Daily Physical Activity Monitoring—Adaptive Learning from Multi-source Motion Sensor Data

Anonymous Authors

Abstract

In healthcare applications, there is a growing need to develop machine learning models that use data from a single source, such as that from a wrist wearable device, to monitor physical activities, assess health risks, and provide immediate health recommendations or interventions. However, the limitation of using single-source data often compromises the model's accuracy, as it fails to capture the full scope of human activities. While a more comprehensive dataset can be gathered in a lab setting using multiple sensors attached to various body parts, this approach is not practical for everyday use due to the impracticality of wearing multiple sensors. To address this challenge, we introduce a transfer learning framework that optimizes machine learning models for everyday applications by leveraging multi-source data collected in a laboratory setting. We introduce a novel metric to leverage the inherent relationship between these multiple data sources, as they are all paired to capture aspects of the same physical activity. Through numerical experiments, our framework outperforms existing methods in classification accuracy and robustness to noise, offering a promising avenue for the enhancement of daily activity monitoring.

Data and Code Availability This paper uses *Daily and Sports Activity* dataset (Altun and Barshan, 2010; Altun et al., 2010; Barshan and Yüksek, 2014), available at https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities, and *Indoor User Movement Prediction from RSS data* (Bacciu et al., 2011; Gallicchio et al., 2012; Bacciu et al., 2013, 2014), available at https://archive.ics.uci.edu/dataset/348/indoor+user+movement+prediction+from+rss+data. Our codes are available at https://anonymous.4open.science/r/HealthTimeSerial-BF8D.

Institutional Review Board (IRB) Our research uses publicly available data, which does not require IRB approval.

1. Introduction

Background & Motivation The healthcare sector is undergoing a transformative era, fueled by the integration of artificial intelligence, data analytics, and sensor technology (Ganju et al., 2022; Lee et al., 2022a; Pillai et al., 2023). As a part of this transformation, wearable motion sensor technologies are emerging as a key driver in reshaping personal health monitoring. These devices enable continuous and non-intrusive tracking of physical activities, providing invaluable insights into individual daily routines (Liu et al., 2021). For instance, for office workers, these sensors serve as proactive health partners, encouraging individuals to cultivate healthier habits and mitigating the risks associated with sedentary lifestyles (Adjerid et al., 2022). Moreover, they offer potential life-saving benefits, such as monitoring vulnerable populations for sudden falls and triggering immediate alerts, thereby reducing the likelihood of severe injuries or complications (Chander et al., 2020).

Thanks to the increased accessibility of data and advancements in machine learning techniques, sensors can now precisely capture activities based on the collected data (Pandl et al., 2021; Xu et al., 2023; Chen et al., 2023; Matton et al., 2023). This data typically comprises location trajectory information, capturing the movements of different body parts during specific activities. For thorough data acquisition, a laboratory setting is essential where multiple wearable motion sensors can be strategically positioned on various parts of an individual's body, such as the wrists, ankles, chest, and head. Participants are then asked to execute specific actions, which serve as the labels for the corresponding data (McCarthy and Grey, 2015). Time series classification methods are then used to analyze and interpret this multi-source data, enabling the construction of prediction models (Zeng et al., 2020). This comprehensive data collection method offers a holistic view of human move-

ments, contributing to the high accuracy of the resulting models.

Challenges However, in daily applications, it is often impractical or undesirable for users to wear multiple motion sensors. Instead, they prefer a single sensor placed on a particular body part, like the left arm, thus gathering data solely from that single source. This limitation poses a dual challenge. On one hand, models that have been trained with multi-source data can be challenging to adapt when provided with the limited perspective of single-source data in daily applications. On the other hand, training classifiers solely with data from one source can compromise activity detection accuracy, as it lacks the broad spectrum of insights that multi-source data offers.

To bridge this gap, we advocate for the adoption of transfer learning techniques (Spathis et al., 2021; Merrill and Althoff, 2023), enabling the effective utilization of multi-source data in daily monitoring tasks. These techniques draw upon the knowledge from interconnected datasets to accomplish the classification task. Moreover, a distinctive characteristic of the multi-source data collected by motion sensors is its inherent pairwise structure—a facet not fully exploited by existing transfer learning methodologies. As previously mentioned, the multi-source data emerges when participants wear an assortment of sensors across various body parts, engaging in specific predefined actions. This setup ensures that time series data from these domains are acquired synchronously, all marked with the same action labels. Such cohesive data collection naturally establishes an intrinsic data pairing between different domains, offering a rich set of correlations that could potentially improve the performance of classification outcomes. On the other hand, existing transfer learning methods fall short in this aspect, processing data from each sensor independently. This oversight may miss out on potential gains in achieving a more comprehensive understanding of activities.

Proposed Approach & Contribution In this work, we introduce a novel transfer learning framework designed to harness the pairwise structure of multi-source motion sensor data. Within this framework, the sensor placed on the body part that is daily monitored acts as the target domain, while sensors on other body parts are considered as source domains. Our proposed framework includes three steps: (1) computing the domain similarities between the target domain and source domains; (2) pre-training the model on source domains based on the domains' similarities; and (3) fine-tuning the pre-trained model on the target domain. In the first step, we propose a novel metric named *Inter-domain Pairwise Distance* (IPD), which factors in the pairwise structure. We pair time series data across different domains in a manner that reflects the data collection procedure and calculate IPD through the method of smooth bootstrapping. In the second step, we train the model (classifier) on all source domains, adjusting the step size (learning rate) based on the calculated IPD. A lower IPD value indicates a closer similarity between the source and target domains, prompting the model to adopt a larger step size and focus more on learning from these similar domains. For the last step, we fine-tune the pre-trained model on the target domain, mirroring traditional transfer learning methods. The procedure of training a classifier and applying the trained classifier in daily physical activity monitoring as well as the proposed transfer learning framework is summarized in **Figure 1** and **Figure 2**.

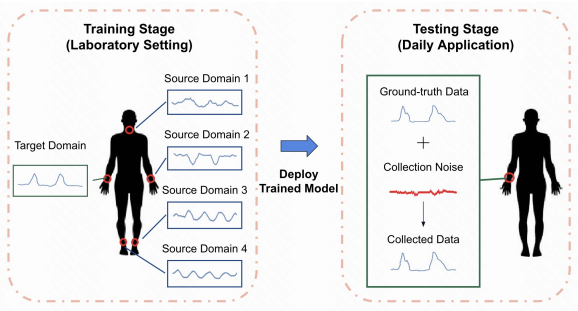


Figure 1: The procedure of training a classifier and applying the trained classifier in daily physical activity monitoring.

This work offers two main contributions to daily physical activity monitoring for healthcare applications. First, we propose a structured framework that leverages multi-source time series data collected in laboratory experiments into daily physical activity monitoring. Through our approach, laboratory-derived data is transformed, providing valuable insights applicable to real-world contexts. Secondly, we emphasize the unique pairwise structure inherent in motion sensor data, which sets it apart from many conventional transfer learning scenarios. We introduce a novel metric to reflect this structured pairing within the modeling framework. This metric has a wide applicability range, being compatible with var-

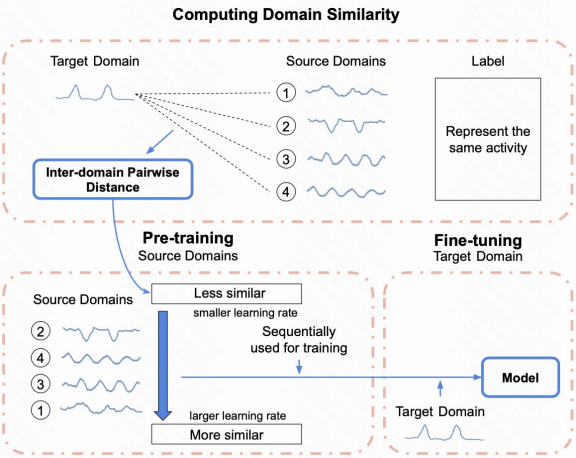


Figure 2: The illustration of the proposed transfer learning framework.

ious time series distance measures. Through rigorous testing on various datasets from the UCI Machine Learning Repository, we demonstrate that our novel framework outperforms existing state-of-the-art methods in both classification accuracy and robustness against input noise. These results mark it as a significant advancement in wearable sensor technology for healthcare applications.

2. Related Work

Physical Activity Monitoring in Healthcare Daily physical activity monitoring in healthcare has garnered significant attention in recent years due to the increasing prevalence of sedentary lifestyles and the associated risk of chronic diseases. In this context, wearable devices and smartphone-based sensors have emerged as popular tools for continuous monitoring of physical activity patterns. These devices typically employ accelerometers, gyroscopes, and heart rate monitors to track various aspects of physical activity, such as step count, energy expenditure, and activity type (Patel et al., 2015; Doherty et al., 2017; Merrill et al., 2023).

Researchers have explored the application of machine learning and deep learning algorithms to classify and predict different types of activities from sensor data (Hammerla et al., 2016; Weatherhead et al., 2022). These studies have demonstrated the potential of daily physical activity monitoring in improving the management of chronic conditions, such as diabetes, cardiovascular diseases, and obesity, by promoting patient adherence to prescribed exercise regimens and facilitating personalized treatment plans (Jakicic et al., 2016). Moreover, the integration of physical activity monitoring with telemedicine and remote patient monitoring systems has enabled healthcare providers to remotely assess patients' progress and offer timely interventions (Swan, 2012; Kvedar et al., 2014; Zheng et al., 2023).

Time Series Classification Time series classification tasks have been significantly improved with the advent of deep learning techniques. Examples of these improvements include disease diagnosis based on time series of physiological parameters, the classification of heart arrhythmias from Electrocardiograms (ECGs) (Bagnall et al., 2017), and human activity recognition (Petitjean et al., 2014). While certain deep learning architectures can achieve state-of-the-art results on large-scale data, their performance tends to be suboptimal when access to a large labeled training dataset is limited (Sutskever et al., 2014; Chen and Shi, 2021).

Owing to the challenges in collecting and annotating time-series data, researchers have increasingly opted against training large deep learning models from scratch. Instead, they employ pre-trained models on source tasks and adapt them to the target task. This transfer learning approach has yielded satisfactory model performance in a range of fields, including computer vision (Bhattacharjee et al., 2017a), natural language processing (Fawaz et al., 2018), and anomaly detection (Bhattacharjee et al., 2017b), particularly when training data is limited. Prior studies have demonstrated that transfer learning can also enhance performance in time series classification problems with limited data (Fawaz et al., 2018).

3. Problem Statement

For effective daily physical monitoring, a model (classifier) is required to interpret the data collected by the daily wearable motion sensor, like smartwatches. This model is designed to accurately predict the type of physical activity being performed. In this context, we provide the mathematical definition of the **time series data** in Definition 1, which represents the data collected by a specific motion sensor.

Definition 1 (Time series data) *Time series data is a sequence of observations collected sequentially over a time period. For a period T of*

observations, the time series data is denoted as $X = [x_1, \ldots, x_T] \in \mathbb{R}^{K \times T}$*, where each* x_t *is a* K*-dimensional vector, representing the measurement taken at the* t*-th moment in the sequence.*

From Definition 1, time series data consist of a series of observations recorded by a motion sensor. Each observation in this series is represented by a K-dimensional vector, detailing motion attributes such as acceleration, rotation, and orientation at a particular timestamp. The entirety of this time series data X represents a specific activity. This activity is categorized under the label C, which belongs to a collection of pre-determined activity types $\mathcal{C}$.

Definition 2 (Domain) *A domain* $\mathcal{D}$ *is comprised of a feature space* $\mathcal{X} \subset \mathbb{R}^{K \times T}$ *and a marginal distribution* $P(\mathcal{X})$*. Within domain* $\mathcal{D}$*, we have multiple samples of time series data* $\hat{\mathcal{D}} = \{X_1, \ldots, X_N\}$ *as a realization, where each* $X_n \in \mathcal{X}$ *denotes a time series data as defined in Definition 1.*

Definition 3 (Pairwise multi-source time series) *A multi-source time series data* $\mathfrak{X} \doteq \left\{\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \ldots, \hat{\mathcal{D}}_V\right\}$ *is defined as a collection of time series data from multiple domains, where* $\hat{\mathcal{D}}_v$ *denotes the data set associated with the domain* $\mathcal{D}_v$ *defined in Definition 2. The pairwise structure of the multi-source time series data requires (i) each set* $\hat{\mathcal{D}}_v$ *has the same number of time series data (i.e.,* $N = \left|\hat{\mathcal{D}}_1\right| = \left|\hat{\mathcal{D}}_2\right| = \ldots = \left|\hat{\mathcal{D}}_V\right|$*), and (ii) for* $n \in \{1, 2, \ldots, N\}$*, the* n*-th time series data across the domains* $\{X_{\mathcal{D}_1,n}, X_{\mathcal{D}_2,n}, \ldots, X_{\mathcal{D}_V,n}\}$ *are paired with an identical label* $C_n \in \mathcal{C}$*, where* $X_{\mathcal{D}_v,n}$ *denotes the* n*-th time series data in the domain* $\mathcal{D}_v$*.*

In a laboratory setting, sensors placed on different body parts each define a unique domain, collectively gathering multi-source time series data to reflect specific actions performed by a participant. This necessitates aligning data from V motion sensors across V domains under consistent labels to represent the executed activities. Introducing a transfer learning framework aimed at leveraging lab-acquired, multi-domain time series data to improve wearable motion sensor data classification in real-world scenarios. By treating one sensor (the target domain $\mathcal{T}$) as the primary focus and the others (source domains $\mathcal{S}_q$, $q = 1, \ldots, Q$, with $Q = V - 1$) as supplementary, the framework learns a classifier from source domain data $\hat{\mathcal{S}}_q = \left\{X_{\mathcal{S}_q,n}\right\}_{n=1}^{N}$ before training it with target domain data $\hat{\mathcal{T}} = \{X_{\mathcal{T},n}\}_{n=1}^{N}$. This process integrates knowledge from source domains to enhance the classifier's ability to accurately categorize activities based on data from daily-used wearable sensors.

4. Method

In this section, we present the adaptive transfer learning framework. Our method works as follows. In Section 4.1, we introduce a new metric Inter-domain Pairwise Distance (IPD) to quantify the similarity between domains. We then provide a practical method to approximate IPD between each source domain and the target domain using the collected time series data. In Section 4.2, we describe the procedure of pre-training the model within the source domains. The degree of knowledge transfer is reflected by adapting the learning rate based on the calculated IPD. We postpone the description of the fine-tuning procedure in the target domain to Appendix.

4.1. Domain Similarity Computation

In this section, we introduce the *Inter-domain Pairwise Distance (IPD)*, a novel metric designed to quantify the similarity between two domains wherein the time series data share a pairwise structure.

Definition 4 *Inter-domain Pairwise Distance (IPD) between two domains* $\mathcal{S}$ *and* $\mathcal{T}$ *is defined as*

$$\mathrm{IPD} = \mathbb{E}\left\{\mathrm{dist}\left(X_{\mathcal{S}}, X_{\mathcal{T}}\right)\right\},$$

where $X_{\mathcal{S}}$ *and* $X_{\mathcal{T}}$ *are paired time series data from domains* $\mathcal{S}$ *and* $\mathcal{T}$ *respectively, and* $\mathrm{dist}(\cdot, \cdot)$ *is a selected distance measure for two time series data.*

As shown in Definition 4, the IPD measures the similarity between two domains by evaluating the expected distance between their paired time series data. A higher IPD value indicates less similarity between these two domains. Notably, our metric offers both flexibility and adaptability, as it can integrate any time series distance measure, including but not limited to, Euclidean distance, Minkowski distance, and dynamic time warping. Furthermore, our approach maintains the pairwise data structure. This contrasts with most of the transfer learning literature, where the pairwise structure is neither present nor considered. Instead, such methods typically treat each time series data point as an independent sample from the distribution within the domain and measure the distance between two domains using metrics for empirical distributions, such as the Wasserstein distance.

Our experimental findings later highlight that disregarding the pairwise structure of motion sensor data and resorting to conventional domain distances can compromise classification accuracy.

It is worth noting that the calculation of IPD requires taking the expectation over domains, however in practice, we typically observe specific realizations. In the following, we describe a threefold procedure of estimating IPD given only a set of paired time series data: (i) Computation of empirical inter-domain difference. We start by calculating the empirical IPD using the available paired time series data samples. (ii) Difference density estimation. We then approximate the probability density function of the empirical IPD with the kernel density estimation (Silverman, 1986). (iii) Difference sampling and distance calculation. Finally, we generate new samples from this approximated density function and use these new samples to approximate IPD. In essence, our methodology shares similar spirits with the smooth bootstrap method (Hall et al., 1989). Compared with merely using the empirical IPD, our approach inherits the advantages from the smooth bootstrap method, including robustness against noise, improved variability, and practicability with small-size samples. We now illustrate this threefold procedure in details.

Computation of Empirical Inter-domain Difference Recall that the time series data in the source domain $\mathcal{S}_q$ is denoted as $\hat{\mathcal{S}}_q = \{X_{\mathcal{S}_q,n}\}_{n=1}^N$ for $q \in [Q]$ and the data in the target domain is $\hat{\mathcal{T}} = \{X_{\mathcal{T},n}\}_{n=1}^N$. Since both $X_{\mathcal{S}_q,n}$ and $X_{\mathcal{T},n}$ are multivariate time series data, we first decompose them into K univariate time series data as $X_{\mathcal{S}_q,n}^{(k)}$ and $X_{\mathcal{T},n}^{(k)}$ for $k \in [K]$, where $X^{(k)}$ denotes the k-th entry of the time series data X. In other words, this decomposition allows us to consider each type of movement information within the time series data separately.

For each n, we compute the pairwise univariate time series distance $D_{q,n}^{(k)} = \text{dist}\left(X_{\mathcal{S}_q,n}^{(k)}, X_{\mathcal{T},n}^{(k)}\right) \in \mathbb{R}$. We then reintegrate all the univariate distances back into the vector form and obtain the difference vector associated with the n-th pair of multivariate time series data $X_{\mathcal{S}_q,n}$ and $X_{\mathcal{T},n}$ as $D_{q,n} = \left(D_{q,n}^{(1)}, D_{q,n}^{(2)}, \ldots, D_{q,n}^{(K)}\right)^\top$. Noted that the distances calculated in this study are exclusively between a specific source domain $\mathcal{S}_q$ and the target domain $\mathcal{T}$. Therefore, we keep q in the subscript to distinguish among source domains, while we omit $\mathcal{S}$ and $\mathcal{T}$ for notational simplicity. Consequently, we use the set of differences between each pair of time series data $\mathcal{M}_q = \{D_{q,1}, D_{q,2}, \ldots, D_{q,N}\}$ to represent the empirical difference between source domain $\mathcal{S}_q$ and the target domain $\mathcal{T}$. We summarize the computation of empirical inter-domain difference in Algorithm 1.

Algorithm 1 Computation of Empirical Inter-domain Difference.

REQUIRE: Source domain data $\hat{\mathcal{S}}_q$, target domain data $\hat{\mathcal{T}}$.
for $n = 1, 2, \ldots, N$ **do**
  Select the n-th pair of multivariate time series $(X_{\mathcal{S}_q,n}, X_{\mathcal{T},n})$ in $\hat{\mathcal{S}}_q$ and $\hat{\mathcal{T}}$;
  for $k = 1, 2, \ldots, K$ **do**
    Compute univariate time series distance as $D_{q,n}^{(k)} = \text{dist}\left(X_{\mathcal{S}_q,n}^{(k)}, X_{\mathcal{T},n}^{(k)}\right) \in \mathbb{R}$;
  end for
  Construct the difference vector of the n-th pair

$$D_{q,n} = \left(D_{q,n}^{(1)}, D_{q,n}^{(2)}, \ldots, D_{q,n}^{(K)}\right)^\top \in \mathbb{R}^K;$$

end for
Return $\mathcal{M}_q = \{D_{q,1}, D_{q,2}, \ldots, D_{q,N}\}$.

We note that we opt to decompose the multivariate time series data into K univariate time series data in our approach. The reason is mainly two-fold. First, the realm of univariate time series has been extensively studied, resulting in a plethora of research on determining distances between such series. In contrast, generalizing these established distances to multivariate scenarios still remains a relatively uncharted domain. Second, it is essential to preserve the multivariate structure of domain similarity in the initial stages. By doing so, we ensure a holistic assessment of domain similarity, enriched by the pairwise structure in subsequent analytical steps. This not only furnishes a more nuanced insight but also sets a robust foundation for in-depth analyses.

Difference Density Estimation We here approximate the probability density function of the inter-domain difference between the source domain $\mathcal{S}_q$ and the target domain $\mathcal{T}$ with the attained sample set $\mathcal{M}_q = \{D_{q,1}, D_{q,2}, \ldots, D_{q,N}\}$, using kernel density estimation (Silverman, 1986):

$$\hat{Q}_q(D) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{K}_{\mathbf{H}}\left(D - D_{q,n}\right), \qquad (1)$$

where the kernel density function $\mathcal{K}_{\mathbf{H}}(D)$ is defined as $\mathcal{K}_{\mathbf{H}}(D) = |\mathbf{H}|^{-1/2}\mathcal{K}\left(\mathbf{H}^{-1/2}D\right)$. Here, $\mathbf{H}$ is a selected symmetric and positive definite $K \times K$ matrix, referred to as the bandwidth. Meanwhile, $\mathcal{K}$ is the selected kernel function. In this work, we specifically select the Gaussian kernel function as a representative. That is, $\mathcal{K}_{\mathbf{H}}(D) = (2\pi)^{-d/2}\,|\mathbf{H}|^{-1/2}\,e^{-\frac{1}{2}D^{\top}\mathbf{H}^{-1}D}$.

Inter-domain Pairwise Distance Calculation In the last step to estimate the IPD between source domain $\mathcal{S}_q$ and the target domain $\mathcal{T}$, we first generate samples from the approximated probability density function (p.d.f.) of the inter-domain difference $\hat{Q}_q(D)$. Given this p.d.f., the samples can be efficiently generated by the Monte Carlo Markov chain (MCMC) algorithm (Asmussen and Glynn, 2007). Suppose we have m generated samples from $\hat{Q}_q(D)$, denoted as $\left\{\hat{D}_{q,i}\right\}_{i=1}^{m}$. By placing all the generated samples into a matrix, we have $\hat{\boldsymbol{D}}_q = \left(\begin{array}{ccccc}\hat{D}_{q,1} & \ldots & \hat{D}_{q,i} & \ldots & \hat{D}_{q,m}\end{array}\right)^{\top} \in \mathbb{R}^{K\times m}$. This matrix $\hat{\boldsymbol{D}}_q$ from sampling contains the difference information between the source domain $\mathcal{S}_q$ and the target domain $\mathcal{T}$. Consequently, we approximate IPD_q with the norm of the matrix $\hat{\boldsymbol{D}}_q$ as

$$\widehat{\mathrm{IPD}}_q \doteq g_q = \frac{1}{m}\left\|\hat{\boldsymbol{D}}_q\right\|.$$

For each source domain, the approximated IPD is attained following this procedure. We represent the IPD vector between all source domains and the target domain as $\boldsymbol{g} = (g_1, g_2, \ldots, g_Q)^{\top} \in \mathbb{R}^Q$.

4.2. Pre-training in Source Domains

In this section, we present the process of pre-training the model (classifier) in the source domains and describe how the estimated Inter-domain Pairwise Distance (IPD) guides the pre-training process. Our framework has two key aspects: (1) Unified model framework. We utilize a singular classifier model, which is sequentially trained and updated across all source domains. (2) Model Flexibility. The framework is inherently flexible, accommodating a diverse range of models including Long Short-Term Memory (LSTM), encoders, and others, without being constrained to a specific model type. We denote the selected model as

$$\boldsymbol{f}(X;\boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{\theta}$ denotes model parameters to be learned. The model takes a single-source time series data $X \in \mathbb{R}^{K\times T}$ as the input and outputs a label $C \in \mathcal{C} = \{c_1, c_2, \ldots, c_L\}$ for classification.

A pivotal aspect of our method is the utilization of the IPD vector $\boldsymbol{g}$, which adaptively adjusts the learning rate of the model within each source domain. Specifically, we increase the learning rate for a source domain with a smaller IPD to enable the model to better incorporate information from that domain. This is because a larger learning rate results in a higher degree of knowledge transfer, as the model assimilates more information about the current source domain over the same number of learning epochs. Thus, the derived IPD serves as a proxy for the similarity between the source and target domains, guiding the model to transfer knowledge from the most relevant source domains. Regarding the sequence in which the source domains are processed during the pre-training process, we sort and renumber all the source domains in descending order based on the associated IPD g_q. Thus, the pre-trained model is updated by and relies more on the data in the source domains that are more similar to the target domain.

In terms of the training process, we employ the gradient descent method to minimize the loss function. We sequentially perform the gradient descent steps on all the source domains $\{\mathcal{S}_q\}_{q=1}^{Q}$, maintaining a consistent initial learning rate, λ^0, and total learning epochs, J, across all these domains. After completing the J learning epochs on source domain $\mathcal{S}_q$, the parameters acquired are used as the initial starting point for the subsequent source domain $\mathcal{S}_{q+1}$, continuing the training the model in domain $\mathcal{S}_{q+1}$.

The j-th learning epoch on source domain $\mathcal{S}_q$ can be represented by

$$\boldsymbol{\theta}_q^{j+1} = \boldsymbol{\theta}_q^{j} - \lambda_q^j \nabla_{\boldsymbol{\theta}} \mathcal{J}_{\mathcal{S}_q}\left(\boldsymbol{\theta}_q^j\right), \tag{3}$$

where

$$\mathcal{J}_{\mathcal{S}_q}\left(\boldsymbol{\theta}_q^j\right) = \mathbb{E}_{\left\{\left(X_{\mathcal{S}_q,n}, C_n\right)\right\}_{n=1}^{N}} \mathcal{L}\left(\boldsymbol{\theta}_q^j\right) \tag{4}$$

is the empirical loss function. $\mathcal{L}$ denotes the categorical cross-entropy loss function, which is commonly used for classification problems (Murphy, 2012). Here, $\boldsymbol{\theta}_q^j$ is the learned parameter of the model (e.g., the weight parameters of a neural network) in the j-th learning epoch and λ_q^j is the corresponding learning rate. Inspired by the adaptive learning rate decay for sequential training on domains (Mirzadeh et al., 2020), our adaptive transfer learning framework updates the learning rate λ_q^j as

$$\lambda_q^{j+1} = \lambda_q^j \cdot (1 - \alpha_q), \tag{5}$$

where α_q is the weight of source domain $\mathcal{S}_q$ relative to all source domains. It is normalized by the sum of all Q importance values of all source domains as

$$\alpha_q = \frac{g_q}{\sum_{l=1}^{Q} g_l}. \tag{6}$$

In this manner, we quantitatively determine the degree of knowledge transfer from each source domain to the target domain. This ensures that the greater the similarity between a source domain and the target domain (indicated by a smaller α_q), the more knowledge is transferred from that source domain (achieved through a larger learning rate).

We postpone the description of fine-tuning the model in the target domain to Appendix, and summarize our framework in Algorithm 2. Once the learning procedure ends, the learned model parameter $\boldsymbol{\theta}_{\mathcal{T}}^{j}$ is used to represent the trained classifier $\boldsymbol{f}\left(X;\boldsymbol{\theta}_{\mathcal{T}}^{j}\right)$ as in (2). Consequently, when the daily wearable motion sensor collects new time-series data X^*, the trained classifier is then employed to classify the physical activity with $\boldsymbol{f}\left(X^*;\boldsymbol{\theta}_{\mathcal{T}}^{j}\right)$, which leverages the information provided by multi-source time series.

5. Experiments

In this section, we conduct numerical experiments to demonstrate the efficacy of our proposed framework. Utilizing time series data collected from motion sensors, we sought to discern and interpret various physical activities, aligning with the daily physical activity monitoring for healthcare applications.

In terms of the dataset, we select the UCI Daily and Sports Activity (DSA) dataset (Altun and Barshan, 2010; Altun et al., 2010; Barshan and Yüksek, 2014). This dataset contains motion sensor data of 19 daily and sports activities carried out by 8 subjects. In particular, participants performed instructed activities while 5 sensors (domains) were placed on the torso, right arm, left arm, right leg, and left leg during data collection. Each sensor captures time series data as a $K = 9$ dimensional vector with a length of $T = 125$. Each activity comprises 480 time series recordings, summing up to a total of $N = 480 \times 19$ time series data per domain. In our experiments, we randomly choose data from 6 out of 8 subjects as the training set in each repetition, with the data from the remaining subjects reserved for validation. For data processing, min-max rescaling is applied to each time series dimension, ensuring that all values are

Algorithm 2 Adaptive Learning from Multi-source Motion Sensor Data.

REQUIRE: Source domain data $\hat{\mathcal{S}}_q$, target domain data $\hat{\mathcal{T}}$, sequence of collected labels $\{C_n\}_{n=1}^{N}$, initial learning rate λ^0, number of learning epochs in each source domain J, number of learning epochs in target domain J_{target}, initial value of the model parameter $\boldsymbol{\theta}^0$, number of partitions $\boldsymbol{k}$, baseline learning rate in target domain $\lambda_{\mathcal{T}}$ and number of the maximum consecutive degeneration R.
// Domain Similarity Computation
for $q = 1 \ldots, Q$ **do**
  Call **Algorithm** 1 to get the sample set of difference vectors $\mathcal{M}_q$;
  Approximate p.d.f. of the difference vector to get $\hat{Q}_q(D)$ as in Eq. (1);
  Generate $\hat{\boldsymbol{D}}_q = \left(\hat{D}_{q,1} \ \ldots \ \hat{D}_{q,m} \right)^\top \in \mathbb{R}^{K \times m}$ from $\hat{Q}_q(D)$;
  Calculate matrix norm $g_q = \frac{1}{m} \left\| \hat{\boldsymbol{D}}_q \right\| \in \mathbb{R}$;
end for
// Pre-training in Source Domains
Sort the source domains such that $g_1 \geqslant g_2 \geqslant \ldots g_Q$ and set $\boldsymbol{\theta}_0^0 = \boldsymbol{\theta}^0$;
for $q = 1, 2 \ldots, Q$ **do**
  Compute the weight of each source domain by Eq. (6)
  Set $\boldsymbol{\theta}_q^0 = \boldsymbol{\theta}_{q-1}^J$ and $\lambda_q^0 = \lambda^0$;
  for $j = 0, 1, 2, \ldots, J$ **do**
    Update the parameter $\boldsymbol{\theta}_q^{j+1}$ via Eq. (3)-(5)
  end for
end for
// Fine-tuning in Target Domain
Set $\boldsymbol{\theta}_{\mathcal{T}}^0 = \boldsymbol{\theta}_Q^J$, $\lambda_{\mathcal{T}}^0 = \lambda_{\mathcal{T}}$, $r = 0$ and $j = 0$;
Randomly partition $\hat{\mathcal{T}}$ as $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{\boldsymbol{k}}\}$;
while $r < R$ and $j \leqslant J_{\text{target}}$ **do**
  Randomly select $\mathcal{B}_j \in \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{\boldsymbol{k}}\}$;
  Compute the learning rate $\lambda_{\mathcal{T}}^j$ by Eq. (9);
  if $\lambda_{\mathcal{T}}^j > \lambda_{\mathcal{T}}^{j-1}$ **then**
    Set $r = r + 1$;
  else
    Set $r = 0$;
  end if
  Update the parameter $\boldsymbol{\theta}_{\mathcal{T}}^{j+1}$ via Eq. (7)-(8) and set $j = j + 1$;
end while
Return $\boldsymbol{\theta}_{\mathcal{T}}^{j}$ as the fine-tuned model.

confined within the range of $[-1, 1]$. This normalization step serves to neutralize the impact of disparities in scale and range between dimensions, thereby enhancing the convergence of stochastic gradient descent during the training phase of the classifiers.

We include several baseline approaches in the experiments as follows: (1) *No Transfer*: This approach does not utilize the time series data in source domains and directly fine-tunes the model in the target domain. (2) *Direct Transfer*: This does not calculate the domain similarities and sets the equal learning rate across all source domains. (3) *No pairing*: This approach pre-trains the model in source domains with the approximated domain distance to the target domain, while the distance between two domains does not take the pairwise structure of the data into consideration. (4) *Freezing*: The freezing method keeps specific layers or weights of a pre-trained model unchanged during the fine-tuning process, allowing the target domain data to update only the unfrozen layers or weights. (5) *Convolutional deep Domain Adaptation model for Time Series data (CoDATS)*: This method applies domain adaptation techniques to align feature distributions (Wilson et al., 2020). In addition, we also employ different categories of models as the classifier in the experiments, including (1) *Long short-term memory networks (LSTM)* (Hochreiter and Schmidhuber, 1997); (2) *Encoder* (Serrà et al., 2018); (3) *residual neural network (ResNet)* (Wang et al., 2017); and (4) *Time series attentional prototype network (TapNet)* (Zhang et al., 2020).

5.1. General Evaluation

In this section, we first present the experimental results on classification accuracy, which is quantified using the ratio of correct classification (RCC):

$$\text{RCC} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{I}\left\{ \boldsymbol{f}\left(X_i; \hat{\boldsymbol{\theta}}\right) = C_i \right\},$$

where N_{test} denotes the size of the validation set, X_i represents one time series data in the validation set and C_i is the associated label, and $\boldsymbol{f}\left(X; \hat{\boldsymbol{\theta}}\right)$ is the trained classifier. Essentially, RCC measures the alignment between the classifier's output label and the actual ground-truth label across the validation dataset. The entire process is repeated for $I = 15$ times. We report both the mean and the standard deviation of RCC across these 15 repetitions.

To align with real application scenarios, such as using wearable sensors to detect falls in vulnerable populations (Kavuncuoğlu et al., 2022; Turan and Barshan, 2022; Koşar and Barshan, 2023), our experiments focus on binary classification. In each set of experiments, we first randomly select a label as the positive, and the remaining labels are all regarded as the negative. Then the classifier is trained to decide whether the time series is associated with the positive label. Regarding the training, we bootstrap the positive samples (upsampling) so that they have the same number of negative samples. When testing the trained model, we randomly select negative samples (downsampling) to ensure a balance. The experimental results for the dynamic time warping (DTW) metric are included in Table 1, where the principal number indicates the mean RCC, and the value within parentheses represents the standard deviation. The findings from our results offer several key insights: Our proposed transfer learning framework with DTW metric consistently achieves the highest classification accuracy when using LSTM, Encoder and TapNet classifiers. With the ResNet classifier, its performance is ranked second and comparable to the best. In addition, *No Transfer* obtains the least classification accuracy, underlining the integral role of transfer learning with multiple sources in boosting classification performance. While *Direct Transfer* and *No Pairing* do improve the performance, they lag considerably behind our proposed method. This underscores the potential of the inherent structure of data, suggesting that disregarding it can dilute the quality of results. The state-to-art CoDATS method achieves slightly worse results for all classifiers, which further validates the efficiency of our method. Among various transfer learning technologies and classifiers, our framework with the TapNet model achieves the best performance. We include additional numerical experiments in Appendix, which also demonstrates the priority of our framework.

5.2. In-depth Evaluation

In this section, we delve deeper into the performance evaluation of our proposed approach. Specifically, we first compare the results with different time series distance metric. Then we conduct experiments to evaluate whether the order of source domains in the pre-training phase affects the performance of our approach. Lastly, we impose noise to the time series data when testing the algorithms.

Table 1: Accuracy of different algorithms with DTW metric on DSA dataset.

Algorithm	LSTM	Encoder	ResNet	TapNet
DTW-Paired (ours)	**.9722(±.0104)**	**.9655(±.0126)**	.9524(±.0155)	**.9726(±.0122)**
No Transfer	.8451(±.0267)	.7632(±.0062)	.6164(±.0204)	.7352(±.0114)
Direct Transfer	.8729(±.0109)	.8856(±.0134)	.9255(±.0134)	.8331(±.0374)
No pairing	.9184(±.0214)	.9265(±.0124)	.9310(±.0102)	.8977(±.0212)
Freezing	-	.9112(±.0137)	**.9655(±.0032)**	.9271(±.0134)
CoDATS	.9392(±.0054)	.9627(±.0185)	.9292(±.0157)	.9622(±.0153)

5.2.1. Sensitivity on Distance Metric

In the previous experiment, we select the DTW distance as a representative of the time series distance metric. We further include the experimental results with another two distances: (1) Euclidean distance and (2) the Bag-of-SFA Symbols (BOSS) algorithm (Schäfer, 2015). We present the experimental results in Table 2. The results indicate that the adaptive transfer learning approach with DTW consistently outperforms other metrics. This can be attributed to the capability of DTW to manage non-linear alignment between time series. It excels at capturing similarities even when patterns in the data have different rates of progression or occur in different phases. However, the experiments with the Euclidean distance deliver satisfactory results, meanwhile calculating the Euclidean distance is more efficient than DTW.

5.2.2. Order of Source Domains in the Pre-training Phase

Recall that, in the pre-training procedure of our proposed framework, we sort and renumber all the source domains in descending order based on the associated IPD. We also conduct experiments where the order of the source domains is randomly determined. We present the results in Table 3. We have the following insights. First, when we employ a sorted order for the source domains in Algorithm 2, there is a noticeable improvement in the RCC compared to a random order. This enhancement can be attributed to the process wherein the pre-trained model, in a sorted order, is predominantly updated and influenced by data from source domains that align more closely with the target domain. Second, when adaptive transfer learning uses a random order for source domains, it tends to display a higher standard deviation in RCC. This is because a random order brings inherent unpredictability during the pre-training phase, leading to more uncertainty. This inconsistency carries through, affecting the performance of the trained model. Lastly, despite the challenges posed by randomness, the adaptive transfer learning framework manages to deliver satisfactory classification results even with a random order of source domains owing to the learning rate tailored for each source domain. Since this rate factors in the similarity between the source and target domain, the overall learning process remains relatively stable to the specific sequence of source domains during pre-training.

5.2.3. Noise Injection

In real applications, time series collected by wearable motion sensors are frequently susceptible to noise from the data collection process. This contrasts with data from controlled laboratory settings, which often offer cleaner readings (Rubin-Falcone et al., 2023). As such, the trained classifier using the laboratory data is supposed to be robust against the noise in the input dataset. Therefore, to replicate real-world conditions more accurately, we introduce synthetic Gaussian noise $\mathcal{N}(\mathbf{0}, 0.02\,\mathrm{Diag}\{x_t\})$ to some of the input time series data X. Our objective is to evaluate the robustness of the trained classifier against the presence of input noise in the context of wearable motion sensor data. As shown in Figure 3, we present the results using both TapNet and LSTM classifiers, factoring in different ratios of the timestamps that are injected with noise, which indicates the robustness of our proposed transfer learning framework.

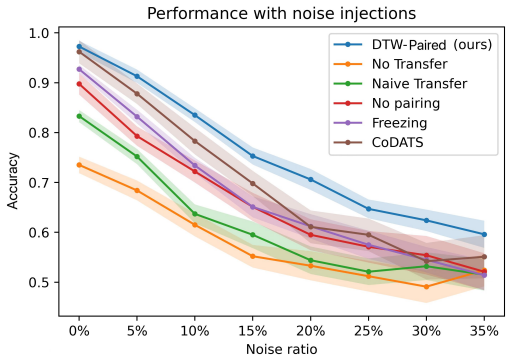
The experimental results in Figure 3 indicate that although the input data noise affects all transfer learning methods, our framework achieves the highest accuracy across different noise ratios. In conclusion, our proposed approach achieves robustness against the noise in the input time series data, which is attributed to the utilization of the smooth bootstrap method when calculating the proposed metric that quantifies the domain similarities.

Table 2: Accuracy of adaptive transfer learning with different time series distance metrics on DSA dataset.

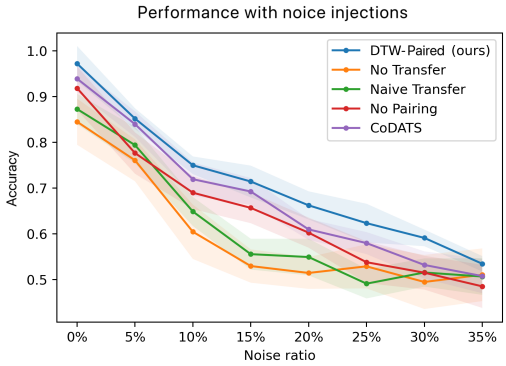
Distance metric	LSTM	Encoder	ResNet	TapNet
DTW	**.9722**(±.**0104**)	**.9655**(±.**0126**)	**.9524**(±.**0104**)	**.9726**(±.**0122**)
Euclidean	.9268(±.0034)	.9288(±.0206)	.9254(±.0204)	.9432(±.0206)
BOSS	.9310(±.0116)	.9492(±.0105)	.9492(±.0105)	.9421(±.0221)

Table 3: Accuracy of adaptive transfer learning with different orders on DSA dataset.

Order	LSTM	Encoder	ResNet	TapNet
Sorted	**.9722**(±.**0104**)	**.9655**(±.**0126**)	**.9524**(±.**0104**)	**.9726**(±.**0122**)
Random	.9465(±.0315)	.9232(±.0115)	.9155(±.0434)	.9552(±.0458)



(*a*) TapNet



(*b*) LSTM

Figure 3: RCC of time series classification approaches across different ratios of noise. The standard deviation of RCC is indicated by the shadow along the line.

6. Discussion

We develop a transfer learning framework for multi-source time series classification, aiming to enhance daily physical monitoring in healthcare applications (e.g., rehabilitation) using data from multiple motion sensors in professional laboratories. We propose a metric to quantify domain similarities that account for the pairwise structure of time series. The framework then pre-trains a classifier on source domains and fine-tunes it on the target domain, where the degree of knowledge transfer is determined by the proposed metric. Our experimental results show the superiority of our approach, achieving higher classification accuracy and robustness against input data noise compared to existing methods. Thus, the application of transfer learning enhances daily physical monitoring compared to traditional methods that are based on single-source time series data, offering improved health monitoring, diagnosis, and intervention.

Regarding future work, our framework goes beyond daily physical activity monitoring, demonstrating adaptability to a broader range of healthcare assessments such as heart and lung function (McDermott et al., 2021). Our framework compensates for the limited scope of data collected by wearable ECG monitors and simple at-home spirometry devices compared to their professional counterparts (Raghu et al., 2022; Spathis et al., 2021). To facilitate this, future enhancements focus on accommodating diverse data forms, including images and audio, within healthcare applications (Xu et al., 2023; Ho et al., 2021). By adapting our transfer learning framework for multi-modal data integration and developing methodologies for cross-domain comparisons (Liu and Lin, 2023), we aim to facilitate more comprehensive and accurate patient care. This approach includes exploring advanced feature extraction technologies and new similarity metrics, paving the way for a more versatile and effective application of the framework in leveraging varied healthcare data.

References

Idris Adjerid, George Loewenstein, Rachael Purta, and Aaron Striegel. Gain-loss incentives and physical activity: the role of choice and wearable health tools. *Management Science*, 68(4):2642–2667, 2022.

Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. In *Human Behavior Understanding: First International Workshop, HBU 2010, Istanbul, Turkey, August 22, 2010. Proceedings 1*, pages 38–51. Springer, 2010.

Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.

Søren Asmussen and Peter W Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer, 2007.

Davide Bacciu, Claudio Gallicchio, Alessio Micheli, Stefano Chessa, and Paolo Barsocchi. Predicting user movements in heterogeneous indoor environments by reservoir computing. In *Proc. of the IJCAI Workshop on Space, Time and Ambient Intelligence (STAMI), Barcellona, Spain*, pages 1–6. Citeseer, 2011.

Davide Bacciu, Stefano Chessa, Claudio Gallicchio, Alessio Micheli, and Paolo Barsocchi. An experimental evaluation of reservoir computation for ambient assisted living. In *Neural Nets and Surroundings: 22nd Italian Workshop on Neural Nets, WIRN 2012, May 17-19, Vietri sul Mare, Salerno, Italy*, pages 41–50. Springer, 2013.

Davide Bacciu, Paolo Barsocchi, Stefano Chessa, Claudio Gallicchio, and Alessio Micheli. An experimental characterization of reservoir computing in ambient assisted living applications. *Neural Computing and Applications*, 24:1451–1464, 2014.

Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31:606–660, 2017.

Billur Barshan and Murat Cihan Yüksek. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, 57(11):1649–1667, 2014.

Sreyasee Das Bhattacharjee, Bala Venkatram Balantrapu, William Tolone, and Ashit Talukder. Identifying extremism in social media with multi-view context-aware subset optimization. In *2017 IEEE international conference on big data (big data)*, pages 3638–3647. IEEE, 2017a.

Sreyasee Das Bhattacharjee, Junsong Yuan, Zhang Jiaqi, and Yap-Peng Tan. Context-aware graph-based analysis for detecting anomalous activities. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1021–1026. IEEE, 2017b.

Harish Chander, Reuben F Burch, Purva Talegaonkar, David Saucier, Tony Luczak, John E Ball, Alana Turner, Sachini NK Kodithuwakku Arachchige, Will Carroll, Brian K Smith, et al. Wearable stretch sensors for human movement monitoring and fall detection in ergonomics. *International journal of environmental research and public health*, 17(10):3554, 2020.

Wei Chen and Ke Shi. Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, 136:126–140, 2021. ISSN 0893-6080.

Zhi Chen, Sarah Tan, Urszula Chajewska, Cynthia Rudin, and Rich Caruna. Missing values and imputation in healthcare data: Can interpretable machine learning help? In *Conference on Health, Inference, and Learning*, pages 86–99. PMLR, 2023.

Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christoper G Owen, et al. Large scale population assessment of physical activity using wrist worn accelerometers: the uk biobank study. *PloS one*, 12 (2):e0169649, 2017.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In *2018 IEEE international conference on big data (Big Data)*, pages 1367–1376. IEEE, 2018.

Claudio Gallicchio, Alessio Micheli, Paolo Barsocchi, and Stefano Chessa. User movements forecasting

by reservoir computing using signal streams produced by mote-class sensors. In *Mobile Lightweight Wireless Systems: Third International ICST Conference, MOBILIGHT 2011, Bilbao, Spain, May 9-10, 2011, Revised Selected Papers 3*, pages 151–168. Springer, 2012.

Kartik K Ganju, Hilal Atasoy, and Paul A Pavlou. Do electronic health record systems increase medicare reimbursements? the moderating effect of the recovery audit program. *Management Science*, 68 (4):2889–2913, 2022.

Peter Hall, Thomas J DiCiccio, and Joseph P Romano. On smoothing and the bootstrap. *The Annals of Statistics*, pages 692–704, 1989.

Nils Y Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

Danliang Ho, Iain Bee Huat Tan, and Mehul Motani. Predictive models for colorectal cancer recurrence using multi-modal healthcare data. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 204–213, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

John M Jakicic, Kelliann K Davis, Renee J Rogers, Wendy C King, Marsha D Marcus, Diane Helsel, Amy D Rickman, Abdus S Wahed, and Steven H Belle. Effect of wearable technology combined with a lifestyle intervention on long-term weight loss: the idea randomized clinical trial. *Jama*, 316(11): 1161–1171, 2016.

Erhan Kavuncuoğlu, Esma Uzunhisarcıklı, Billur Barshan, and Ahmet Turan Özdemir. Investigating the performance of wearable motion sensors on recognizing falls and daily activities via machine learning. *Digital Signal Processing*, 126:103365, 2022.

Enes Koşar and Billur Barshan. A new cnn-lstm architecture for activity recognition employing wearable motion sensor data: Enabling diverse feature extraction. *Engineering Applications of Artificial Intelligence*, 124:106529, 2023.

Joseph Kvedar, Molly Joel Coye, and Wendy Everett. Connected health: a review of technologies and strategies to improve patient care with telemedicine and telehealth. *Health affairs*, 33(2): 194–199, 2014.

Kwanhyung Lee, Hyewon Jeong, Seyun Kim, Donghwa Yang, Hoon-Chul Kang, and Edward Choi. Real-time seizure detection using eeg: a comprehensive comparison of recent approaches under a realistic setting. *arXiv preprint arXiv:2201.08780*, 2022a.

Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Out-of-distribution robustness via disagreement. In *The Eleventh International Conference on Learning Representations*, 2022b.

Shuo Shuo Liu and Lin Lin. Adaptive weighted multiview clustering. In *Conference on Health, Inference, and Learning*, pages 19–36. PMLR, 2023.

Xin Liu, Chen Zhao, Bin Zheng, Qinwei Guo, Xiaoqin Duan, Aziguli Wulamu, and Dezheng Zhang. Wearable devices for gait analysis in intelligent healthcare. *Frontiers in Computer Science*, 3: 661676, 2021.

Wenao Ma, Cheng Chen, Shuang Zheng, Jing Qin, Huimao Zhang, and Qi Dou. Test-time adaptation with calibration of medical image classification nets for label distribution shift. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 313–323. Springer, 2022.

Katie Matton, Robert Lewis, John Guttag, and Rosalind Picard. Contrastive learning of electrodermal activity representations for stress detection. In *Conference on Health, Inference, and Learning*, pages 410–426. PMLR, 2023.

Margaret McCarthy and Margaret Grey. Motion sensor use for physical activity data: methodological considerations. *Nursing research*, 64(4):320, 2015.

Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive ehr timeseries pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 257–278, 2021.

Mika A Merrill and Tim Althoff. Self-supervised pretraining and transfer learning enable\titlebreak flu and covid-19 predictions in small mobile sensing

datasets. In *Conference on Health, Inference, and Learning*, pages 191–206. PMLR, 2023.

Mike A Merrill, Esteban Safranchik, Arinbjörn Kolbeinsson, Piyusha Gade, Ernesto Ramirez, Ludwig Schmidt, Luca Foshchini, and Tim Althoff. Homekit2020: A benchmark for time series classification on a large mobile sensing dataset with laboratory tested ground truth of influenza infections. In *Conference on Health, Inference, and Learning*, pages 207–228. PMLR, 2023.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Konstantin D Pandl, Fabian Feiland, Scott Thiebes, and Ali Sunyaev. Trustworthy machine learning for health care: scalable data valuation with the shapley value. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 47–57, 2021.

Mitesh S Patel, David A Asch, and Kevin G Volpp. Wearable devices as facilitators, not drivers, of health behavior change. *Jama*, 313(5):459–460, 2015.

François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE international conference on data mining*, pages 470–479. IEEE, 2014.

Arvind Pillai, Subigya Nepal, and Andrew Campbell. Rare life event detection via mobile sensing using multi-task learning. In *Conference on Health, Inference, and Learning*, pages 279–293. PMLR, 2023.

Aniruddh Raghu, Divya Shanmugam, Eugene Pomerantsev, John Guttag, and Collin M Stultz. Data augmentation for electrocardiograms. In *Conference on Health, Inference, and Learning*, pages 282–310. PMLR, 2022.

Harry Rubin-Falcone, Joyce M Lee, and Jenna Wiens. Denoising autoencoders for learning from noisy patient-reported data. In *Conference on Health, Inference, and Learning*, pages 393–409. PMLR, 2023.

Patrick Schäfer. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29:1505–1530, 2015.

J Serrà, S Pascual, and A Karatzoglou. Towards a universal neural network encoder for time series. *Artif Intell Res Dev Curr Chall New Trends Appl*, 308:120, 2018.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Dimitris Spathis, Ignacio Perez-Pozuelo, Soren Brage, Nicholas J Wareham, and Cecilia Mascolo. Self-supervised transfer learning of physiological representations from free-living wearable data. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 69–78, 2021.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Melanie Swan. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator networks*, 1(3):217–253, 2012.

Mustafa Şahin Turan and Billur Barshan. Classification of fall directions via wearable motion sensors. *Digital Signal Processing*, 125:103129, 2022.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

Addison Weatherhead, Robert Greer, Michael-Alice Moga, Mjaye Mazwi, Danny Eytan, Anna Goldenberg, and Sana Tonekaboni. Learning unsupervised representations for icu timeseries. In *Conference on Health, Inference, and Learning*, pages 152–168. PMLR, 2022.

Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1768–1778, 2020.

Li Xu, Bo Liu, Ameer Hamza Khan, Lu Fan, and Xiao-Ming Wu. Multi-modal pre-training for medical vision-language understanding and generation: An empirical study with a new benchmark. In *Conference on Health, Inference, and Learning*, pages 117–133. PMLR, 2023.

Zhengxin Zeng, Moeness G Amin, and Tao Shan. Arm motion classification using time-series analysis of the spectrogram frequency envelopes. *Remote Sensing*, 12(3):454, 2020.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6845–6852, 2020.

Jessica Zheng, Hanrui Wang, Anand Chandrasekhar, Aaron D Aguirre, Song Han, Hae-Seung Lee, and Charles G Sodini. Machine learning for arterial blood pressure prediction. In *Conference on Health, Inference, and Learning*, pages 427–439. PMLR, 2023.

Appendix A. Fine-tuning in Target Domain

After completing the pre-training procedure, we obtain the pre-trained model parameter $\boldsymbol{\theta}_Q^J$ from all source domains. The last step of the adaptive transfer learning involves fine-tuning the model based on using the target domain data $\hat{\mathcal{T}} = \{X_{\mathcal{T},n}\}_{n=1}^N$ and associated labels $\{C_n\}_{n=1}^N$. The classifier's parameters are initialized with the pre-trained model parameter, denoted as $\boldsymbol{\theta}_{\mathcal{T}}^0 = \boldsymbol{\theta}_Q^J$. For fine-tuning in the target domain, we use the mini-batch gradient descent method combined with $\boldsymbol{k}$-fold cross-validation. The target domain dataset $\hat{\mathcal{T}}$ is randomly partitioned into $\boldsymbol{k}$ equal-sized, disjoint subsets as $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{\boldsymbol{k}}\}$. During each learning epoch, we randomly select a subset $\mathcal{B}_i, i \in \boldsymbol{k}$ with equal probability as the validation set, while the remaining $\boldsymbol{k}-1$ subsets are used as the training set. In this way, the model parameter $\boldsymbol{\theta}_{\mathcal{T}}^j$ is updated in the j-th learning epoch as

$$\boldsymbol{\theta}_{\mathcal{T}}^{j+1} = \boldsymbol{\theta}_{\mathcal{T}}^j - \lambda_{\mathcal{T}}^j \nabla_{\boldsymbol{\theta}} \mathcal{J}_{\mathcal{T}} \left(\boldsymbol{\theta}_{\mathcal{T}}^j\right), \tag{7}$$

Here, $\mathcal{J}_{\mathcal{T}} \left(\boldsymbol{\theta}_{\mathcal{T}}^j\right)$ is the loss function given by

$$\mathcal{J}_{\mathcal{T}} \left(\boldsymbol{\theta}_{\mathcal{T}}^j\right) = \mathbb{E}_{\{\hat{\mathcal{T}} \setminus \mathcal{B}_j\} \cup \mathcal{C}_j} \mathcal{L} \left(\boldsymbol{\theta}_{\mathcal{T}}^j\right). \tag{8}$$

where $\{\hat{\mathcal{T}} \setminus \mathcal{B}_j\}$ is the training set in the j-th epoch and $\mathcal{C}_j$ denotes the set of labels associated with the training set. The learning rate in the j-th epoch is determined as follows:

$$\lambda_{\mathcal{T}}^j = \left(1 - \mathbb{E}_{\mathcal{B}_j \cup \mathcal{C}_j'} \mathcal{L} \left(\boldsymbol{\theta}_{\mathcal{T}}^j\right)\right) \lambda_{\mathcal{T}}, \tag{9}$$

where $\mathcal{B}_j$ is the selected validation set, $\mathcal{C}_j'$ is the associated set of labels and $\lambda_{\mathcal{T}}$ is the prescribed baseline learning rate in the target domain. The performance of the current model is evaluated with the validation set, and the learning rate for the next epoch is adjusted accordingly, decreasing if the current model exhibits good performance. The learning procedure stops if there is performance degeneration on the validation set for R consecutive epochs or if the learning epoch has been repeated for J_{target} times. Here R and J_{target} are prescribed hyperparameters. The complete procedure of the proposed transfer learning with multi-source time series data is summarized in Algorithm 2.

A.1. Domain Shift

In addition to fine-tuning the classifier with the target domain data, we also include a discussion on the

additional domain shift here. That is, although the target domain aligns with the wearable motion sensor in daily use regarding the body part, the data collected in laboratory settings are under a more controllable environment. In this way, there is an additional domain shift between the distribution of the target domain data and that of daily collected data.

In the experiments presented in the main text, we, therefore, consider imposing additional noise to the time series when testing the trained classifier. This procedure imitates the real environment with more noise. Experimental results indicate that our transfer learning framework is more robust to the noise compared with the baseline approaches, owing to the calculation of g_q.

Moreover, our framework can manage broader domain shifts through domain adaptation techniques if more labeled time series data is gathered in daily use. Test Time Adaptation (TTA) is one feasible approach, allowing the model to adapt during the testing phase to bridge the gap between the distributions of the training and testing data. We here summarize a general procedure of implementing TTA that is applicable to our framework:

1. **Identifying Domain Shift:** Before adjusting the model (classifier), the distribution shift is required to be identified. Methods include:
   - **Statistical Tests:** Apply statistical tests (e.g., Chi-square test) to compare the distribution of features in the training data and the newly collected data.
   - **Performance Metrics Monitoring:** Monitor the performance of the classifier on the new data. A significant drop in performance metrics (accuracy, precision, recall) might be attributed to a domain shift.
2. **Data Processing:** If the distribution shift is identified, the classifier is then required to be adjusted. It should be guaranteed that the new data are processed and normalized in the same way as the training data so that the classifier can learn from the new data.
3. **Model Updating:** The model is updated by fine-tuning with new data. Specifically, the learning rate of fine-tuning is suggested to be small and decrease, ensuring a conservative updating.
4. **Rolling Validation:** Implement a rolling validation procedure where the classifier is periodically validated using a recent subset of the data. This continuous validation facilitates ongoing observation of the adaptation process.
5. **Continuous Monitoring and Adaptation:** Continuously monitor the model's performance on new data. When there is a domain shift, iterate previous steps to adjust the classifier.

We refer to Wang et al. (2020); Ma et al. (2022); Lee et al. (2022b) for detailed procedures of TTA. Since the open-access datasets used in our experiments are collected in a stationary manner and do not exhibit domain shifts, we do not include TTA in our experiments. We defer the specific integration of TTA with our framework to future work.

Appendix B. Experiment Details

In the experiments, regarding our proposed adaptive transfer learning framework described in Algorithm 2, we set:

1. The initial learning rate $\lambda^0 = 5 \times 10^{-4}$;
2. The number of learning epochs in each source domain $J = 50$;
3. The number of learning epochs in the target domain $J_{\text{target}} = 100$;
4. The value of the model parameter is initiated by randomly sampling each weight parameter of the neural network from Uniform$[0, 1]$ and each bias parameter as 0;
5. The number of partitions $\boldsymbol{k} = 10$;
6. The baseline learning rate in the target domain $\lambda_{\mathcal{T}} = 1 \times 10^{-3}$;
7. The number of the maximum consecutive degeneration $R = 5$.

In terms of the neural network models employed as the classifiers, we consider (1) Long short-term memory networks (LSTM); (2) Encoder; (3) Residual neural network (ResNet); and (4) Time series attentional prototype network (TapNet). We briefly describe the employed models as follows.

1. LSTM uses three gates to control the information flow of a sequence of data, which can capture the hidden patterns of input sequences (Hochreiter and Schmidhuber, 1997). The exact implementation follows https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html?highlight=lstm#torch.nn.LSTM. We also note that the freezing technology for transfer learning is not applicable to the LSTM models due to their sequential nature, which makes learned features highly interconnected and task-specific. LSTMs also have limited depth, reducing the chances of hierarchical feature learning that enables transfer learning with layer freezing in deep neural networks. Instead, alternative transfer learning strategies are more suitable for LSTM models.

2. Encoder applies deep neural networks compressing the raw input sequences into a low-dimensional representation, and make predictions and classifications directly based on the encoded variable. We refer to Serrà et al. (2018) for reference and the implementation of Encoders can be found in https://github.com/sktime/sktime-dl/blob/master/sktime_dl/regression/_encoder.py.

3. ResNet introduces the residual connection in neural networks, which can avoids gradient vanishing and information loss in learning the pattern of a sequences. We refer to Wang et al. (2017) for reference and https://github.com/sktime/sktime-dl/blob/master/sktime_dl/regression/_resnet.py for implementation.

4. TapNet applies temporal attention mechanism to learn the importance of different timesteps of a sequence. Exact implementation follows Zhang et al. (2020) and https://www.sktime.net/en/latest/api_reference/auto_generated/sktime.classification.deep_learning.TapNetClassifier.html?highlight=tapnet.

All the experiments were run by Python 3.8 and Pytorch on a server with two 32-Core AMD Ryzen Threadripper PRO 3975WX processors and three NVIDIA RTX A6000 GPUs.

Appendix C. Additional Experiments

We present additional numerical experiments in this Section.

C.1. Influential Source Domains

In this section, we present the two most influential source domains for each target domain in **Table 4**, based on the DSA dataset. Given a specific target domain, the most influential domains are determined by the calculated g_q. Specifically, the smaller g_q is, the more similarities there are between the two domains, indicating that the source domain is more influential.

In addition, we also consider pre-training the time series classifiers using the two most influential source domains, instead of across all source domains. We present the numerical results in **Table 5**. Compared to pre-training across all source domains, excluding the least influential source domains leads to a decrease in classification accuracy but also slightly reduces model uncertainty, as indicated by the standard deviation of the classification accuracy. Moreover, focusing pre-training on the most influential source domains results in time savings during training. Thus, whether to select the most influential domains for pre-training depends on the requirements of the applications. We also note that adaptive transfer learning from the two most influential source domains using our framework still outperforms existing algorithms.

C.2. Additional Experiments on RSS Data Set

In this section, we present the experimental results based on another data set that contains time series data collected by multiple motion sensors. Specifically, we select the data set " Indoor User Movement Prediction from RSS data Data Set". The data set can be used for a binary classification task consisting of predicting the pattern of user movements from time series generated by a Wireless Sensor Network (WSN). Input data contains temporal streams of radio signal strength (RSS) measured between the nodes of a WSN, comprising 5 sensors. For the given dataset, RSS signals have been re-scaled to the interval $[-1, 1]$, singly on the set of traces collected from each anchor. Target data consists of a class label indicating whether the user's trajectory will lead to a change in the spatial context (i.e. a room change) or not.

Table 4: The two most influential source domains for each target domain.

Target Domain	Most Influential Source Domain	2nd Most Influential Source Domain
Torso	Left Leg	Right Leg
Right Arm	Left Arm	Right Leg
Left Arm	Right Arm	Left Leg
Left Leg	Right Leg	Left Arm
Right Leg	Left Leg	Right Arm

Table 5: Accuracy of different source domain selections on DSA dataset.

Source Domain Selection	LSTM	Encoder	ResNet	TapNet
All Source Domains	**.9722**(±**.0104**)	**.9655**(±**.0126**)	**.9524**(±.0155)	**.9726**(±**.0122**)
Two Most Influential Source Domains	.9533(±.0087)	.8911(±.0092)	.9124(±.0118)	.9324(±.0096)

The experimental settings are the same as the experiments on the DSA data set and the experimental results are included in Table 6. The results provide insights as follows. First, in this set of experiments, our proposed adaptive transfer learning framework achieves the best performance across different selections of classifiers. Second, directly fine-tuning the classifier in the target domain without transfer learning from source domains achieves acceptable results. Consequently, knowledge from source domains without appropriate methodology does not always enhance the classification performance, which is different from the experimental results in Section 5. This may be attributed to the fact that the task here is a binary classification problem, which is simpler, and therefore, the target domain data provides enough information to facilitate the task. Lastly, our proposed approach with the utilization of TapNet achieves the best classification performance across different transfer learning methodologies and employed classifiers, which is the same as the experimental results in Section 5.

Table 6: Accuracy of different algorithms with DTW metric on dataset 'Indoor User Movement Prediction from RSS'.

Algorithm	LSTM	Encoder	ResNet	TapNet
DTW-Paired (ours)	**.9722(±.0075)**	**.9865(±.086)**	**.9923(±.0044)**	**.9926(±.0004)**
No Transfer	.9138(±.0312)	.8742(±.0072)	.9704(±.0109)	.9694(±.0052)
Direct Transfer	.9428(±.0230)	.9256(±.094)	.9255(±.0134)	.9744(±.0134)
No pairing	.9514(±.0064)	.9566(±.084)	.9310(±.0102)	.9585(±.0262)
Freezing	-	.9612(±.0137)	.9245(±.0032)	.9474(±.0064)
CoDATS	.9673(±.0071)	.9797(±.0078)	.9899(±.0102)	.9612(±.0182)