbol{a'}, \ (\boldsymbol{q'}, \Delta t') \xrightarrow{S_\omega} \boldsymbol{\omega'}. \tag{9}$$

In practice, a mapping $F(\cdot)$ that has a physical embedding $G(\cdot)$ indicates the transition of readings can take place through a physical process in reality. In this paper, we thus define three types of data augmentation based on the above mathematical model:

- **Complete data augmentation**, where its mapping $F(\cdot)$ is connected with a physical embedding $G(\cdot)$, and $F(\cdot)$ can be fully formulated with original observations and known physical states.
- **Approximate data augmentation**, where its mapping $F(\cdot)$ is connected with a physical embedding $G(\cdot)$, but $F(\cdot)$ involves unknown physical states and can be approximated by a formulation of known states.
- **Flaky data augmentation**, where we cannot find a physical embedding $G(\cdot)$ to support its mapping $F(\cdot)$.

In this paper, we refer complete and approximate data augmentations to *physics-informed data augmentations*, which both have underlying support of physical embeddings. In the following, we characterize each of the three data augmentation types based on which we develop more effective augmentation adoption strategies that are grounded in the physical principles of IMU sensing.

4.2.1 Complete data augmentation. This type of data augmentation accurately generates augmented observations using the known physical states and original observations. We elaborate on a few instances.

Acceleration normalization. Accelerometer and gyroscope readings usually have different distributions. The range difference may affect the performance of deep learning models [62]. A simple method is to narrow the difference by normalizing accelerometer readings with the gravity (9.81 m/s^2), i.e, $\boldsymbol{a'} = F(\boldsymbol{a}) = \frac{\boldsymbol{a}}{\|\boldsymbol{g}\|}$. There exists a physical embedding $\boldsymbol{l'} = G(\boldsymbol{l}) = \frac{\boldsymbol{l}}{\|\boldsymbol{g}\|}$, $\boldsymbol{g'} = G(\boldsymbol{g}) = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$. Other physical states

[2]Note that Equation (4) is an approximation formula but studies [30, 39, 48] have shown that it works well in practice when Δt is small.

including orientation $\boldsymbol{q}$ and time interval Δt remain the same. The acceleration of transformed physical states is

$$\begin{aligned} S_a(\boldsymbol{q}', \boldsymbol{l}', \boldsymbol{g}') &= \boldsymbol{q}'^{*} \otimes (\boldsymbol{l}' + \boldsymbol{g}') \otimes \boldsymbol{q}' \\ &= \boldsymbol{q}^{*} \otimes (\frac{\boldsymbol{l}+\boldsymbol{g}}{\|\boldsymbol{g}\|}) \otimes \boldsymbol{q} = \frac{\boldsymbol{a}}{\|\boldsymbol{g}\|} = \boldsymbol{a}'. \end{aligned} \tag{10}$$

The $F(\boldsymbol{a})$ only involves the known physical state $\boldsymbol{g}$, so acceleration normalization is a complete data augmentation.

Local rotation. The placement diversity causes significant differences in triaxial distributions of IMU data. To simulate the IMU data collected from different device orientations, this augmentation applies an extra rotation to the device and augments orientation in the local frame by $\boldsymbol{q}' = G(\boldsymbol{q}) = \boldsymbol{q} \otimes \Delta\boldsymbol{q}$, where $\Delta\boldsymbol{q}$ is generated rotation and known. The observations of transformed physical states are

$$\begin{aligned} S_a(\boldsymbol{q}', \boldsymbol{l}', \boldsymbol{g}') &= (\boldsymbol{q} \otimes \Delta\boldsymbol{q})^{*} \otimes (\boldsymbol{l}' + \boldsymbol{g}') \otimes (\boldsymbol{q} \otimes \Delta\boldsymbol{q}) \\ &= \Delta\boldsymbol{q}^{*} \otimes \boldsymbol{q}^{*} \otimes (\boldsymbol{l} + \boldsymbol{g}) \otimes \boldsymbol{q} \otimes \Delta\boldsymbol{q} \\ &= \Delta\boldsymbol{q}^{*} \otimes \boldsymbol{a} \otimes \Delta\boldsymbol{q}, \end{aligned} \tag{11}$$

$$\begin{aligned} S_\omega(\boldsymbol{q}', \Delta t') &= \frac{2}{\Delta t}(\boldsymbol{q} \otimes \Delta\boldsymbol{q})^{*} \otimes (\boldsymbol{q}_t \otimes \Delta\boldsymbol{q} - \boldsymbol{q}_{t-1} \otimes \Delta\boldsymbol{q}) \\ &= \frac{2}{\Delta t}\Delta\boldsymbol{q}^{*} \otimes \boldsymbol{q}^{*} \otimes (\boldsymbol{q}_t - \boldsymbol{q}_{t-1}) \otimes \Delta\boldsymbol{q} \\ &= \Delta\boldsymbol{q}^{*} \otimes \boldsymbol{\omega} \otimes \Delta\boldsymbol{q}. \end{aligned} \tag{12}$$

The $F(\cdot)$ can be designed with $\boldsymbol{a}' = F(\boldsymbol{a}) = \Delta\boldsymbol{q}^{*} \otimes \boldsymbol{a} \otimes \Delta\boldsymbol{q}$ and $\boldsymbol{\omega}' = F(\boldsymbol{\omega}) = \Delta\boldsymbol{q}^{*} \otimes \boldsymbol{\omega} \otimes \Delta\boldsymbol{q}$. The augmented observations can be derived from original observations and known $\Delta\boldsymbol{q}$, so local rotation is a complete data augmentation. The local rotation significantly diversifies the triaxial distributions of the original readings and maintains other human motion information, e.g., the magnitude and the fluctuation pattern.

Dense sampling. Existing studies [5, 17, 40, 50, 62, 64, 65] simply divide IMU readings using low overlapping rates (e.g., zero or 50% overlapping) and as a result underutilize the data. To fully use the collected IMU data, higher overlapping rates with dense sampling may be adopted. The rationale is that most daily activities are periodic, which means any time can be viewed as the start of the motion. Dense sampling shifts observations along the temporal dimension by $\boldsymbol{a}'_t = F(\boldsymbol{a}) = \boldsymbol{a}_{t+n}$, $\boldsymbol{\omega}'_t = F(\boldsymbol{\omega}) = \boldsymbol{\omega}_{t+n}$, where n is a random value. The augmented observations are partitioned with a fixed window and then put into HAR models for training, which can enlarge the number of training samples with existing sensor readings. Its physical embedding is shifting the physical states by n accordingly, e.g., $\boldsymbol{l}'_t = G(\boldsymbol{l}) = \boldsymbol{l}_{t+n}$. The $F(\cdot)$ does not require unknown physical states and dense sampling is also a complete data augmentation.

4.2.2 Approximate data augmentation. This type of data augmentation has its physical embedding, but the augmented observations depend on the approximation of original observations or known physical states.

Linear Upsampling. IMU data are discrete signals and upsampling can enrich data samples. The linear upsampling interpolates physical states by $\boldsymbol{q}'_{\tilde{t}} = G(\boldsymbol{q}) = \alpha\boldsymbol{q}_t + (1-\alpha)\boldsymbol{q}_{t-1}$ and $\boldsymbol{l}'_{\tilde{t}} = G(\boldsymbol{l}) = \alpha\boldsymbol{l}_t + (1-\alpha)\boldsymbol{l}_{t-1}$, where α is a value within $[0, 1]$ and $\tilde{t} = \alpha t + (1-\alpha)(t - \Delta t) = t - (1-\alpha)\Delta t$. The corresponding augmented observations are

$$S_a(\boldsymbol{q}', \boldsymbol{l}', \boldsymbol{g}') = \boldsymbol{q}'^{*}_{\tilde{t}} \otimes (\boldsymbol{l}'_{\tilde{t}} + \boldsymbol{g}) \otimes \boldsymbol{q}'_{\tilde{t}}, \tag{13}$$

$$S_\omega(\boldsymbol{q}', \Delta t') = \frac{2}{\Delta t}\boldsymbol{q}^{*}_{\tilde{t}-1} \otimes (\boldsymbol{q}_{\tilde{t}} - \boldsymbol{q}_{\tilde{t}-1}), \tag{14}$$

both involving unknown physical states $\boldsymbol{q}$ and $\boldsymbol{l}$. Linear upsampling is an approximate data augmentation and its augmented observations can be approximated as

$$\begin{aligned} \boldsymbol{a}' &= S_a(\boldsymbol{q}', \boldsymbol{l}', \boldsymbol{g}') = \boldsymbol{q}'^{*}_{\tilde{t}} \otimes (\alpha\boldsymbol{l}_t + (1-\alpha)\boldsymbol{l}_{t-1} + \boldsymbol{g}) \otimes \boldsymbol{q}'_{\tilde{t}} \\ &= \alpha\boldsymbol{q}'^{*}_{\tilde{t}} \otimes (\boldsymbol{l}_t + \boldsymbol{g}) \otimes \boldsymbol{q}'_{\tilde{t}} + (1-\alpha)\boldsymbol{q}'^{*}_{\tilde{t}} \otimes (\boldsymbol{l}_{t-1} + \boldsymbol{g}) \otimes \boldsymbol{q}'_{\tilde{t}} \\ &\approx \alpha\,\boldsymbol{q}^{*}_t \otimes (\boldsymbol{l}_t + \boldsymbol{g}) \otimes \boldsymbol{q}_t + (1-\alpha)\,\boldsymbol{q}^{*}_{t-1} \otimes (\boldsymbol{l}_{t-1} + \boldsymbol{g}) \otimes \boldsymbol{q}_{t-1} \\ &= \alpha\boldsymbol{a}_t + (1-\alpha)\boldsymbol{a}_{t-1}, \end{aligned} \tag{15}$$

$$\begin{aligned} \boldsymbol{\omega}' &= S_\omega(\boldsymbol{q}', \Delta t) = \frac{2}{\Delta t}\boldsymbol{q}^{*}_{\tilde{t}-1} \otimes (\alpha(\boldsymbol{q}_t - \boldsymbol{q}_{t-1}) - \\ &\qquad (1-\alpha)(\boldsymbol{q}_{t-1} - \boldsymbol{q}_{t-2})) \\ &\approx \frac{2\alpha}{\Delta t}\boldsymbol{q}^{*}_{t-1} \otimes (\boldsymbol{q}_t - \boldsymbol{q}_{t-1}) + \frac{2(1-\alpha)}{\Delta t}\boldsymbol{q}^{*}_{t-2} \otimes (\boldsymbol{q}_{t-1} - \boldsymbol{q}_{t-2}) \\ &= \alpha\boldsymbol{\omega}_t + (1-\alpha)\boldsymbol{\omega}_{t-1}, \end{aligned} \tag{16}$$

where $\boldsymbol{q}'_{\tilde{t}}$ approximately equals to $\boldsymbol{q}_{t-1}$ or $\boldsymbol{q}_t$ if Δt is small. Linear upsampling enlarge the size of data but introduces approximation errors.

Time wrapping. A same type of activities may vary in duration across users due to distinct behavioral patterns. To mitigate the temporal divergence, time wrapping accelerates or decelerates changes of physical states in the temporal dimension, e.g., $\boldsymbol{q}'_t = G(\boldsymbol{q}) = \boldsymbol{q}_{k*t}$, where k is a scaling factor usually chosen within $[0.8, 1.2]$. The augmented observations are accordingly stretched in the temporal dimension, e.g., $\boldsymbol{a}'_t = F(\boldsymbol{a}) = \boldsymbol{a}_{k*t}$. To facilitate such a transformation, time wrapping adopts linear upsampling to obtain continuous observations. Therefore, time wrapping is also approximate data augmentation, which enhances temporal diversity and also with approximation errors.

4.2.3 Flaky data augmentation. We find many IMU data augmentation methods, although widely adopted in existing works [4, 36, 40, 52, 55, 57], do not have their physical embeddings. For example, some data augmentations randomly negate observations [36, 40, 52] or reverse the observations along the temporal dimension [36, 40, 52]. The permutation [36, 40, 52, 55] slices the observations sequence within a temporal window and randomly swaps sliced segments to generate a new sequence. The shuffling [36, 40, 52] randomly

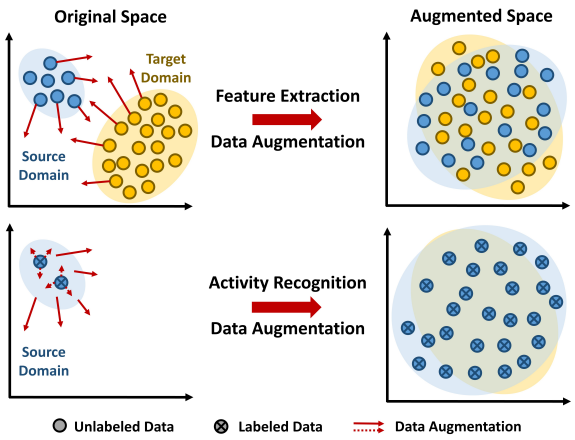


Figure 6: Adoption of data augmentations.

rearranges the channels of sensor observations to change the triaxial distribution. These methods may not be associated with the underlying physical principles.

The jittering [36, 40, 52, 55] that adds additional random noise to the original observations is a special flaky data augmentation. It aims at augmenting the sensor model by introducing the sensor noises. But the applied noise distributions may not match the true distributions, which vary across different devices and are hard to determine [50].

Flaky data augmentations only operate with IMU observations and are not explainable with the underlying physical process. The adoption of them may lead to unbounded errors in the generated data distributions.

4.3 Data Augmentation Adoption

UniHAR incorporates *physics-informed data augmentation* methods differently during the two stages of the framework based on their respective characteristics and Figure 6 explains the rationale.

During the feature extraction stage, although unlabeled data are abundant, they are from different users, devices, and environments, which are subject to significant domain shift. The purpose of incorporating data augmentation is to generalize the data distributions and improve the inter-domain data representativeness (as illustrated in the top of Figure 6). On the other hand, it is challenging to control the data quality when approximation errors are introduced at scale. Therefore, UniHAR only employs complete data augmentations to unlabeled data in this stage.

During the activity recognition stage, labeled data from the source domain are utilized but they are scarce. In addition to aligning the data distributions across domains, the data augmentation is also expected to enrich the source domain labels and improve the intra-domain data representativeness (as illustrated in the bottom of Figure 6). Since supervised training with labels is more robust to errors [3], a wider range of data augmentation methods can be integrated and UniHAR applies both complete and approximate data augmentation to augment labeled data in this stage.

Flaky data augmentations are prohibited throughout the entire training process because they may lead to completely wrong data distributions. We thoroughly investigate the effect of data augmentation methods with experiments and show how their varied usage can either improve or deteriorate model performance in Section 7.4.

5 UNIHAR ADOPTION

Putting UniHAR to practical adoption, we consider two application scenarios and make further optimizations for improved performance.

5.1 Data-decentralized Scenario

In the data-decentralized scenario, the raw data transmission from the target users to the cloud server is supposed not allowed due to practical constraints, e.g., prohibitive processing and transmission overheads or privacy concerns.

5.1.1 Feature extraction. To extract effective features from local unlabeled datasets, UniHAR adopts self-supervised learning to train the encoder and decoder. There are several state-of-the-art self-supervised representation models [12, 40, 52, 62] for IMU data. However, many methods [12, 40, 52] are entangled with data augmentation and flaky data augmentation methods are integrated, which we believe may harm the end performance. We identify LIMU-BERT [62] as an effective foundation model for IMU-based sensing, and embed it into our design to build the representation model. LIMU-BERT is orthogonal to the data augmentation employed in UniHAR.

We employ acceleration normalization and local rotation for data augmentation, both being complete data augmentation. As shown in Figure 4, the encoder and decoder jointly predict the original values of the randomly masked IMU readings. By the reconstruction task, the encoder learns the underlying relations among IMU data and extracts effective features. The Mean Square Error (MSE) loss is used to compute the differences between the original and predicted values, which is defined as follows:

$$\ell_{rec}(w; \boldsymbol{X}) = \frac{1}{|\boldsymbol{X}|} \sum_{i=1}^{|\boldsymbol{X}|} \sum_{j}^{j \in M^{[i]}} \text{MSE}\,(\hat{\boldsymbol{X}}_{\cdot j}^{[i]} - \boldsymbol{X}_{\cdot j}^{[i]}), \qquad (17)$$

where w denotes the model weights of the encoder and decoder, and $M^{[i]}$ represents the set of the position indices of masked readings for the i-th IMU sample $\boldsymbol{X}^{[i]}$. The $\hat{\boldsymbol{X}}^{[i]} \in \mathbb{R}^{F \times m}$ denotes the predicted data as shown in Figure 4.

To avoid the transmission of raw data, UniHAR integrates a federated learning structure to collaborate with all mobile clients and train a more generalized feature extraction model. UniHAR aggregates local models from all clients and obtains a general global model as shown in Figure 4 (green part). In each round of training, the cloud server distributes the latest global model to the clients, which then make use of their individual local datasets to update the model with the ℓ_{rec} defined in Equation (17) for 5 epochs. There are two potential options to aggregate local models: aggregating gradient and aggregating model weights [24]. Our experiments show that the latter does introduce less bias to the clients with more samples and achieves better overall performance. Therefore, the server aggregates local models with weights defined as the ratio of the number of samples at each client to the total number of samples, which can be expressed as $w_g \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_l^k$. The w_g and w_l denote the parameters of global and local models, respectively. The aggregation weight $\frac{n_k}{n}$ is the ratio of the number of samples at the k-th client to the total number of samples n. The process repeats until the global model converges.

To optimize the training process, UniHAR initializes the weights of the encoder and decoder trained with the source domain data such that each mobile client can fine-tune the models with fewer epochs.

5.1.2 Activity recognition. Based on the encoder trained with massive unlabeled data, the server exploits the augmented labeled data from the source users and trains an activity recognizer. Figure 4 gives the workflow in training the recognizer. In addition to acceleration normalization and local rotation, UniHAR further applies dense sampling and time wrapping to augment the source domain labeled data. To control the approximation errors, time wrapping is applied with a probability of 0.4 which is fine-tuned based on our empirical experiments. A refiner is designed to distill the representations and extract activity-specific features. The classifier is trained to recognize activity types with refined features. The training loss is defined as follows:

$$\ell_{act}(w, r, c; \boldsymbol{X}) = \frac{1}{|\boldsymbol{X}|} \sum_{i=1}^{|\boldsymbol{X}|} \mathrm{CE}\left(\hat{y}^{[i]}, y^{[i]}\right), \tag{18}$$

where ℓ_{act} is defined with the Cross-Entropy (CE) loss, r and c represent the weights of the refiner and classifier, respectively. The $\hat{y}^{[i]}$ and $y^{[i]}$ are the estimated softmax probability and corresponding ground truth. Note that the encoder is fine-tuned according to ℓ_{act} during training.

In UniHAR, the refiner contains two Gated Recurrent Unit (GRU) layers (bi-directional) with the same hidden sizes of 10 and the input size of 36. Only the hidden features at the last position are input into the classifier, which consists of a dropout layer with a drop rate of 0.5 and a fully-connected layer with 10 units.

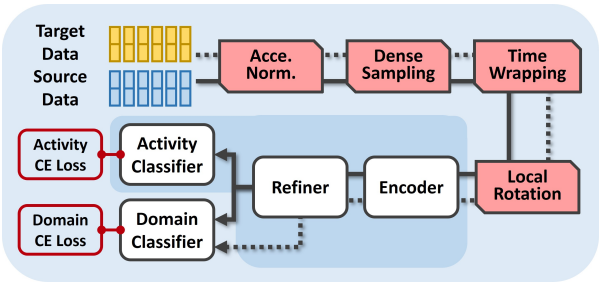


Figure 7: Workflow for training the recognizer in the data-centralized scenario.

5.2 Data-centralized Scenario

We consider a second scenario where some target users may share their unlabeled data with the cloud server for improved HAR performance. In such a case, UniHAR is able to incorporate unsupervised learning techniques to design a more sophisticated activity recognizer and further eliminate the domain discrepancies. Specifically, UniHAR injects extra information, i.e., domain label specifying which domain the IMU data belong to, into the activity recognizer training process using adversarial domain adaptation techniques.

Figure 7 illustrates the workflow of the activity recognition stage in the data-centralized scenario. Both the source and target domain data are augmented and then processed by the encoder and refiner. The *domain classifier* learns to distinguish the domain with the training loss

$$\ell_{dom}(w, r, d; \boldsymbol{X}) = \frac{1}{|\boldsymbol{X}|} \sum_{i=1}^{|\boldsymbol{X}|} \mathrm{CE}\left(\hat{y}_d^{[i]}, y_d^{[i]}\right), \tag{19}$$

where $\hat{y}_d^{[i]}$ and $y_d^{[i]}$ denote the predicted probability and actual domain label, respectively. The weights of the domain classifier d are updated with ℓ_{dom} while the encoder, refiner, and activity classifier are trained with the mixed loss:

$$\ell_{mix}(w, r, c; \boldsymbol{X}) = \ell_{act}(w, r, c; \boldsymbol{X}) - \alpha \cdot \ell_{dom}(w, r, d; \boldsymbol{X}), \tag{20}$$

where α is a weight set to 0.6. By minimizing ℓ_{mix}, the encoder and refiner are trained against the domain classifier and thus capture domain-independent features, which further mitigates the data heterogeneity issue.

The domain classifier contains two fully-connected layers with the hidden and output size of 72 and 2, respectively. The first fully-connected layer is followed by the Rectified Linear Unit (ReLU) [2] activation function layer.

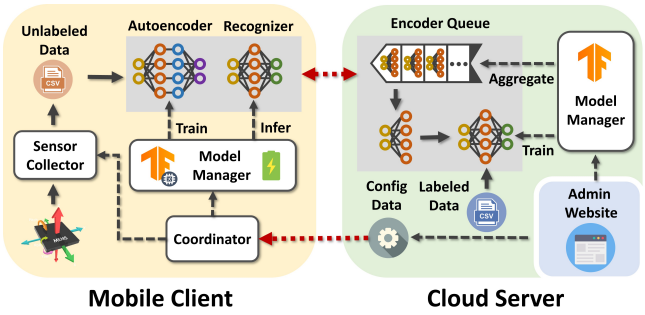


Figure 8: UniHAR implementation.

Table 1: System Overhead.

Model		Autoencoder	Recognizer
Latency	Client	train: 66ms	infer: 25ms
	Server	aggre.: /	train: 15ms
Size		62.6 KB	68.5 KB
Client	CPU	13%	11%
	Memory	102 MB	93 MB
	Energy	light	light

6 IMPLEMENTATION AND SYSTEM EVALUATION

UniHAR is fully prototyped on the mobile platform and Figure 8 illustrates the system designs for the mobile client and the cloud server. The mobile client is an Android application that supports real-time data collection and model training, inference, and sharing. The sensor collector accesses IMU sensors and saves readings as files implicitly. The model manager implemented with TensorFlow lite [1] is then activated to train the autoencoder (including the encoder and decoder) with unlabeled data. The server aggregates all client encoders in the encoder queue and trains the recognizer with labeled data. All components are guided by the configuration data (e.g., sampling rate and batch size) set by the admin website. To avoid affecting the daily use of other applications, the model manager is triggered only when the smartphone is not actively used and is being charged.

The input sensor data are accelerometer and gyroscope readings down-sampled to 20 Hz. The input window contains 20 readings. UniHAR defines the encoder and decoder [62] with R_{num} = 1, A_{dim} = 4, H_{dim} = 36, and F_{dim} = 72. The two stages adopt the same learning rate and batch size, which are 0.001 and 64, respectively. The recognizer is trained for 500 epochs. The model weights are updated with Adam [18] optimizer. To evaluate the system overhead, we install the client on a Samsung Galaxy S8 SM-G9500 (Octa-core CPU and 4 GB RAM). The cloud server is deployed on a computer equipped with an Intel(R) Core(TM) i9-9820X 3.30GHz CPU, 128 GB memory, and four NVIDIA GEFORCE 2080Ti GPUs.

Latency. Table 1 shows the latency of the autoencoder and recognizer on the two platforms. The smartphone requires 66ms and 25ms to train the autoencoder and infer the recognizer for one batch of samples, respectively. The first-time training and inference may take longer time, i.e., 2.5s and 0.5s, respectively, which may be attributed to the model file initialization process. The training time of the recognizer is about 15 ms per batch on the server and the aggregation time depends on the number of autoencoders shared by clients.

Communication overhead. The models are first initiated with Tensorflow lite files on the client. The model weights are exchanged in the format of Tensorflow checkpoint files. The total communication overhead of the federated training process with 100 rounds for each client is about 18.9 MB ($100 \times 2 \times 62.6$ KB + 100×68.5 KB), which is easily affordable with nowadays's 4G/5G data bundles.

Computational overhead and energy consumption. The Android Profiler [6] indicates that the training of autoencoder and inference of recognizer cause about 10% CPU load and require about 100 MB memory on the Samsung Galaxy S8. The energy usages are both below the level of "light" defined by the Android Profiler.

In summary, UniHAR introduces low system overhead to mobiles and the cloud server.

7 COMPARATIVE EVALUATION

To ensure direct and fair comparisons between UniHAR and a variety of existing works with open datasets, we re-build UniHAR and implement baseline models with Pytorch [35] and emulate mobile clients that hold offline local datasets.

7.1 Experiment Setup

7.1.1 Datasets. We evaluate UniHAR with four publicly available datasets, which have been widely used in previous studies [40, 62–64]. These datasets cover a wide variety of {*user, device, placement, environment*} combinations.

■ **HHAR** [50] contains accelerometer and gyroscope readings from 9 users performing 6 different activities (*sitting, standing, walking, upstairs, downstairs,* and *biking*) with 6 types of mobile phones (3 models of Samsung Galaxy and one model of LG). All smartphones were carried by the users around their waists. The dataset was collected in Denmark.

■ **UCI** [38] has raw accelerometer and gyroscope data with 30 volunteers aged from 19 to 48 years from Italy. The readings of 6 basic activities (*standing, sitting, lying, walking, walking downstairs,* and *walking upstairs*) were collected at

50 Hz with a Samsung Galaxy S II carried on the waist.

■ **MotionSense** [31] dataset (abbreviated as Motion in our paper) adopted an iPhone 6s to gather accelerometer and gyroscope time-series data. 24 participants from UK performed 6 activities (*sitting, standing, walking, upstairs, downstairs,* and *jogging*) with the device stored in their front pockets. All data were collected at a 50 Hz sampling rate.

■ **Shoaib** [44] et al. collected data of seven daily activities (*sitting, standing, walking, walking upstairs, walking downstairs, jogging,* and *biking*) in the Netherlands. The 10 male participants were equipped with five Samsung Galaxy SII (i9100) smartphones placed at five on-body positions (*right pocket, left pocket, belt, upper arm, and wrist*). The IMU readings were collected at 50 Hz.

To demonstrate the effectiveness of UniHAR across diverse datasets, we select four common activities (i.e., *still, walk, walk upstairs, walk downstairs*) contained in all four open datasets. For *still* activity, we merge several similar activities, for example, *sit* and *stand* in the HHAR dataset into *still*. In addition to the diversity of {*user, device, placement, environment*}, the merged dataset has general label definitions (i.e., *still*). The activity distributions of the four activities also differ in the four datasets.

7.1.2 Baseline models. We consider both data-decentralized and data-centralized scenarios, and compare UniHAR with relevant state-of-the-art solutions in each scenario. In the data-decentralized scenario, we extend three most relevant approaches as baselines:

■ **DCNN** [63] designs a CNN-based HAR model that outperforms many traditional methods. It assumes labeled data are abundant and adequately representative.

■ **TPN** [40] learns features from unlabeled data by recognizing the applied data augmentations. It only requires limited data for training but with an implicit assumption that they are not biased.

■ **LIMU-GRU** [62] learns representations by a self-supervised autoencoder LIMU-BERT. It uses limited labeled data and assumes unbiased data distributions.

In the data-centralized scenario, we compare UniHAR with three existing unsupervised domain adaptation approaches.

■ **HDCNN** [17] handles domain shift by minimizing the Kullback-Leibler divergence between the fully-labeled source domain features and unlabeled target domain features.

■ **FM** [4] minimizes the feature distance across domains by maximum mean discrepancy [54]. It requires full supervision with adequate labeled data from the source domain.

■ **XHAR** [65] is an adversarial domain adaptation model and needs to select the source domain before adapting models to the unlabeled target domain.

The SelfHAR [52] and ASTTL [37] are not compared because SelfHAR inherits the training scheme from TPN, which is already selected as one of the baselines, and the performance of ASTTL as originally reported in [37] is poor.

Table 2: Cross-dataset transfer setup.

Case	Souce Domain with labels	Target Domain without labels
1	HHAR	UCI, Motion, Shoaib
2	UCI	HHAR, Motion, Shoaib
3	Motion	HHAR, UCI, Shoaib
4	Shoaib	HHAR, UCI, Motion

7.1.3 Cross-dataset evaluation. To demonstrate the generalizability of UniHAR, we design four cross-dataset evaluation cases and Table 2 indicates each of the cases, e.g., in case 1, UniHAR transfers models from the HHAR dataset to other three datasets without activity labels. The clients in the source domain share a small portion of labeled data with the cloud server, while the clients in the target domain only contribute local unlabeled data. Each mobile client has a local dataset containing the IMU data collected from the same user and our setup has a total of 73 clients. Each local dataset is partitioned into training (80%), validation (10%), and test (10%) sets. The training sets of all clients participate in the federated training process of the encoder and decoder. To train the recognizer, we randomly select a small portion of labeled samples (i.e., 1,000) from the training sets in the source domain. The validation sets are utilized to select models (e.g., the encoder and classifier) and the trained recognizers are evaluated on the test sets in the target domain.

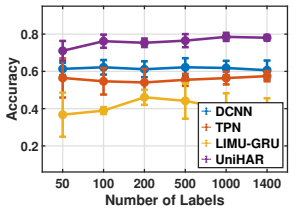
7.1.4 Metrics. We compare the performance of HAR models with average accuracy and F1-score of all users in the target domain, which are defined as $\overline{a} = \sum a_i, \overline{f} = \sum f_i, s.t.\ i \in \mathcal{D}_t$, where a_i and f_i are the activity classification accuracy and F1-score of the i-th user, respectively.

7.2 Overall Performance

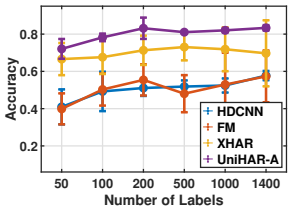
Table 3 compares the performance of UniHAR and other baseline models in the data-decentralized and data-centralized scenarios. The two numbers in each cell denote average accuracy and F1-score, respectively. The row of UniHAR-A gives the performance of the UniHAR recognizer trained with centralized target user data. In the data-decentralized scenario, DCNN, TPN, and LIMU-GRU achieve poor performance mainly because they overlook the data diversity among the source and target domain users. In contrast, UniHAR achieves 78.5% average accuracy and 67.1% F1-score, which outperforms the best of three baselines by at least 15%. For the data-centralized scenario, UniHAR-A also yields better accuracies and F1-scores when compared with HDCNN, FM, and XHAR in most cases. Although XHAR delivers the

Table 3: Performance comparison. (The two numbers in each cell are accuracy and F1-score.)

Scenario	Model	Cross-Dataset Transfer Case				Average
		1	2	3	4	
Data Decentralized	DCNN	0.594, 0.438	0.583,0.373	0.628, 0.437	0.668, 0.465	0.618, 0.428
	TPN	0.584, 0.361	0.530,0.281	0.541, 0.302	0.601, 0.350	0.564, 0.324
	LIMU-GRU	0.306, 0.174	0.435, 0.178	0.353, 0.248	0.497, 0.337	0.398, 0.234
	UniHAR	**0.757**, **0.611**	**0.785**, **0.667**	**0.789**, **0.704**	**0.810**, **0.702**	**0.785**, **0.671**
Data Centralized	HDCNN	0.557, 0.439	0.515, 0.233	0.487, 0.293	0.518, 0.376	0.524, 0.335
	FM	0.386, 0.250	0.757, 0.507	0.410, 0.273	0.564, 0.369	0.539, 0.350
	XHAR	0.648, 0.430	0.615, 0.433	0.733, 0.566	**0.879**, **0.777**	0.719, 0.552
	UniHAR-A	**0.805**, **0.674**	**0.833**, **0.708**	**0.819**, **0.731**	0.824, 0.723	**0.820**, **0.709**

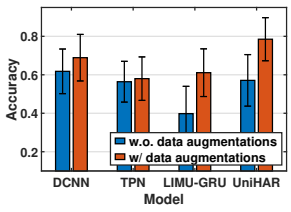


(a) Data-decentralized.

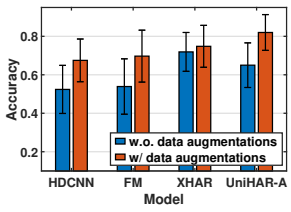


(b) Data-centralized.

Figure 9: Accuracies with different numbers of labels.



(a) Data-decentralized.



(b) Data-centralized.

Figure 10: Effect of data augmentations.

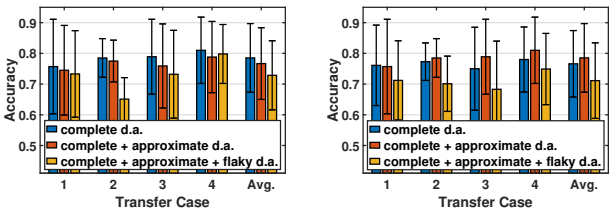
best performance for case 4, it is shown to be sensitive to the source domain dataset and falls much lower in other cases. UniHAR-A, however, achieves accuracies consistently higher than 80.0% across all cases and in average outperforms the best of the three by 10% in terms of accuracy and 15% in terms of F1-score. In summary, the results demonstrate the outstanding performance of UniHAR(-A), credited to the effective data augmentations and feature extraction.

7.3 Impact of Labeled Sample Size

We then investigate how UniHAR and the baseline models perform with different amount of labeled samples. We vary the size of labeled samples from 50 to 1,400. Figure 9 plots the average accuracies achieved in data-decentralized and data-centralized scenarios, respectively. The error bar represents the standard deviation of the accuracies over the four cross-dataset evaluation cases. The results suggest that UniHAR outperforms other models in all cases by at least 10% in accuracy (and up to 20% in the data-decentralized scenario). Since HDCNN, TPN, and LIMU-GRU are prone to overfitting to the source domain, their performances on the target domain are not very related to the number of labeled samples from the source domain. On the other hand, the models in the data-centralized scenario can achieve higher accuracies when more labeled data are employed. UniHAR-A consistently outperforms the other two models in the data-centralized scenario. The UniHAR(-A) is able to achieve average accuracies of 71.0% and 72.1% in the two scenarios, when only 50 labeled samples are used. The experiment results also suggest a more robust performance of UniHAR(-A).

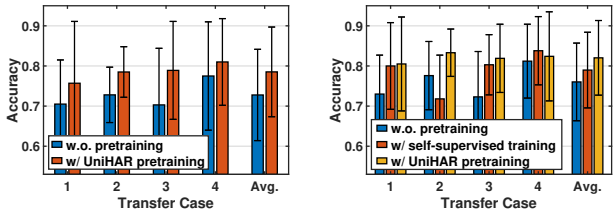
7.4 Effect of Data Augmentation

We devise an ablation study to evaluate how different data augmentation methods are effective in supporting the training objective. Figure 10 compares the performance of UniHAR and baseline models in both data-decentralized and data-centralized. It is shown that the accuracies of all models increase when integrating with the proposed data augmentation methods. Although UniHAR(-A) may lower accuracy when data augmentation is not employed, its training architecture fully exploits the potential of data augmentation with the whole learning framework and eventually outperforms the best baseline models by 10% in both scenarios.



(a) Feature extraction. (b) Activity recognition.

Figure 11: Impact of data augmentation combinations.



(a) Data-decentralized. (b) Data-centralized.

Figure 12: Accuracies of pretraining approaches.

we then examine how effective is the proposed way of integrating complete and approximate data augmentation in UniHAR. We compare the three choices of i) using only complete data augmentation, ii) using both complete and approximate data augmentation, and iii) using all including flaky data augmentations, and during both the feature extraction and activity recognition stages. Figure 11 plots the achieved accuracies in the data-decentralized scenario. Results show that if approximate data augmentation is adopted in the feature extraction stage, it may lead to an accuracy drop of 2.0% because of the approximation errors introduced to massive unlabeled data. On the other hand, applying approximate data augmentation in the activity recognition stage can further enrich data diversity and thus increase accuracy by 1.9%. Flaky data augmentation, however, only introduces a negative impact to almost all cases when applied in either stage. The UniHAR performance drops by 5.7% on average when data augmentation jittering and permutation are applied. Our results with the data-centralized scenario show similar results (omitted due to page limits).

We also investigate the detailed performance gains when different data augmentation methods are adopted. The *local rotation* introduces the largest gains, i.e., 19.4% and 15.3% accuracy improvements in the two scenarios. The *dense sampling* and *time wrapping* together improve the average accuracies by 2.6% and 2.3% in the two scenarios, respectively.

7.5 Effect of Feature Extraction

We evaluate the effectiveness of feature extraction by comparing the performance of three training approaches, i.e., without any pretraining, with self-supervised learning (only available for the data-centralized scenario), and with both self-supervised and federated learning (UniHAR pretraining). Figure 12 plots the end performance of different pertaining approaches and the results show that the UniHAR training approach consistently achieves better performance in the data-decentralized scenario. For the data-centralized scenario, although the self-supervised training slightly outperforms the UniHAR training in case 4, it only achieves an accuracy of 71.2% in case 2. It is because the encoder is sensitive to the number of samples [28], the recognizer with the biased encoder degrades significantly in case 2, where the source domain dataset UCI has the fewest samples. The experiment suggests that UniHAR(-A) can achieve more robust performance with federated training. The encoder and decoder are first initialized with the source domain dataset in the feature extraction process, which gains a 1.5% improvement in average accuracy. The possible reason is that mobile clients can better adapt a good initialized model to their local datasets [7].

Table 4: Efficiency comparison.

Model	Para.	Size	Train. Time	Infer. Time
DCNN	17 K	76 KB	4.2 ms	0.8 ms
TPN	26 K	194 KB	7.6 ms	2.2 ms
LIMU-GRU	54 K	239 KB	8.1 ms	5.5 ms
HDCNN	28 K	118 KB	3.6 ms	0.8 ms
FM	50 K	203 KB	5.8 ms	2.9 ms
XHAR	700 K	2607 KB	17.0 ms	14.2 ms
UniHAR	15 K	78 KB	8.9, 17.6 ms	3.6 ms

7.6 Model Size and Latency

Table 4 compares UniHAR with the baseline models in terms of the number of parameters, model size, training time, and inference time. The models are optimized by lite Pytorch Mobile [35]. The training time is the time the server takes to train a mini-batch (64) of samples and the inference time is the execution time for inferring one IMU sample on the Samsung Galaxy S8. The two training time of UniHAR corresponds to the models without and with the domain classifier. In summary, the model size of UniHAR is small and its training and inference time is comparable with others.

8 RELATED WORK

Wearable-based HAR systems [9, 10, 15, 17, 27, 37, 40, 64, 65] are ubiquitous and low-cost. Conventional HAR models [15, 43, 47, 51, 58, 63] adopt deep neural networks and achieve high performance with the help of sufficient well-annotated

datasets. However, IMU data heterogeneity prevents them from achieving promising performance in practice.

Recent federated learning schemes [21–23, 34, 53] allow for distributed training without accessing raw data but they require fully-labeled data at the target users, which cannot be directly applied to the considered scenario.

Self-supervised learning works [14, 36, 40, 52, 57, 62] have shown effectiveness in extracting useful features from unlabeled data and thereby improving the performance of downstream HAR models. For example, the encoder models from TPN [40] and LIMU-BERT [62] may be viewed as the early efforts in building "foundation" models to extract contextual features from unlabeled IMU data, with which task-specific models can achieve superior performances with limited labeled data. However, these models still require some labeled data to train HAR classifiers, which can be overfitted to specific domains and fail to achieve high performance for target users without any labeled data.

Unsupervised domain adaptation approaches have been introduced to HAR applications [17, 37, 65] and reduce the distribution divergence between different domains. Specifically, HDCNN [17] learns transferable features by minimizing Kullback-Leibler divergence between the source and target domains. XHAR [65] extracts domain-independent features by adversarial training. Unfortunately, our experiments demonstrate that purely learning-based domain adaptation approaches fail to handle highly heterogeneous IMU data across domains and cannot achieve satisfactory performance in adapting models across different user groups.

Prior works [49, 55] devise a range of IMU data augmentation methods, e.g., random noising, to increase label size and prevent overfitting to specific domains. And recent studies [36, 40, 52, 57] have explored self-supervised learning with data augmentation techniques for leveraging unlabeled data. However, many flaky data augmentations have been adopted in those studies [36, 40, 49, 52, 55, 57], which may generate readings that do not conform to the physical sensing principles and undermine the data distributions.

Different from existing studies, UniHAR aims at building a general HAR framework, in which a representation model is first built with massive unlabeled data, and supervised training with limited labeled data is thereafter adopted to adapt the model across user domains. UniHAR specifically explores physics-informed data augmentation that aligns with the underlying physical process and constructively embeds them into different learning stages to improve both intra-domain and inter-domain data representativeness.

9 DISCUSSION

Impact of orientation representation. The core idea of the physics-informed data augmentation is general and the existence of physical embedding is independent of representations. While quaternion is one of several possible ways to represent the device orientation, augmentation with physical embedding can be expressed with other representations. For example, we may also represent sensing models using rotation matrices: $\boldsymbol{a} = \boldsymbol{R}^{-1}(\boldsymbol{l} + \boldsymbol{g}), \boldsymbol{\omega} = f_g^{-1}(\boldsymbol{R}_{t-1}^{-1}\boldsymbol{R}_t)/\Delta t$, where $\boldsymbol{R}$ is the rotation matrix representing the device orientation and f_g^{-1} converts the rotation matrix to angular changes [48]. Taking local rotation as example, its physical embedding is $\boldsymbol{R}' = \boldsymbol{R}\Delta\boldsymbol{R}$, and augmented readings are derived:

$$\boldsymbol{a}' = (\boldsymbol{R}\Delta\boldsymbol{R})^{-1}(\boldsymbol{l} + \boldsymbol{g}) = \Delta\boldsymbol{R}^{-1}\boldsymbol{R}^{-1}(\boldsymbol{l} + \boldsymbol{g}) = \Delta\boldsymbol{R}^{-1}\boldsymbol{a}, \tag{21}$$

$$\begin{aligned}\boldsymbol{\omega}' &= f_g^{-1}((\boldsymbol{R}_{t-1}\Delta\boldsymbol{R})^{-1}\boldsymbol{R}_t\Delta\boldsymbol{R})/\Delta t = f_g^{-1}(\Delta\boldsymbol{R}^{-1}\boldsymbol{R}_{t-1}^{-1}\boldsymbol{R}_t\Delta\boldsymbol{R})/\Delta t \\ &= f_g^{-1}(\Delta\boldsymbol{R}^{-1}f_g(\boldsymbol{\omega}\Delta t)\Delta\boldsymbol{R})/\Delta t \\ &= f_g^{-1}(\Delta\boldsymbol{R}^{-1}f_g(\boldsymbol{\omega})\Delta\boldsymbol{R}) = \Delta\boldsymbol{R}^{-1}\boldsymbol{\omega}.\end{aligned} \tag{22}$$

These equations demonstrate the same results as those obtained with quaternions (Equation 11 and 12).

Frequency-Domain data augmentation. Some studies [26, 36] propose data augmentations in the frequency domain. To establish the physical embedding of these data augmentations, we may accordingly perform Fourier Transform on physical states like orientation. However, frequency domain operations can potentially violate the constraints of orientation representation after the inverse Fourier Transform. For example, a low-pass filter on orientation quaternions can lead to non-unit quaternions and a loss of their physical meaning. Further study is needed to understand their relationships with *physics-informed data augmentation.*

10 CONCLUSION

In this paper, we practically adopt HAR with realistic overhead for mobile devices. The proposed UniHAR framework effectively adopts *physics-informed data augmentation* on massive unlabeled and limited labeled IMU data to overcome the data heterogeneity across various users. UniHAR is prototyped in the mobile platform and tested introducing low overhead. Extensive evaluation with cross-dataset experiments demonstrates its outstanding performance compared with state-of-the-art approaches.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Prepositioning (IAF-PP) Funding Initiative, under its NRF Investigatorship (NRFI) NRF-NRFI08-2022-0010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. This research is also supported by Singapore MOE Tier 1 (RG88/22). Mo Li is the corresponding author.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.

[2] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* (2018).

[3] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* 116, 32 (2019), 15849–15854.

[4] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. 2020. A systematic study of unsupervised domain adaptation for robust human-activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–30.

[5] Yiqiang Chen, Jindong Wang, Meiyu Huang, and Han Yu. 2019. Cross-position activity recognition with stratified transfer learning. *Pervasive and Mobile Computing* 57 (2019), 1–13.

[6] Android Developers. [n.d.]. Profile your app performance. https://developer.android.com/studio/profile

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems* 33 (2020), 3557–3568.

[8] Siwei Feng and Marco F Duarte. 2019. Few-shot learning-based human activity recognition. *Expert Systems with Applications* 138 (2019), 112782.

[9] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. Metasense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 110–123.

[10] Andreas Grammenos, Cecilia Mascolo, and Jon Crowcroft. 2018. You are sensing, but are you biased? a user unaided sensor calibration approach for mobile sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–26.

[11] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.

[12] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. ColloSSL: Collaborative Self-Supervised Learning for Human Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–28.

[13] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. 2019. SenseHAR: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 15–28.

[14] Sijie Ji, Yaxiong Xie, and Mo Li. 2022. SiFall: Practical Online Fall Detection with RF Sensing. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 563–577.

[15] Wenchao Jiang and Zhaozheng Yin. 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*. 1307–1310.

[16] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. 2009. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 37–42.

[17] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. 2018. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 1–9.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020), 1–29.

[20] Hyeokhyen Kwon, Bingyao Wang, Gregory D Abowd, and Thomas Plötz. 2021. Approaching the real-world: Supporting activity recognition training with virtual imu data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–32.

[21] Ang Li, Jingwei Sun, Pengcheng Li, Yu Pu, Hai Li, and Yiran Chen. 2021. Hermes: an efficient federated learning framework for heterogeneous mobile clients. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 420–437.

[22] Chenglin Li, Di Niu, Bei Jiang, Xiao Zuo, and Jianming Yang. 2021. Meta-HAR: Federated Representation Learning for Human Activity Recognition. In *Proceedings of the Web Conference 2021*. 912–922.

[23] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. 2022. PyramidFL: A fine-grained client selection framework for efficient federated learning. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 158–171.

[24] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*.

[25] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S Burd. 2016. Deep learning for rfid-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 164–175.

[26] Dongxin Liu, Tianshi Wang, Shengzhong Liu, Ruijie Wang, Shuochao Yao, and Tarek Abdelzaher. 2021. Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective. In *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–10.

[27] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. GIobalFusion: A Global Attentional Deep Learning Framework for Multisensor Information Fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[28] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[29] Yang Liu, Zhenjiang Li, Zhidan Liu, and Kaishun Wu. 2019. Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 287–299.

[30] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics*. IEEE, 1–7.

[31] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. 2019. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*. 49–58.

[32] Alan Mazankiewicz, Klemens Böhm, and Mario Bergés. 2020. Incremental Real-Time Personalization in Human Activity Recognition Using Domain Adaptive Batch Normalization. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020),

1–20.

[33] Xiaomin Ouyang, Xian Shuai, Jiayu Zhou, Ivy Wang Shi, Zhiyuan Xie, Guoliang Xing, and Jianwei Huang. 2022. Cosmo: contrastive fusion learning with small data for multimodal human activity recognition. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 324–337.

[34] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. 2021. ClusterFL: a similarity-aware federated learning system for human activity recognition. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 54–66.

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[36] Hangwei Qian, Tian Tian, and Chunyan Miao. 2022. What makes good contrastive learning on small-scale wearable-based tasks?. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3761–3771.

[37] Xin Qin, Yiqiang Chen, Jindong Wang, and Chaohui Yu. 2019. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–25.

[38] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-aware human activity recognition using smartphones. *Neurocomputing* 171 (2016), 754–767.

[39] Angelo Maria Sabatini. 2011. Kalman-filter-based orientation determination using inertial/magnetic sensors: Observability analysis and performance evaluation. *Sensors* 11, 10 (2011), 9182–9206.

[40] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.

[41] Aaqib Saeed, Flora D Salim, Tanir Ozcelebi, and Johan Lukkien. 2020. Federated self-supervised learning of multisensor representations for embedded intelligence. *IEEE Internet of Things Journal* 8, 2 (2020), 1030–1040.

[42] Andrea Rosales Sanabria, Franco Zambonelli, Simon Dobson, and Juan Ye. 2021. ContrasGAN: Unsupervised domain adaptation in Human Activity Recognition via adversarial and contrastive learning. *Pervasive and Mobile Computing* (2021), 101477.

[43] Zhiyao Sheng, Huatao Xu, Qian Zhang, and Dong Wang. 2022. Facilitating Radar-Based Gesture Recognition With Self-Supervised Learning. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 154–162.

[44] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. 2014. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* 14, 6 (2014), 10146–10176.

[45] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.

[46] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data* 8 (2021), 1–34.

[47] Pekka Siirtola and Juha Röning. 2012. Recognizing human activities user-independently on smartphones based on accelerometer data. *IJIMAI* 1, 5 (2012), 38–45.

[48] Joan Sola. 2017. Quaternion kinematics for the error-state Kalman filter. *arXiv preprint arXiv:1711.02508* (2017).

[49] Odongo Steven Eyobu and Dong Seog Han. 2018. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors* 18, 9 (2018), 2892.

[50] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 127–140.

[51] Scott Sun, Dennis Melamed, and Kris Kitani. 2021. IDOL: Inertial Deep Orientation-Estimation and Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6128–6137.

[52] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. SelfHAR: Improving Human Activity Recognition through Self-training with Unlabeled Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–30.

[53] Linlin Tu, Xiaomin Ouyang, Jiayu Zhou, Yuze He, and Guoliang Xing. 2021. FedDL: Federated Learning via Dynamic Layer Sharing for Human Activity Recognition. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 15–28.

[54] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[55] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*. 216–220.

[56] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[57] Jinqiang Wang, Tao Zhu, Jingyuan Gan, Liming Luke Chen, Huansheng Ning, and Yaping Wan. 2022. Sensor Data Augmentation by Resampling in Contrastive Learning for Human Activity Recognition. *IEEE Sensors Journal* 22, 23 (2022), 22994–23008.

[58] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing* 21, 5 (2020), 1798–1811.

[59] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478* (2020).

[60] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems* 33 (2020), 6256–6268.

[61] Haifeng Xing, Jinglong Li, Bo Hou, Yongjian Zhang, and Meifeng Guo. 2017. Pedestrian stride length estimation from IMU measurements and ANN based algorithm. *Journal of Sensors* 2017 (2017).

[62] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. 2021. LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 220–233.

[63] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition.. In *Ijcai*, Vol. 15. Buenos Aires, Argentina, 3995–4001.

[64] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. 351–360.

[65] Zhijun Zhou, Yingtian Zhang, Xiaojing Yu, Panlong Yang, Xiang-Yang Li, Jing Zhao, and Hao Zhou. 2020. XHAR: Deep Domain Adaptation for Human Activity Recognition with Smart Devices. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.