Invariant Feature Learning for Sensor-Based Human Activity Recognition

Yujiao Hao, Rong Zheng, *Senior Member, IEEE*, and Boyu Wang

Abstract—Wearable sensor-based human activity recognition (HAR) has been a research focus in the field of ubiquitous and mobile computing for years. In recent years, many deep models have been applied to HAR problems. However, deep learning methods typically require a large amount of data for models to generalize well. Significant variances caused by different participants or diverse sensor devices limit the direct application of a pre-trained model to a subject or device that has not been seen before. To address these problems, we present an invariant feature learning framework (IFLF) that extracts common information shared across subjects and devices. IFLF incorporates two learning paradigms: 1) meta-learning to capture robust features across seen domains and adapt to an unseen one with similarity-based data selection; 2) multi-task learning to deal with data shortage and enhance overall performance via knowledge sharing among different subjects. Experiments demonstrated that IFLF is effective in handling both subject and device diversion across popular open datasets and an in-house dataset. It outperforms a baseline model of up to 40 percent in test accuracy.

Index Terms—Wearable sensor, human activity recognition, neural network, meta-learning, multi-task learning

1 Introduction

Human activity recognition (HAR) is the foundation to realize remote health services and in-home mobility monitoring. Although deep learning has seen many successes in this field, training deep models often requires a large amount of sensory data that is not always available [1]. For research ethics compliance, it often takes months to design study protocols, recruit volunteers and collect customized sensory datasets. At the same time, public inertial measurement unit (IMU) sensor datasets on HAR are typically collected by different groups of researchers following different experiment protocols, making them difficult to be used by others. The significant variability among human subjects and device types in data collection limits the direct reuse of data as well. Deep learning methods have poor generalization ability when testing data (target domain) differs from training data (source domain) due to device and subject heterogeneities (generally known as the *domain shift* problem). Fig. 1 shows the effects of cross-domain data variability. Fig. 1a visualizes features from subjects that are seen to the deep model for HAR (left) and as held-out data to the model (right), respectively. The features are well clustered when subjects are seen to the model and are inseparable for the unseen subject even though she performs the same group of activities wearing the same sensor at the same location. Fig. 1b demonstrates the effect of device diversity. The data is collected when a person performs several activities with devices A and B attached to the same on-body locations. A deep learning model is trained with device A's data. We find that despite its high inference accuracy on the testing data from the same device, the accuracy drops drastically on device B's data.

In addressing the aforementioned domain shift problems, a pooling task model (PTM) that mixes data from different domains (e.g., subjects, devices) will have low discriminative power as it ignores the dissimilarity among the domains. On the other hand, a model trained solely on data from a specific domain requires a lot of training data as it fails to take the advantage of the similarity among different sources. Since collecting and labeling sensory data with sufficient diversity is time-consuming, it is impractical to train a task-specific model for each new subject or device encountered from scratch. A few previous works have investigated domain shifts caused by device and subject diversity in HAR. In [2] and [3], the problem is formulated as domain adaptation between a pair of participants or devices. However, in practice, HAR is rarely limited to transferring knowledge between a pair of domains, rather from a group of source domains (e.g., subjects, placements, or devices) to a target domain. Furthermore, unsupervised domain adaptation approaches trade-off their performance with data labeling efforts. For example, in [2], the authors report an F1-score ≤ 0.8 in testing compared to 0.92 from [4] with supervised learning on the Opportunity dataset [5].

It is expected that despite their differences, sensory data of the same activity from different subjects or devices intrinsically share common characteristics. A common assumption is the existence of shared representations among source and target domains. When tasks sampled from source domains can 'cover' the representation space, linear predictors built upon feature extractor for each task is sufficient for good prediction [6], [7]. Based on this assumption, we

- *Yujiao Hao and Rong Zheng are with the Department of Computing and Software, McMaster University, Hamilton, ON L8S 4L8, Canada. E-mail: {haoy21, rzheng}@mcmaster.ca.*
- *Boyu Wang is with the Department of Computer Science, Western University, London, ON N6A 3K7, Canada. E-mail: bwang@csd.uwo.ca.*

Manuscript received 24 June 2020; revised 8 Feb. 2021; accepted 1 Mar. 2021. Date of publication 8 Mar. 2021; date of current version 3 Oct. 2022. (Corresponding author: Yujiao Hao.)
Digital Object Identifier no. 10.1109/TMC.2021.3064252

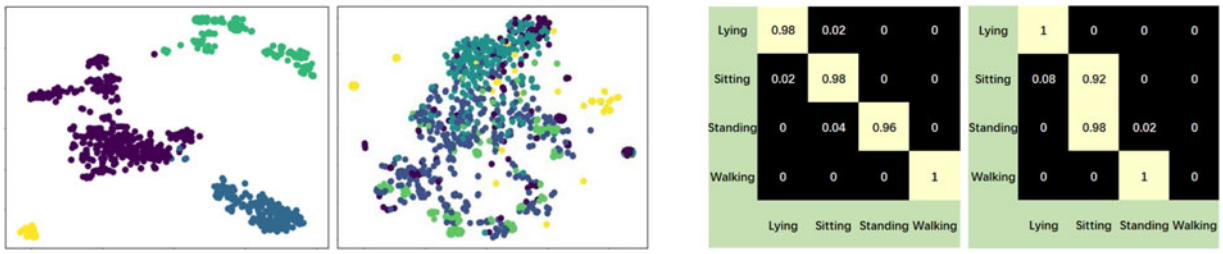


Fig. 1. Typical data variance problem caused by subject difference and device diversity. (a) depicts the impact of subject difference, where different colors in the t-SNE plots denote different activities. (b) shows the impact of device diversity. Left confusion matrix shows predictions on device seen to the model while right confusion matrix from a new unseen one. The prediction on unseen device's data is totally confused except for the 'lying' activity.

formulate the domain shift problem as a meta-learning problem with the aim to learn invariant features. By extracting features shared across domains and build task-specific layers for different source domains, the trained meta-model has better generalizability and can be adapted to a new target domain with few labeled data (a.k.a *fast adaptation*). To further reduce the amount of labeled data, we devise a metric to qualify the similarity of activities from different domains. Such a metric allows us to selectively collect new labeled data for activities exhibit high domain shifts. We have evaluated IFLF using multiple public HAR datasets and one in-house datasets collected from older adults. Extensive experiments demonstrate that a test accuracy $\geq 90\%$ can be achieved when only 10 seconds of sensory data per activity class is available from a target domain.

The main contributions of this work are:

- We present a deep learning framework that can handle various sources of domain shifts by extracting domain invariant features across multiple source domains.
- IFLF alleviates data shortage through feature sharing from multiple source domains.
- The proposed method achieves superior performance in extensive experiments over multiple datasets than the state-of-the-art meta-learning approach.
- A similarity metric is proposed to further reduce the amount of labeled data needed from a target domain for model adaptation.

The rest of paper is organized as follows. In Section 2 we discuss related work. Section 3 introduces the proposed invariant feature learning framework for HAR. We present the evaluation results on publicly available datasets and our own dataset in Section 4. Finally, Section 5 concludes the paper and lists future directions of study.

2 Related Work

In this section, we first give an overview of HAR models. Next, we discuss two categories of approaches that address variations among different domains, namely, 1) domain adaptation and 2) domain-invariant feature learning.

2.1 Human Activity Recognition Models

Before the tide of deep learning, one popular approach to solve HAR problems is extracting a set of handcrafted features based on domain knowledge and training a shallow machine learning model [8], [9], [10]. In [10], two types of features are utilized, namely, time domain features (Mean, variance or standard deviation, energy, entropy, correlation between axes, signal magnitude area, tilt angle, and autoregressive coefficients) and frequency domain features (fast Fourier transform and discrete cosine transform coefficients). Accuracy of 99 and 92 percent is reported with a support vector machine model and a one-layer neural network model, respectively, built upon the features when classifying 16 activities. However, the effectiveness of handcrafted features can be highly activity specific.

With deep learning, features can be learned from data automatically. A convolutional neural network is usually incorporated as part of the feature extractor. Deep models are reported to achieve state-of-the-art results on many popular open datasets [4], [11], [12], [13]. However, deep learning has its own limitations. It requires a large amount of data to train and is sensitive to domain shifts. For example, the t-distributed stochastic neighbor embedding (t-SNE) visualization [14] of 2D features, a form of non-linear embedding for high-dimension data, in Fig. 1a was originally 128 dimensions extracted by DeepConvLSTM [4]. Its predictive accuracy drops dramatically when applied to unseen subjects and devices. DeepSense [13] is a neural network architecture that is robust to domain shifts by merging local interactions of different sensory modalities into global interactions. However, it requires multi-sensor modalities, long data windows to achieve good performance, and is sensitive to class imbalance, making it unsuitable for transient or highly dynamic activities.

It should be noted that the proposed framework is model agnostic. In other words, we can incorporate any state-of-the-art deep learning architecture for HAR including DeepSense as the invariant feature extractor.

2.2 Domain Adaptation

As a sub-category of transfer learning approaches, domain adaptation mitigates the problem when the training data used to learn a model has a different distribution from the data on which the model is applied [15]. Differed by the information available for the target task, domain adaptation approaches can be further divided into supervised [16], [17], [18], semi-supervised[19] and unsupervised domain adaptation [2], [20], [21], [22].

Previous work on sensor-based HAR mostly falls in the category of unsupervised domain adaptation. Three types

of domain shifts have been considered, namely, subject difference, device diversity and sensor location divergence. In [2], Soleimani and Nazerfard focus on mitigating subject differences when abundant unlabeled data is available in the target domain. A generative adversarial neural network (GAN) based solution is proposed to generate shared feature representation across a pair of source and target domains. Though not targeting HAR, the work [20] is relevant to mitigate domain shifts due to device diversity. It utilizes a cycle-consistent generative adversarial network (CycleGAN) to transform target domain data to a source domain, and then apply a classifier trained on the source domain. In [21], Akbari and Jafari propose a deep generative model to transfer knowledge between a labeled source sensor and an unlabeled target. A mechanism to annotate pseudo labels for a target sensor was proposed by majority voting and intra-class correlation in [22]. It is based on handcrafted features and a SVM model. Despite the popularity of unsupervised approaches, they trade-off the ability of learning invariant features that are robust to new domain with data labeling efforts. Thus, the performance tends to be noticeably inferior to supervised approaches. Also, the problem setup is limited to transfer knowledge between a pair of source and target domains which is limiting in practice.

2.3 Domain-Invariant Feature Learning

Learning invariant features across different domains can facilitate a better generalization of a deep learning model. Specifically, meta-learning (a.k.a learning to learn [23], [24]) is one approach to achieve this goal. Model-agnostic meta-learning (MAML) [25] introduces an episodic training paradigm with gradient-based parameter updating. It inspired meta-learning based HAR approaches in [26], [27], [28]. The first two target computer vision tasks, while MetaSense[28] is designed for sensor-based HAR and thus is the most relevant to our work. In MetaSense, Gong *et al*. proposed to sample tasks both randomly within each source domain and across source domains. It achieves good performance with few-shot learning tests, but one limitation is the task sampling method requires each source domain to have the same number of classes. This assumption does not always hold especially when the activity set involves difficult or intensive motions. Both IFLF and MetaSense are meta-learning approaches but operate under distinctive assumptions. IFLF assumes that the source domains and the target domain share invariant features. In contrast, MAML and its variants such as MetaSense assume the existence of weights that are only a few gradient steps away from the optimal ones in every domain. These assumptions lead to different ways of updating models with data from the target domain: MAML and its variants update parameters of the whole model while IFLF only updates the task-specific layers.

Multi-task learning [29], [30] also helps to extract invariant features by learning the knowledge shared by different tasks (or domains). In [31], the authors propose a personalized shallow model for HAR, with a test accuracy between 63.9 and 72.8 percent on different experiment settings. It also considers a subject-level similarity as transfer factor that controls model parameter update in a gradient step. The work in [12], [32] adopt deep learning methods. In [12], Peng *et al*. handle simple and complex activities with different task-specific layers on top of invariant features across them; whereas self-supervised learning is utilized through training an invariant feature extractor that is capable of extracting features behind various signal distortions in [32]. There are 8 types of manually added signal distortions (random noise, scaled, rotated, negated, horizontally flipped, permuted, time-warped, and channel-shuffled) involved, but limited by the types of predefined signal distortions, it reaches an overall accuracy between 75.55 and 88.55 percent on 6 open datasets.

To the best of our knowledge, ours is the first work to comprehensively deal with the domain shifts arising from multiple sources and data shortage problem. It differs from previous meta-learning approaches in that instead of updating all parameters of a meta-model in a gradient step, it trains a model in an alternating optimization manner [33] to separate the task-specific and domain-invariant knowledge. In IFLF, a small amount of labeled data is required in meta-test step. Comparing to unsupervised domain adaptation approaches, doing so results in better classification accuracy at low costs and the ability to handle missing classes in source or target domains.

3 Method

Let the input and label spaces be $\mathcal{X}$ and $\mathcal{Y}$, respectively. The target domain and the set of source domains are $\mathcal{D}_{tgt} = \{(x_n, y_n)\}_{n=1}^{M}$ and $\mathcal{D}_{src} = \{D_1, D_2, \ldots, D_K\}$, respectively. $\mathcal{D}_{tgt}$ and $\mathcal{D}_{src}$ follow different distributions on the joint space $\mathcal{X} \times \mathcal{Y}$. A *domain* $D_k = \{(x_n^{(k)}, y_n^{(k)})\}_{n=1}^{N_k}$ corresponds to a source of variation, e.g., a subject or a device, where N_k is the number of labeled data samples. In HAR, each *task* is a multi-class classification problem that predicts the activity being performed from data sampled from the respective domain. The problem of meta-learning aims to learn well-generalized features from multiple source domains, and adapts the trained model to the target domain with small amount of labeled data. Since we assume the existence of domain-invariant features across the source and target domains, only the domain specific layers of the model need to be updated when applying to the target domain.

In this section, we present the detail of the proposed method. First, we illustrate the overall idea in Section 3.1. Second, the detail of invariant feature learning will be introduced in Section 3.2. Third, we explain in Section 3.3 the similarity-based fast adaptation.

3.1 Overview

The intuition behind IFLF is to learn two types of knowledge from multiple source domains: the shared features that can boost the generalization of a machine learning model, and the task-specific knowledge that provides the discriminative power within a specific domain. This is intrinsically reasonable for HAR problems: the task variations caused by different subjects or devices can be captured by task-specific parameters of IFLF. On the other hand, the signals of the same activity also have commonality, which can be embedded in an invariant feature representation that is shared across tasks. More importantly, such invariant

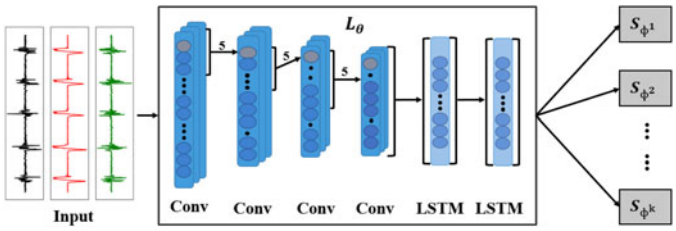


Fig. 2. An example of IFLF model which employs 4 convolutional neural network (CNN) layers and 2 long short-term memory (LSTM) layers as L_θ, and K softmax layers as S_{ϕ^k} each corresponding to a domain in $\mathcal{D}_{src}$.

features can also be transferred to a new HAR task to build a reliable model with very few data. To model the domain invariant features and task-specific ones respectively, IFLF is built upon a multi-task learning strategy where a task is associated with one of the source domains.

There are two key components in the proposed learning framework: 1) a feature extractor $L_\theta : \mathcal{X} \rightarrow \mathcal{Z}$, where $\mathcal{Z}$ is the feature space and θ denotes the parameters of L, and 2) a group of task-specific networks: $S_{\phi^k} : \mathcal{Z} \rightarrow \mathbb{R}^C$, where k denotes the kth task or domain, ϕ^k are the parameters of kth task-specific layer S^k, and C is the number of classes in $\mathcal{Y}$. Accordingly, the loss function is also composed of a feature extraction objective ℓ_L and a task-specific objective ℓ_{S^k}, which will be detailed in Section 3.2. The output of a task-specific network is given by $\hat{y} = softmax(S_{\phi^k}(L_\theta(x)))$, where $softmax(z)_j = e^{z_j} / \sum_{k=1}^{C} e^{z_k}$, for $j = 1, \ldots, C$. IFLF is model-agnostic as L_θ and S_{ϕ^k} can be any reasonable neural network. An example neural network architecture of IFLF is shown in Fig. 2.

In the training step, an IFLF model is meta-trained on $\mathcal{D}_{src}$ to obtain parameters θ and ϕ. During testing in a target domain, the trained feature extractor network L_θ will be directly reused, while a new task-specific network need to be trained with task-specific data from $\mathcal{D}_{tgt}$. Algorithm 1 depicts the overall training process of IFLF, with learning rates hyperparameters α, β. The algorithm optimizes θ and ϕ in an alternating way.

Algorithm 1. Invariant Feature Learning for Domain Adaptation

Input: Source domains $\mathcal{D}_{src} = \{D_k\}_{k=1}^{K}$, hyperparameters α, β
Output: IFLF model with parameter θ and ϕ

```
Random initialize θ, φ = {φ^1, φ^2, ..., φ^K}
repeat
    Sample tasks T = {T_1, T_2, ..., T_K} over D_src;
    //Update φ^k with fixed θ:
    for k is 1 to K do
        Freeze parameters of φ except φ^k;
        φ^k ← φ^k − β∇_{φ^k} ℓ_{S^k}(T_k, θ; φ^k);
    end for
    //Update θ with fixed φ;
    θ ← θ − α∇_θ ℓ_L(T, φ; θ);
until convergence
```

3.2 Invariant Feature Learning

To learn invariant features across source domains, one needs to consider three key factors: training strategy, feature extractor objective, and task-specific objective.

Alternating Training. If an IFLF model is trained by simply iterating among tasks sampled from D_1 to D_K, catastrophic forgetting may occur [34], namely, a model forgets previously learned tasks, and can only works properly on newly learned tasks. To avoid catastrophic forgetting, we adopt the alternating training strategy from [33], to update L_θ and S_{ϕ^k} separately. In each training epoch, we first freeze the parameters of the feature extractor layers, and update parameters of each task-specific layer with its respective data; then, we freeze parameters of the task-specific layers, and update the invariant feature extractor using all data from the previous step.

Feature Extractor. By the merit of multi-task learning, L_θ is designed to generalize well across domains through the sharing of representations between related tasks [35]. Despite its model-agnostic nature, we adopt a simple architecture described in [4] that is shown to be effective for HAR (See Fig. 2). The network includes four convolutional neural network (CNN) layers and two long short-term memory (LSTM) layers. Because the application of convolution operator depends on the input dimension, we use a 1D kernel to convolve with one-dimensional temporal sequence (a.k.a the sensor signal) [36]. 1D temporal CNNs are widely used in the area of sensor-based HAR (see [37] for a detailed survey), the combination of CNN and LSTM is beneficial for acquiring contextural knowledge and extracting meaningful features for time series data.

The objective function ℓ_L works on multiple source domains to learn a domain invariant feature representation that clusters the features by their labels. It is defined as follows:

$$\ell_L = \sum_{k=1}^{K} loss_L(T_k, \phi; \theta), \tag{1}$$

where $loss_L$ is a loss function calculated on each T_k with given θ and ϕ. Two types of loss functions are employed in this work. The first one is simply a categorical cross-entropy loss, defined as $loss_L = -\sum_{i=1}^{C} y_i log(\hat{y}_i)$ on data from each task k. An IFLF model that uses cross-entropy in the loss term in Equation (1) is called a *basic multi-task learning model* (BMTL).

In light of encouraging features to be locally clustered according to class regardless of the domain, we introduce the second type of loss function which utilizes triplet loss [38]. To calculate the triplet loss, one needs to sample m triplets from raw data in T_k, and a triplet is $t = (x_a, x_p, x_n)$. The corresponding output of a triplet in the feature space $\mathcal{Z}$ is $L_\theta(t) = (z_a, z_p, z_n)$, where x_a denotes the anchor sample, x_p is the positive sample from the same class as x_a, and x_n is a negative sample from class other than x_a. Here, the objective is to maximize the distance between (z_a, z_n) and minimize the distance between (z_a, z_p). Since it is difficult to determine a fixed threshold in a high dimensional space that separate data points into groups that are sufficiently close (and thus belong to the same class), a triplet loss is suitable for learning features that maximizes inter-class distances while minimizing intra-class distances. We compute the triplet loss as

$$loss_L(T_k, \phi; \theta) = \sum_{i=1}^{m} \max\{0, \left\| z_a^i - z_p^i \right\|^2 - \left\| z_a^i - z_n^i \right\|^2 + \epsilon\}, \tag{2}$$

TABLE 1
Summary of Datasets Used in Our Experiment

Dataset	Sampling rate	#Sensors	Placement	#Activities	#Subjects	Missing classes	Balanced	Length (in mins)
PAMAP2	100 Hz	1	dominant side's ankle	8	8	Yes	No	59.67
USCHAD	100 Hz	1	right hip	10	14	No	Yes	29.54
WISDM	20 Hz	1	pant pocket	7	51	Yes	Yes	53.59
MobilityAI	50 Hz	4	waist	4	25	Yes	No	15.29

These datasets are selected based on the diversity of participants and popularity in the HAR research field. The number of activities listed here are locomotion only, and missing classes denotes the number of activity classes per subject may vary. Length of trials is reported as an average on subject. Further details on data pre-processing and information of each dataset is discussed in Section 4.1.

where ϵ is a margin enforced between positive and negative pairs [39]. An IFLF model with a triplet loss is called *triplet multi-task learning model* (TMTL). Similar to BMTL, the loss function of TMTL is also calculated on each individual task. We then take the summation of losses over all source domains as the final objective function (1).

Task-Specific Networks. Under the assumption that if the shared feature generalizes well across all source domains, it will work on the target domain as well, L_θ should be capable of exploring the entire latent space $\mathcal{Z}$ and extracting domain invariant feature. At the same time, a task-specific network S_ϕ^k should be simple to save the labor for fast adaptation, and be sparse to take only a subset (selected feature columns) from $\mathcal{Z}$ as its inputs. A lightweight architecture of a task-specific layer S_{ϕ^k} includes a fully connected layer with a softmax activation function. The task-specific objective function is defined as the sum of a categorical cross-entropy loss and an ℓ_1-norm regularization term as follows:

$$\ell_{S^k} = -\sum_{i=1}^{C} y_i^{(k)} log\left(\hat{y}_i^{(k)}\right) + \mu|\phi^k|_1, \tag{3}$$

where μ is a hyperparameter to control the sparsity. The regularization term imposes sparsity on the task-specific layers and helps mitigate overfitting. During meta-test, we can adapt the trained model to $\mathcal{D}_{tgt}$ by either initiating a new task-specific layer from scratch or updating the parameters of a selected S_{ϕ^k}. An observation is that when features extracted by L_θ are well-clustered, we can randomly select one ϕ^k to conduct fast adaptation without much variance on the performance.

3.3 Similarity-Based Fast Adaptation

Aiming at further reducing the amount of labeled data required from the target domain for fast adaptation, a metric helps to identify the similarity or dissimilarity of motion patterns is required. We assume that if similar patterns are observed on an activity among all source domains, it is highly likely that the same activity in the target domain follows the same pattern as well. To quantify the similarity of two sensor signals, we propose a similarity metric in Equation (4), which is calculated between data from the same activity across all source domains

$$similarity_{i,j}^c = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{cov(p^i, p^j)}{\sigma_{p^i}\sigma_{p^j}}, \tag{4}$$

where σ is standard deviation, $(p^i, p^j) = \text{DTW}(x_i, x_j)$ is the pair of warped signals from sensor readings x_i and x_j from subject i and j, c is the activity class and $cov(\cdot,\cdot)$ is the covariance. Dynamic time warping (DTW) [40] calculates the best match between two temporal sequences, which may vary in speed. Here we use it to align raw sensory readings to mitigate time shifts and speed divergence. The Pearson correlation coefficient calculated on a pair of warped signals in Equation (4) provides a normalized similarity score that measures if activity c is performed similarly between a pair of participants. Data needs to be pre-processed (e.g., interpolated, noise filtered and normalized) before feeding to DTW. To eliminate the impact of misaligned sensor axis, we use the magnitude per sensor (e.g., an accelerometer or a gyroscope) as input to the similarity calculation.

Consider measurements from two sensors attached to two subjects (Subject 1 and Subject 2) performing the same activity. If the warped distance of the resulting measurements is small, this implies that the movement patterns are similar between the two subjects for the activity (despite possible differences in pace). Therefore, we can safely substitute Subject 1's data with that of Subject 2 and vice versa. After an IFLF model is trained, we no longer need to obtain labeled data from every class in $\mathcal{D}_{tgt}$. For activities that are considered similar across all source domains, we simply sample from the corresponding activity data in the source domains. These samples together with labeled data for remaining classes from the target domain are then used in updating the parameters in the task-specific layers while keeping the feature extraction layers unchanged during gradient descent. In the experiments, we find that a threshold ≥ 0.8 is suitable for distinguishing whether an activity is performed similarly among different subjects using the similarity measure defined in Equation (4).

4 Performance Evaluation

As our research mainly focuses on assessing the mobility status of older adults with IMU sensory data, we choose to conduct the experiments on open datasets and our own dataset on locomotion or lower limb exercises. During data collection, sensors are mainly attached to the trunk or lower limbs of participants. Nevertheless, the method proposed in this work is generic and can be applied to other types of activities and sensor placements.

4.1 Datasets

We consider three publicly available datasets to cover a wide variety of device types, data collection protocols, and activity classes to be recognized, and one in-house dataset

that contains measurement data from multiple IMU sensors of different vendors on older patients. Important aspects of the datasets are summarized in Table 1 with brief descriptions listed below.

1) *PAMAP2.* The Physical Activity Monitoring version 2 (PAMAP2) [41] is a dataset collected from one dominant ankle sensor (accelerometer and gyroscope) for 8 different activities, i.e., lying, sitting, walking, running, cycling, nordic walking, ascending stairs and descending stairs. Eight participants performed these activities freely without time constraints and have the option to skip some activities. Thus, there exist missing classes in some participants' data as well as unbalanced data samples across the classes. During data collection, sensors of the same model are instrumented on different subjects running at a sampling rate of 100 Hz.
2) *USCHAD.* This dataset [42] is collected from 14 participants performing 10 types of locomotions (i.e., walking forward, walking left, walking right, walking upstairs, walking downstair, running forward, jumping up, sitting, standing and sleeping). All activities are performed by each subject in a controlled environment. A sensor (acclerometer and gyroscope) with a sampling rate of 100 Hz is attached to the right hip of each participant. 5 data trials per activity were collected per participant, and the duration of each data trial varies.
3) *WISDM.* This dataset [43] contains a large number of subjects. Raw accelerometer and gyroscope data have been collected from a smartphone in each participant's pant pocket at a rate of 20 Hz. There are a total of 51 test subjects performing 7 locomotion activities (i.e., walking, jogging, stairs, sitting, standing, kicking soccer ball, playing tennis) for 3 minutes apiece to get equal class distribution.
4) *MobilityAI.* The mobility analysis by artificial intelligence (MobilityAI) dataset is collected from 25 in-hospital patients whose ages are ≥ 65. The objective of collecting such a dataset is to monitor patients' mobility status during their hospital stay, and to quantify the effectiveness of an early mobilization protocol. To identify the most suitable device, four IMU sensors from different vendors are attached to patients' waists using elastic bands as shown in Fig. 3. Each subject performs four activities (lying for 5 minutes, sitting for 5 minutes, standing for 5 minutes and 20 meters walking) for mobility status assessment. The four devices utilized are MetaMotionR [44], Fitbit Versa [45], Mox One [46] and Actigraph [47]. All sensors are set to have a 50 Hz sampling rate, and only accelerometer readings are captured. Due to data outage and the limited functional mobility of some participants, there exist missing classes in the dataset.

4.2 Implementation and Evaluation Process

In addition to BMTL and TMTL discussed in the previous section, we have also implemented four baseline models: a single-task learning model (STL), a pooling task model (PTM) and MetaSense [28]. STL is trained solely on the target domain for comparison with domain adaptation, whereas PTM is trained with mixed training data from all source domains, to highlight the domain shift problem.

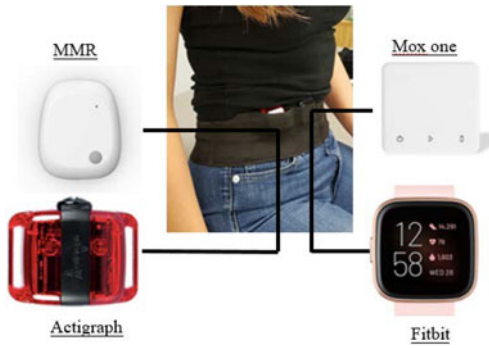


Fig. 3. The sensors and their placement when collecting data for the MobilityAI dataset.

Data Preparation. Although deep neural networks can directly learn useful features from raw data [11], data preprocessing such as interpolation, noise filtering, normalization, and the division of sliding windows are still needed. A Butterworth low-pass filter [48] with a cut-off frequency of 10 Hz is employed to remove high frequency noise from interpolated data. After low-pass filtering, we normalize the data, calculate similarity metrics, and then segment it into sliding windows with a fixed length of 2 seconds with 80 percent overlapping for all datasets. To eliminate the impact of different orientations of sensors in MobilityAI, we rotate the orientations of Actigraph, Mox one and Fitbit 3-axis accelerometers to be aligned with that of MetaMotionR.

Implementation. The implementation of feature extractor follows DeepConvLSTM [4] for IFLF, STL and PTM models. It includes four layers of 1D CNN and two LSTM layers with 128 hidden units and a drop-out rate of 0.25 to prevent overfitting [49]. The CNN layers have 64 channels with kernel size 5 and stride 1. For a fair comparison with MetaSense, we also implement TMTL use the same network architecture as in [28] based on an open source implementation of MAML [50]. The feature extractor network has five CNN layers and two fully-connected layers, including 128 and 64 hidden units respectively. The reason for not including LSTM layers to MetaSense is two-fold: 1) MAML based approaches require a 2-steps update on layer parameters, keeping intermediate variable for calculated gradients without actual updating. But existing deep learning libraries combines gradient calculation with backpropogation for recurrent neural network parameters, leaving no API for gradient calculation only, and 2) as a model agnostic approach, it is interesting to investigate the performance of IFLF without LSTM as well. Tasks in MetaSense are sampled both within and cross different source domains, keeping activity labels consistent across all tasks.

For STL and PTM, the output layer corresponds to a fully-connected layer with a softmax activation function. Both models are trained with a RMSProp optimizor [51] at a learning rate of 10^{-3} and a decay factor of $p = 0.9$. The maximum iteration number is set to be 100. IFLF models utilize the aforementioned network structure as L_θ, the number of S_{ϕ^k} branches is determined by the number of source

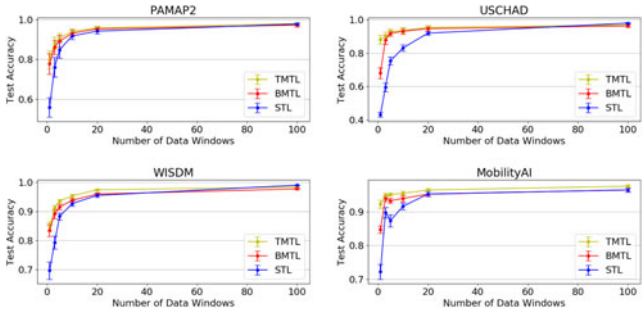


Fig. 4. Evaluation on subject difference across all datasets when gradually increase the number of data windows per activity class from $\mathcal{D}_{tgt}$. The test accuracy and standard error are averaged across different subjects in leave-one-out experiments.

domains, and each S_{ϕ^k} may have a different output shape depending on the number of classes. IFLF models are trained with an Adam [52] optimizer at a learning rate of $10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$, and hyper-parameters $\mu = 0.8$. The batch size is set to 100 and the maximum number of training epochs is 30 with early-stopping. In each epoch, TMTL samples m pairs of (x_a, x_p) and m pairs of (x_a, x_n) to form m^2 (x_a, x_p, x_n) triplets from each source domain as task T_k. Similarly, n pairs of (x_a, x_p) and n pairs of (x_a, x_n) are sampled to form n^2 (x_a, x_p, x_n) triplets as validation set $(m > n)$. In most experiments, we set $m = 100$, $n = 10$ and hyperparameter $\epsilon = 0.4$. The hyper-parameters and optimizer of each model are the same across all datasets.

Evaluation Process. Leave-one-domain-out evaluations are conducted on all datasets. Under different problem settings, a domain can be a subject, a sensor device or a combination of subject and device. In each experiment, a target domain was randomly selected. Similarity calculation and model meta-training utilize data from source domains only.

After the IFLF model is trained, we randomly sample a fixed test set from $\mathcal{D}_{tgt}$, and randomly select i data windows (of 2s length) per class from the remaining data as the training set to update a trained MetaSense model and an arbitrary S_{ϕ^k} layer of the IFLF model. This process is also called i-shot learning. An STL model is also trained on this training set and test accuracy is recorded for each model by gradually increasing i from 1 to 100. Each evaluation is repeated 5 times and the average is reported.

4.3 Results

4.3.1 Evaluate With Various Domain Shifts

In order to evaluate IFLF's capability to handle domain shifts, three types of experiments are conducted on subject difference, device diversity and their combination (with both unseen participant and device).

Subject Difference. Fig. 4 shows the averaged test accuracy with standard errors across all datasets. The average is computed over all subjects of each dataset.

As demonstrated in Fig. 4, in terms of the overall test accuracy among the four models follows, BMTL is better than STL, and TMTL is better than BMTL when few data samples are available from $\mathcal{D}_{tgt}$. However, since STL is solely trained on the target domain, when i is sufficiently large, its accuracy approaches 100 percent and tends to be better than both IFLF models. We also observe that with

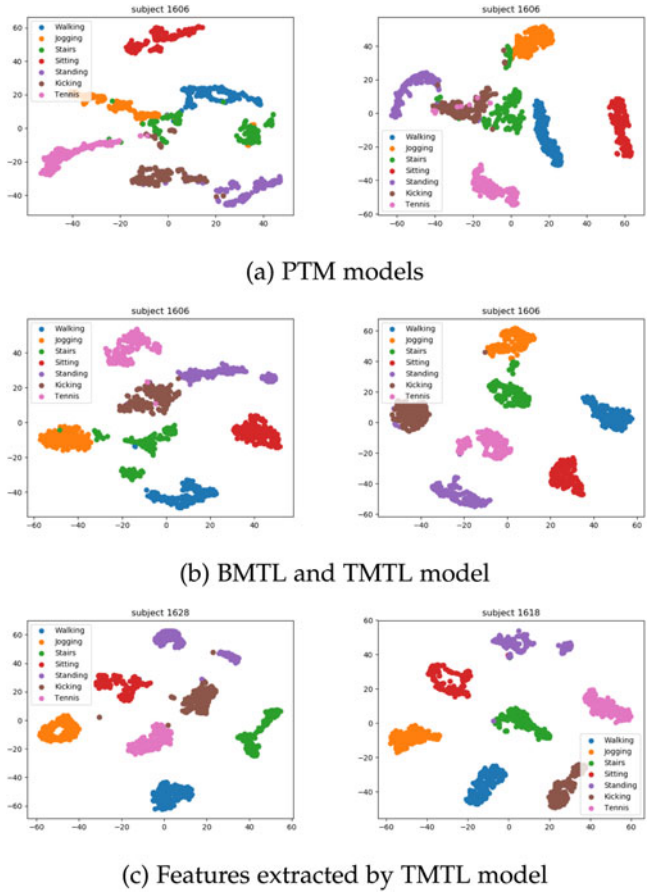


Fig. 5. t-SNE visualization of the learned representations. We visualize the features from each model with the output of L_θ by projection them on 2D space. This example is generated by different models with subject 1630 to 1646 as source domains. (a) are PTM models; (b) left is BMTL and (b) right is TMTL; (c) are from randomly selected subjects as target domains other than the one in (a) and (b).

different subjects as the target domain, the converging rate of STL is dramatically different, an indication of subject differences.

To visualize the features produced by the methods, Fig. 5 shows the t-SNE of unseen subject's features produced by each model [14]. When generating these plots, we randomly pick a set of subjects as source domains (e.g., subject 1630 to 1646) and one subject as the target domain. The comparison is made among a PTM model, leaving this subject out in the BMTL model and TMTL model. To demonstrate the generality of the results, plots from two PTM models are presented with different random splits of the training set for each source domain. Although the figures are generated from the WISDM dataset, similar observations can be made for other datasets. Also, we examine the existence of invariant features across domains by extracting features from unseen target domains.

From Figs. 5a and 5b, it is clear that the features for different activities generated by PTMs are entangled regardless of the splits between training and validation set. BMTL improves the separation among activities to some extent, whereas TMTL generates a set of features with clear clustered structures and large margins. Fig. 5c further demonstrates the existence of invariant features across domains as features extracted from unseen target domains are well-

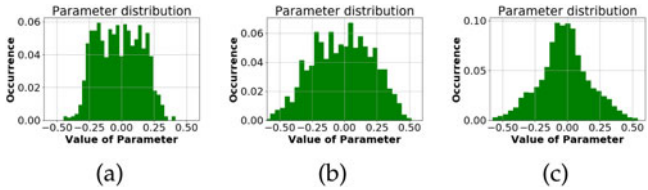


Fig. 6. Comparison of the parameter ϕ's distribution for different models. From (a) to (c) are: STL, BMTL, TMTL. X-axis is the value of parameter, y-axis is the normalized occurrence.

clustered and linearly separable. Due to space limit, only 3 subjects are shown in Fig. 5, but other subjects exhibit similar characteristics. This observation also explained why the fast adaptation can be made on any trained S_{ϕ^k} layer of BMTL and TMTL. As the features are well separated, different choices of the task-specific layer for parameter update have little impact on the performance. However, as the amount of data is quite limited in the fast adaptation step, initializing L_θ randomly will impair the performance of IFLF. But even in this case, we find the IFLF models still work better than STL.

To further illustrate the advantage of TMTL over the other two models, the sparsity of the parameters (ϕ) of the learned task-specific layer is compared. In Fig. 6, we plot the distribution of ϕ in the three models trained on WISDM dataset.

As shown in Fig. 6, the parameters in STL are roughly a uniformly distributed between -0.4 and 0.4. In comparison, the parameters of BMTL follow a zero mean Gaussian distribution but with a large variance. Lastly, the majority of TMTL parameters are centered around 0 with a noticeably smaller variance (than that of BMTL). The sparsity of task-specific layer's parameters indicates the easiness of separating the generated feature representations.

Device Diversity. Similar to subject difference experiment, IFLF can also tackle domain shift problems caused by device diversity. To understand the behavior of STL, BMTL, and TMTL to handle device diversity, we consider data from different sensor devices attached to the same subject at the same on-body position. Since only the MobilityAI dataset has such characteristics, it is utilized in the subsequent experiments.

In following experiments, data from Actigraph, Fitbit, Mox One constitute the source domains, while MMR device's data is selected as D_{tgt}. One data window per activity sampled from D_{tgt} is utilized. To demonstrate the presence of domain shifts between different devices, besides the STL, BMTL and TMTL models, we further present the confusion matrix from PTM trained with data from Actigraph, Fitbit and Mox one but tested on MMR. From the confusion matrices in Table 2, it is clear that BMTL benefiting from invariant feature learning outperformed STL, while TMTL rarely misclassifies any activity in the dataset. Table 2 d shows that PTM is incapable of learning robust features that generalize well to the target domain.

Fig. 7 visualizes features extracted by PTM (the two plots in Fig. 7a are generated from different random splits of the training data), BMTL and TMTL models. Similar to the case of subject differences, we observe that the PTM models fail to extract separable features from MetaMotionR data while BMTL does a better job, but the resulting features are still not well-clustered. In contrast, TMTL gives rise to

TABLE 2
The Experiment on Sensor Diversity

(a) The confusion matrix of the single task model trained on MetaMotionR sensor data.

	Lying	Sitting	Standing	Walking
Lying	0.98	0.02	0	0
Sitting	0.02	0.98	0	0
Standing	0	0.35	0.65	0
Walking	0	0	1	0

(b) The confusion matrix of the BMTL model fast adapted with MetaMotionR data.

	Lying	Sitting	Standing	Walking
Lying	0.98	0.02	0	0
Sitting	0.02	0.98	0	0
Standing	0	0.42	0.58	0
Walking	0	0	0	1

(c) The confusion matrix of the TMTL model fast adapted with MetaMotionR data.

	Lying	Sitting	Standing	Walking
Lying	0.98	0.02	0	0
Sitting	0.08	0.90	0.02	0
Standing	0	0.04	0.96	0
Walking	0	0	0	1

(d) The confusion matrix of the PTM model trained without MetaMotionR and tested on MetaMotionR data directly.

	Lying	Sitting	Standing	Walking
Lying	1	0.02	0	0
Sitting	0	0.99	0.01	0
Standing	0.01	0.94	0.05	0
Walking	0	0.48	0.52	0

Confusion matrices are generated with only 1 data window per activity involved.

features with clear boundaries in the feature space and clustered structures. Fig. 8 depicts the distribution of parameters ϕ in the task specific layers of each model. Similar to Fig. 6, we observe that TMTL has the most sparsity, followed by BMTL, whereas STL leads to the least sparsity. The evaluation on device diversity further demonstrates the IFLF models' capability of learning invariant features across domains.

Both Subject and Device are Unseen. Encouraged by the promising results on the subject difference and device diversity experiments, we further evaluate situations where both device and subject are unseen to the model. We randomly selected a subset from the MobilityAI dataset, which includes 8 subjects with waist attached sensors. As each subject has 4 sensor devices attached, the total combination of subject and sensor is 32. A pair of BMTL and TMTL models are trained on data from 5 participants each with 3 sensors attached, and evaluated on the 4th device data collected from participants other than the five in the training data. We conduct leave-one-out experiment on both subject and device. To keep brief, we only present results from one division of subjects as the other cases are quite similar. Specifically, the meta-training set includes data from Subject 1 to 5. The test data is from Subject 6, 7 and 8.

Fig. 9 compares the performance when both the subject and device are unseen to IFLF models. In this case, both BMTL and TMTL have better performance than STL, especially when the

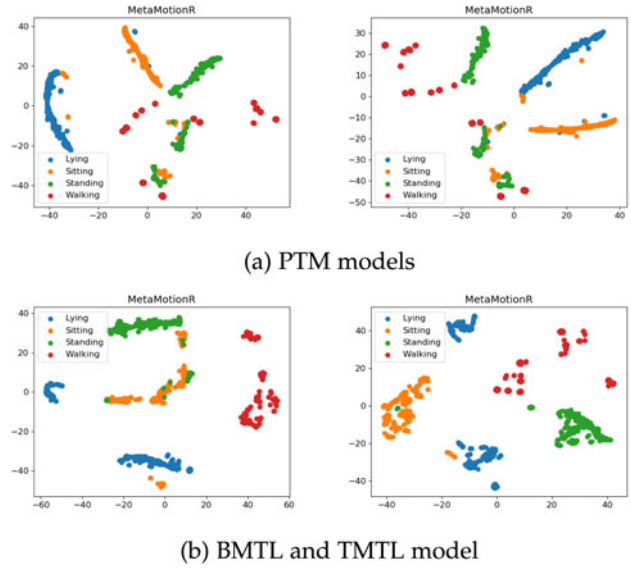


Fig. 7. t-SNE visualization of the learned representations. This example is generated by different models on MobilityAI dataset. (a) are PTM models; (b) left is BMTL and (b) right is TMTL.

amount of labeled data from the target domain is small. Compared to Fig. 4, we observe large gaps between TMTL and STL test accuracy. This can be attributed to the significant diversity among different devices. Furthermore, by comparing the results for Subject 6 and 8 for the same device (e.g., Mox One or MMR), we find that Subject 8 appears to have larger differences from subjects in the training set than Subject 6. Despite such differences, TMTL consistently outperforms BMTL and STL. Meanwhile, large standard errors are observed with STL as it is heavily dependent on the training set.

In summary, IFLF is a general method to capture invariant features, and works well regardless of the cause of domain shifts.

4.3.2 Comparison With MetaSense

To this end, we conclude TMTL outperforms STL, BMTL in handling domain shifts caused by subject and device diversity. Next, we present the comparison between TMTL without LSTM and the state-of-the-art meta-learning model for HAR, MetaSense. With PAMAP2, WISDM and USCHAD which only contain data from a single sensor, we compare the performance of these two models on mitigating domain shifts caused by subject differences. With MobilityAI dataset that include data from multiple subjects and devices, the performances of the two models when both subject and device are unseen are compared.

From Fig. 10, it is clear that in 19 out of 20 cases tested, TMTL performs better than or comparably as MetaSense.

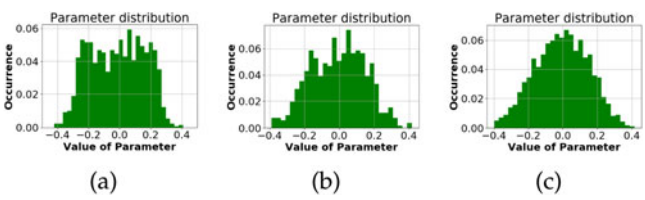


Fig. 8. Comparison of the parameter ϕ's distribution for different models. From (a) to (c) are: STL, BMTL, TMTL.

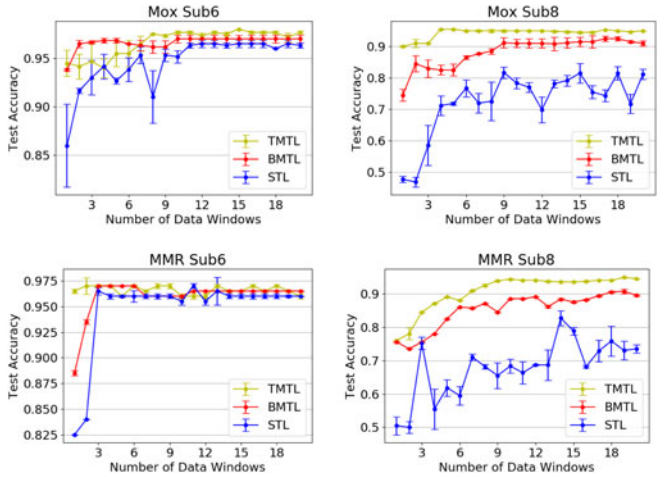


Fig. 9. Combinations of different sources of diversity (Both the test subject and the test device are not included in the training data for IFLF models).

The advantage of TMTL is more prominent when few data samples are available from the target domain. The superior performance of TMTL over MetaSense for very few shot learning is due to the fact that MetaSense needs to update the entire model whereas TMTL only updates a task-specific layer. The latter approach is more data efficient and less likely to over-fit. However, as the number of available data samples from the target domain grows, the performance of MetaSense is comparable to or even slightly better than TMTL with sufficient labeled data from the target domain.

Compared to the TMTL with LSTM in Section 4.3.1, the performance of TMTL with LSTM drops by an average 2.35 percent for 1-shot learning across all datasets. However, as the number of shots increases to more than 10, the performance gap is negligible ($<0.5\%$). This fact further demonstrates that IFLF is a model agnostic method and can be used in conjunction with any suitable network architecture for feature extraction.

4.3.3 Similarity Metric and Fast Adaptation

Using the similarity metric defined in Section 3.3, in this section, we first compare inter-subject and intra-activity similarity. The purpose of this study is to understand whether there exist subjects with similar movement patterns in all activities, and whether there exists an activity with little

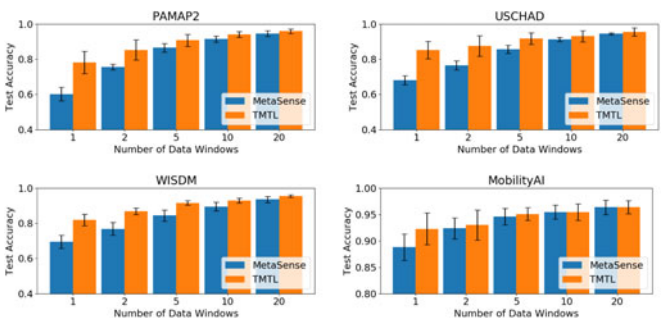


Fig. 10. Comparison between TMTL and MetaSense on different domain adaptation tasks. Results are reported for 1, 2, 5, 10, 20-shots (data windows) per activity class from $\mathcal{D}_{tgt}$ with average test accuracy and standard error.

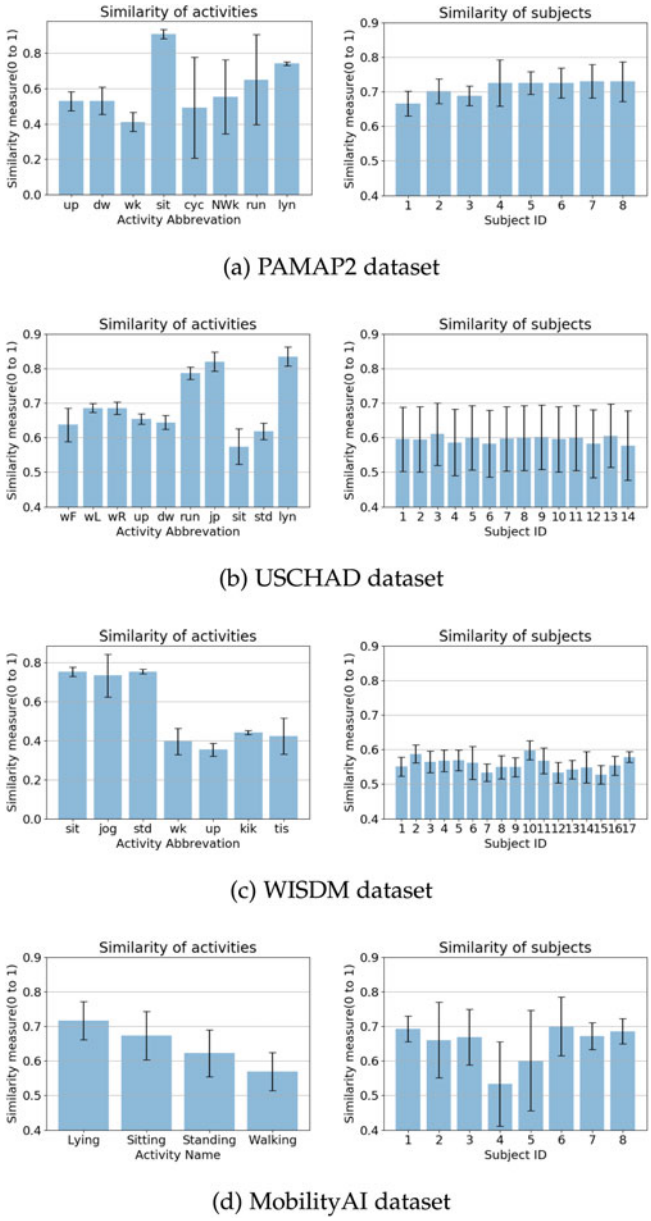


Fig. 11. The evaluated similarity measure on each datasets. Plots on the left are similarity of activities, while plots on the right are similarity of subjects. Abbreviations in (a) (b) (c) are activities: upstairs, downstairs, walking, sitting, cycling, Nordic walking, running and lying; walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing and lying on bed; sitting, jogging, standing, walking, upstairs, kicking soccer ball, playing tennis

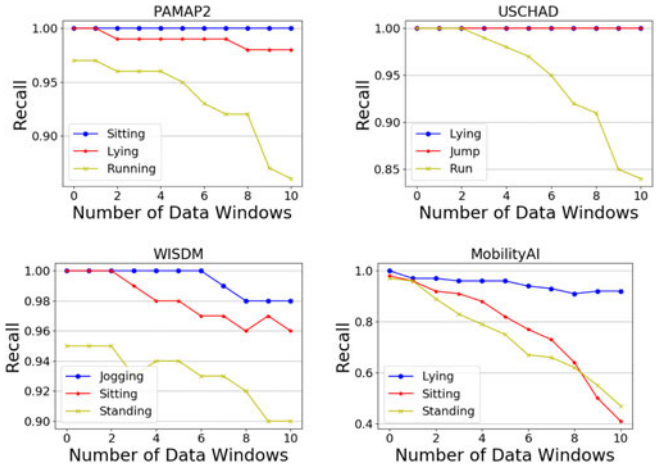


Fig. 12. The evaluation of applying similarity metric in the fast adaptation step. Activities with top 3 similarity scores are replaced in each dataset.

inter-subject variation. Second, we evaluate the use of intra-activity similarity in further improving data efficiency in fast adaptation. For activity similarity (left column in Fig. 11), we compute the averages and standard deviations of similarities of the same activity across different subjects. The right column in Fig. 11 shows the similarity scores between one subject and the others averaged over matching activities.

From Fig. 11, it is clear that some activities have higher similarity scores than the others. For example, in the PAMAP2 dataset, 'sitting' appears to be similar across different subjects, while 'walking' has the least similarity. This indicates diverse postures during walking but less variation during sitting in this dataset. On the other hand, when examining subject similarities, we find that some degree of similarity exists across almost all subjects (e.g., with a similarity score > 0.5). However, no two subjects perform activities the same way (e.g., the maximum similarity score is below 0.8). An exception is observed in the MobilityAI dataset, Subject 4 and 5 have noticeably lower similarity from others and larger standard deviations. It is because these two people have mobility issues and have to stand or walk with a rollator. It is expected that larger diversity exists among older adults with different underlying physical conditions. Homogeneity is more pronounced among younger populations as evident from the first three datasets. Another interesting observation is that depending on sensor placements, participants and the protocol of data collection, the same activity may have different cross-subject similarity in different datasets. For example, 'lying on bed', 'sitting' and 'standing' have noticeably different similarity scores across the four datasets.

Next, we investigate whether target domain data can be safely replaced by data from source domains for certain activities in fast adaptation. We first sample 10 labeled data windows from each activity from $\mathcal{D}_{tgt}$ to perform fast adaptation on a trained TMTL with the procedure described earlier. Then, the ten data windows of activities with top 3 similarity scores are replaced with data randomly sampled from $\mathcal{D}_{src}$ to update the TMTL model. The performance metric used is $recall = \frac{tp}{tp+fn}$, as it reveals whether a single activity is wrongly classified. A non-decreasing recall as more and more sampled data from the target domain is replaced by source domain data implies that doing so has little impact on the performance with less data collection efforts. The results for different datasets are shown in Fig. 12.

As evident from Fig. 12, 'sitting' from PAMAP2, 'lying' and 'jump up' from USCHAD can be safely replaced with marginal impacts on the recall. This result is in accordance with the similarity scores presented in Fig. 11 as all these three activities have similarity score ≥ 0.8. Although 'lying' appears in three datasets, it can only be safely replaced in the USCHAD. In other words, by sampling activities with high similarity scores and small inter-subject variance from source domains, we can improve data efficiency by 12 to 20 percent on top of the reduction from fast adaptation.

5 Conclusion and Future Work

In this paper, we presented an invariant feature learning framework based on meta-training and multi-task learning paradigm to effectively address domain shifts and data shortage in HAR. As demonstrated in Section 4.3, IFLF has been shown to work efficiently in few-shot learning, especially when the number of shots are few (1 or 2 shots). A $> 10\%$ performance margin has been observed when compared to MetaSense under such condition. Also, the proposed TMTL model implicitly handles class imbalance and class missing problems as well. A similarity measure was proposed to further reduce the amount of data required in fast adaptation step. Though only domain shifts due to human and device variations have been considered in this work, we believe it can also be applied to handle sensor placement diversity, which will be investigated as part of our future work.

Another research topic is to explore the application of domain generalization (DG) in sensor-based HAR. With the knowledge acquired from source domains, a model with DG can be directly utilized on the target domain without any data from it. Replacing current relatively simple feature extractor with a more SOTA neural network such as DeepSense will be helpful to further improve the model generalization, and we will leave it as a future work.

Acknowledgments

This work was supported in part by the NSERC Discovery and in part by the CREATE programs.

References

[1] H. Kwon, *et al.* "Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," 2020, *arXiv: 2006.05675*.

[2] E. Soleimani and E. Nazerfard, "Cross-subject transfer learning in human activity recognition systems using generative adversarial networks," 2019, *arXiv: 1903.12489*.

[3] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervasive Mobile Comput.*, vol. 57, pp. 1–13, 2019.

[4] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[5] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 17th Int. Conf. Networked Sens. Syst.*, 2010, pp. 233–240.

[6] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 766–785, Mar. 2021.

[7] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei, "Few-shot learning via learning the representation, provably," 2020, *arXiv: 2002.09434*.

[8] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1082–1090, Aug.2008.

[9] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International Workshop On Ambient Assisted Living*. Berlin, Germany: Springer, 2012, pp. 216–223.

[10] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and KDA-based SVMs," *Int. J. Distrib. Sensor Net.*, vol. 10, no. 5, 2014, Art. no. 503291.

[11] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[12] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, 2018, pp. 1–16.

[13] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 351–360.

[14] L. V. d. Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[15] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

[16] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5716–5726.

[17] Y. Tas and P. Koniusz, "CNN-based action recognition and supervised domain adaptation on 3D body skeletons via kernel feature maps," 2018, *arXiv: 1806.09078*.

[18] B. Wang, J. A. Mendez, M. Cai, and E. Eaton, "Transfer learning via minimizing the performance gap between domains," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10644—10654.

[19] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8049–8057.

[20] A. Mathur, A. Isopoussu, F. Kawsar, N. Berthouze, and N. D. Lane, "Mic2mic: Using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems," in *Proc. 18th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2019, pp. 169–180.

[21] A. Akbari and R. Jafari, "Transferring activity recognition models for new wearable sensors with deep generative domain adaptation," in *Proc. 18th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2019, pp. 85–96.

[22] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, "Stratified transfer learning for cross-domain activity recognition," in *2018 IEEE Int. Conf. Pervasive Comput. Commun.*, 2018, pp. 1–10.

[23] J. Schmidhuber, "Evolutionary principles in self-referential learning," Diploma thesis, Institut f. Informatik, Tech. Univ. Munich, Germany, 1987.

[24] Y. Bengio, S. Bengio, and J. Cloutier, "Learning a synaptic learning rule," in *Proc. Int. Joint Conf. Neural Netw.*, 1991, p. II–A969.

[25] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.-Vol. 70*, 2017, pp. 1126–1135.

[26] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. Moura, "Few-shot human motion prediction via meta-learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 432–450.

[27] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," 2018, *arXiv: 1802.01557*.

[28] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, "Metasense: Few-shot adaptation to untrained conditions in deep mobile sensing," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 110–123.

[29] C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, and C. Gagné, "A principled approach for learning task similarity in multitask learning," 2019, *arXiv: 1903.09109*.

[30] B. Wang, H. Zhang, P. Liu, Z. Shen, and J. Pineau, "Multitask metric learning: Theory and algorithm," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 3362–3371.

[31] X. Sun, H. Kashima, R. Tomioka, N. Ueda, and P. Li, "A new multi-task learning method for personalized activity recognition," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 1218–1223.

[32] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technolo.*, vol. 3, no. 2, pp. 1–30, 2019.

[33] A. Kumar and H. Daume, III, "Learning task grouping and overlap in multi-task learning," 2012, *arXiv:1206.6417*.

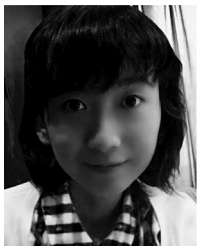
[34] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[35] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv: 1706.05098*.

[36] M. Zeng *et al.* "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc.6th Int. Conf. Mobile Comput., Appl. Serv.*, 2014, pp. 197–205.

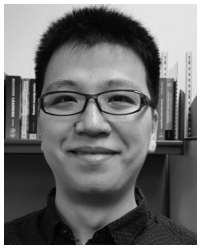
[37] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, 2019.

[38] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit*, 2015, pp. 84–92.
[39] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
[40] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. TASSP-26, no. 1, pp. 43–49, Feb. 1978.
[41] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, 2012, pp. 108–109.
[42] M. Zhang and A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. 2012 ACM Conf. Ubiquitous Comput.*, 2012, pp. 1036–1043.
[43] G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and smartwatch-based biometrics using activities of daily living," *IEEE Access*, vol. 7, pp. 13190–133202, Sep. 2019.
[44] "Metamotionr product description," Accessed: Apr. 1, 2020. [Online]. Available:https://mbientlab.com/metamotionr
[45] "Fitbit Versa product description," Accessed: Apr. 1, 2020. [Online]. Available:https://www.fitbit.com/us/products/smartwatches/versa
[46] "MOX1 product description," Accessed: Apr. 1, 2020. [Online]. Available:https://www.accelerometry.eu/mox1/
[47] "Actigraph product description," Accessed: Apr. 1, 2020. [Online]. Available:https://actigraphcorp.com/actigraph-wgt3x-bt/
[48] S. Butterworth, "On the theory of filter amplifiers," *Exp. Wirel. Wirel. Eng.*, vol. 7, no. 6, pp. 536–541, 1930.
[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[50] L. Long, "Maml-pytorch implementation," 2018. [Online]. Available: https://github.com/dragen1860/MAML-Pytorch
[51] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio, "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization," 2015, *arXiv:1502.04390V1*.
[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.



Yujiao Hao received the B.S. degree in software engineering and the M.S. degree in computer science from the Department of Software Engineering, Northeastern University, Shenyang, China. Since 2018, she has been currently working toward the Ph.D. degree from the Department of Computing and Software, McMaster University, Hamilton, ON, Canada. Her research interests include sensor-based human activity recognition and motion tracking.

Rong Zheng (Senior Member, IEEE) is currently a professor at the Department of Computing and Software, McMaster University, Hamilton, ON, Canada. She is an expert in wireless networking, mobile computing, and mobile data analytics. She is the recipient of the National Science Foundation CAREER Award in 2006, and was a Joseph Ip Distinguished Engineering Fellow from 2015 to 2018. He was the Editor for the *IEEE Transactions on Wireless Communications* and *IEEE Transactions on Network Science and Engineering*. She is currently an Editor for the *IEEE Transactions on Mobile Computing*.



Boyu Wang received the B.Eng. degree in electronic information engineering from Tianjin University, Tianjin, China, the M.Sc. degree in electrical and computer engineering from the University of Macau, Zhuhai, China, and the Ph.D. degree in computer science from McGill University, Montreal, QC, Canada. He is currently an assistant professor at the Department of Computer Science, University of Western Ontario, London, ON, Canada. He is also affiliated with the Brain and Mind Institute and the Vector Institute. He was a postdoctoral research fellow with the University of Pennsylvania and Princeton University. His research interests include machine learning theory, algorithms, and applications.