Local Domain Adaptation for Cross-Domain Activity Recognition

Jiachen Zhao, Fang Deng, *Senior Member, IEEE*, Haibo He, *Fellow, IEEE*, and Jie Chen, *Fellow, IEEE*

***Abstract*—Sensor-based human activity recognition (HAR) aims to recognize a human's physical actions by using sensors attached to different body parts. As a user-specific application, HAR often suffers poor generalization from training on an individual to testing on another individual, or from one body part to another body part. To tackle this cross-domain HAR problem, this article proposes a domain adaptation (DA) method called local domain adaptation (LDA), whose core is to align cluster-to-cluster distributions between the source domain and the target domain. On the one hand, LDA differs from existing set-to-set alignment by reducing the distribution discrepancy at a finer granularity. On the other hand, LDA is superior to the class-to-class alignment because it can provide more accurate soft labels for the target domain. Specifically, LDA contains three main steps: 1) groups the activity class into several high-level abstract clusters; 2) maps the original data of each cluster in both domains into the same low-dimension subspace to align the intracluster data distribution; 3) predicts the class labels for target domain in the low-dimension subspace. Experimental results on two public HAR benchmark datasets show that LDA outperforms state-of-the-art DA methods for the cross-domain HAR.**

***Index Terms*—Domain adaptation, human activity recognition, transfer learning, wearable sensor.**

I. Introduction

SENSOR-BASED human activity recognition (HAR) aims to identify humans' actions using the data collected by sensors worn on a human body, which has many potential applications in the fields of human-computer interaction [1], healthcare [2], ubiquitous computing [3], etc. Fig. 1 shows the diagram of sensor-based HAR, where inertial measurement units (IMU) are installed on different positions of the human body and used to record the motion patterns of the torso and limbs. Then, the collected data can be used to monitoring daily activity to provide personalized recommendations or assist patients with chronic impairments. Comprehensive surveys in [4] and [5] provide more technical details on sensor-based HAR.

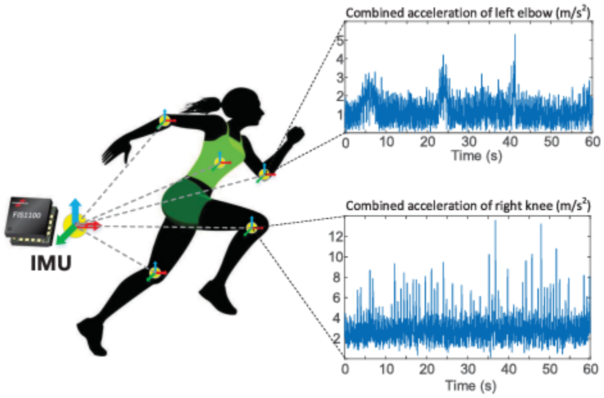

Combined acceleration of left elbow (m/s^2)
1
2
3
4
5
0
10
20
30
40
50
60
Time (s)
FIS1100
IMU
Combined acceleration of right knee (m/s^2)
2
4
6
8
10
12
0
10
20
30
40
50
60
Time (s)

Fig. 1. Diagram of sensor-based HAR. Five IMUs are placed on the chest, left arm, right arm, left leg, and right leg to record the motion pattern of each position to identify the human body motion further. (Figure source: https://www.tiankong.com)

It is a typical scenario in real life that designers can only collect limited labeled data from a certain number of individuals, but designers need to identify the actions of many other people. For example, smart bracelet manufacturers cannot label all consumers' data to train their HAR model but need to identify every user's activity. Another common scenario is that people usually put their smartphones in different positions of the body over time, such as in the left hand or the right pocket of the pants. Many existing papers [6], [7] pointed out that a HAR model trained on certain persons or certain positions (i.e., the source domain) does not generalize well to other persons and positions (i.e., the target domain). Such problems are called cross-person HAR and cross-position HAR, respectively. Moreover, both problems are referred to as cross-domain HAR. To clearly show the negative effect of domain discrepancy, we perform simple cross-person HAR experiments on the DSADS dataset proposed in [8]. We use the signal of the accelerometer and gyroscope mounted on the torso, and train a nearest neighbor classifier on the data of one object to recognize the other objects' actions. We also adopt 10-fold cross-validation to get robust results. As shown in the accuracy variation matrix in Fig. 2, the diagonal elements display the average self-action recognition accuracy

Manuscript received December 26, 2019; revised August 17, 2020; accepted October 20, 2020. Date of publication December 9, 2020; date of current version January 13, 2021. This work was supported in part by the Key Program of National Natural Science Foundation of China under Grant 61933002 and in part by National Science Foundation under Grant ECCS 1917275. This article was recommended by Associate Editor G. Fortino. *(Corresponding author: Haibo He.)*

Jiachen Zhao and Fang Deng are with the Department of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, 401120, China (e-mail: zhao_jiachen@163.com; dengfang@bit.edu.cn).

Haibo He is with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: he@ele.uri.edu).

Jie Chen is with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Shanghai 200092, China, and also with the Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University, Shanghai 200092, China (e-mail: chenjie@bit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/THMS.2020.3039196.

Digital Object Identifier 10.1109/THMS.2020.3039196

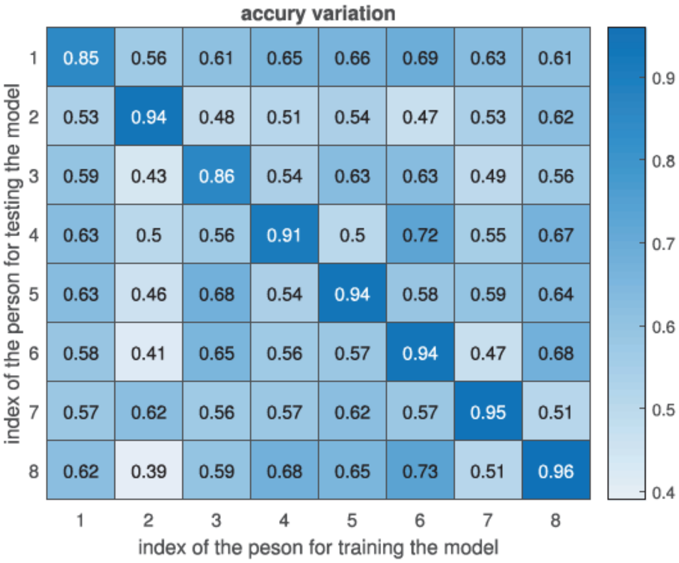


Fig. 2. Classification accuracy matrix of cross-person activity recognition without considering the differences between individuals.

of 91.87%, while the cross-person accuracy is only 57.71%. This result shows it is necessary to adapt the classifier for each individual.

DA methods have been applied to cross-domain HAR with some success. Such as Hong *et al.* [9] presented a semipopulation approach for cross-person HAR. Wang *et al.* [6] proposed a deep siamese neural network for cross-position HAR. Wang *et al.* [10] proposed an extreme learning machine based method for cross-location HAR. Different from traditional classification methods, DA takes the distribution discrepancy between domains into account. Therefore, the main idea of DA methods is to reduce the distribution discrepancy by learning domain-invariant representations. The most related DA methods of this article are transfer component analysis (TCA) [11] and stratified transfer learning (STL) [12]. TCA is one of the most classic methods, which maps the original features to a reproducing kernel Hibert space (RKHS). In the RKHS, the maximum mean discrepancy distance (MMD) between domains is minimized. However, TCA has restricted performance because it tries to map all the source samples and the target samples into the same subspace, where may still exist obvious distribution discrepancy. TCA can be regarded as a global domain adaptation method that matches the set-to-set distributions. To overcome the drawbacks of TCA, STL performs class-to-class distribution alignment based on the predicted soft labels of the target domain. However, it is usually difficult to give accurate soft labels at the class level.

Different from the abovementioned works, this article takes a compromise between the set-to-set and the class-to-class alignment, then proposes a novel cluster-to-cluster distribution alignment method called local domain adaptation (LDA). LDA avails the hierarchy of human activity categories. For example, sitting, lying, and standing can be summed into “static activity cluster,” “walking, running, cycling” belong to “dynamic cluster.” By performing cluster-to-cluster distribution alignment, LDA can reduce the distribution discrepancy at a finer granularity than TCA, and can match the distribution based on more accurate soft labels for the target domain than STL. Specifically, LDA divides all individual activity classes into several prespecified abstract activity clusters based on human knowledge. Then, LDA trains multiple classifiers on the source domain and predicts the cluster labels on the target domain with voting decision. Finally, LDA maps the samples in the same cluster in both domains into the same subspace and predicts the class label for the target domain. Furthermore, we propose autoclustering LDA (ACLDA) by introducing a clustering technique named to group the activity classes into clusters instead of that predefined clusters.

To sum up, our study has three main contributions as follows.

1) We propose a novel domain adaptation method LDA for cross-domain HAR, whose main idea is to align the distributions of source and target domains at the cluster level. Taking a compromise between set-to-set and class-to-class alignment methods, LDA relieves the obvious disadvantages of both methods.
2) We improve LDA by canceling the process of artificially defining clusters and propose ACLDA. ACLDA can find the local clusters automatically and then performs the local transfer without human intervention. Therefore, ACLDA has the potential to solve a wider range of DA problems.
3) We conduct comprehensive experiments for cross-persons HAR on two public datasets. The experiment results demonstrate the effectiveness of LDA and ACLDA.

The rest of this article is organized as follows. Section II reviews the related works. Section III presents the proposed LDA and ACLDA. Section IV reports the experiment results. Finally, Section Vconcludes this article.

II. Related Works

A considerable number of researchers have investigated sensor-based HAR [13]. Bulling *et al.* [14] introduced the general-purpose process pipeline for designing and evaluating activity recognition systems. Anguita *et al.* [15] built a multiclass support vector machine model on a smartphone platform to implement six-class locomotion activities recognition. Several deep learning methods, such as the one-dimensional convolutional neural network [16] and recurrent neural network [17] were also used in HAR system. However, such deep models often require higher computation time and memory resources. All the abovementioned works assume that the persons in the test set and the training set are the same, so the data follows the same distribution. However, it is too expensive, even impossible, to collect and label data from consumers or patients using the product. To capture the interpersonal variability, one can increase the amount of training data from different persons or use person-independent features. The former is costly, and the latter is a tradeoff between a discriminative feature set and a generic feature set. Differently, this article aims to tackle this challenge via a transfer learning method.

The main idea of transfer learning (TL) is to capture the knowledge from a source domain $\mathcal{D}_s$ to solve the learning task in a target domain $\mathcal{D}_t$, where two domains are different but related.

For the classification task, transfer learning is also named as domain adaptation. Its goal is to learn a classifier from the source domain to classify the samples in the target domain [18], [19]. Throughout this article, we do not distinguish the differences between TL and DA. Generally, DA problem has two types of settings: unsupervised DA, which assumes that only source samples are labeled; supervised DA, which assumes that parts of target samples are labeled. In this article, we consider the more challenging unsupervised setting.

Recently, several researches investigates how to apply transfer learning methods to solve cross-domain HAR problems, including cross-sensor-modalities [20], [21], cross-sensor-installation-position [6], [12], cross locations [22], [23], and cross persons [24]–[26]. Deng *et al.* [27] proposed the TLRKELM model for HAR, whose main idea is to train a reduced kernel extreme learning machine on the source domain and use it to classify the target samples, then label the high confident samples and add them to the training set. Zhao *et al.* [28] proposed TransEMDT model to recognize different persons activities based on a smartphone. Both TLRKELM and TransEMDT belong to the instance-based transfer learning methods. In contrast, our method is feature based and distinguishes from existing works by taking a compromise between set-to-set alignment and class-to-class alignment, to perform the cluster-to-cluster alignment.

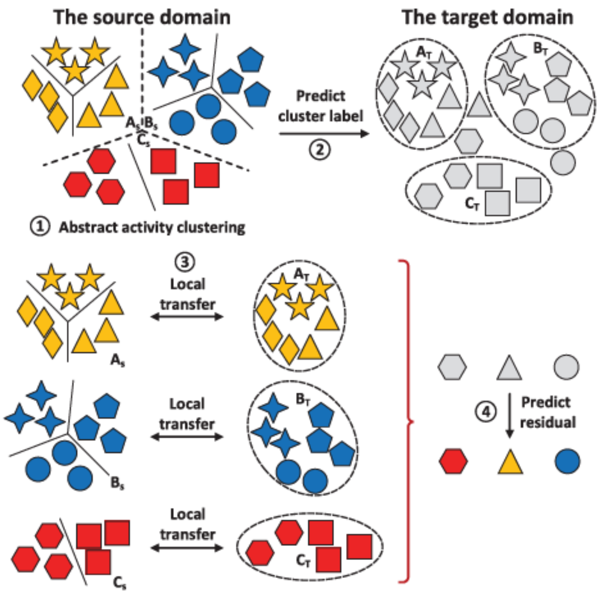


Fig. 3. Framework of LDA. ① Generate the abstract activity cluster labels (A_s, B_s, C_s) on the source domain artificially (LDA) or automatically (ACLDA). ② Predict the cluster labels for candidate samples (in the dotted circle) leaving out the residual samples (out of the dotted circle). ③ Perform local transfer and predict the class labels for the candidate samples. ④ Predict the class labels for the residual samples using the source domain and candidate samples.

III. Proposed Method

In this section, we first formulate the cross-domain HAR problem, then introduce the framework of LDA, and present how to modify LDA into ACLDA at last.

A. Problem Definition

We formulate the cross-domain HAR as an unsupervised domain adaptation problem, which means that the samples in the source domain are labeled, i.e., $\mathcal{D}_s = \{x_i, y_i\}_{i=1}^{m}$, but the target domian have no ground-truth labels, i.e., $\mathcal{D}_t = \{x_j\}_{j=1}^{n}$, where x_i and y_i are the feature vector and ground-truth label of ith sample; m and n are the sizes of $\mathcal{D}_s$ and $\mathcal{D}_t$. Furthermore, both domains share the same feature space $\mathcal{X}_s = \mathcal{X}_t$ and label space $\mathcal{Y}_s = \mathcal{Y}_t$, but have different conditional and marginal distributions $Q_s(y_s|x_s) \neq Q_t(y_t|x_t)$, $P_s(x_s) \neq P_t(x_t)$. The goal is to learn a classifier $f{:}x_t \mapsto y_t$ from source domain to classify the data in the target domain. In this article, we consider two types of cross-domain HAR: cross-person HAR, which means the data from different persons serve as different domains, and cross-position HAR, which means the data from different body parts of the same person serve as different domains.

B. LDA Framework

As shown in Fig. 3, LDA works as follows.

1) Group the source samples into several predefined abstract activity clusters based on the ground-truth class label. Therefore, each source sample is labeled at both the cluster level and class level.
2) Predict the cluster label for each target sample using a multiposition multiclassifier voting technique. Here, the samples that get voting consistency will be regarded as candidates and will be used to perform the local transfer in the following steps, but other samples are left out as residual samples.
3) Perform local transfer between source samples and target candidates with the same cluster label, and then predict the class labels for target candidates.
4) Use the source domain and target candidate samples to predict the class labels of the residual samples. The details of LDA are shown in the following sections.

C. Abstract Activity Clustering

Our method is a divide-and-conquer-like algorithm that breaks down a problem into subproblems. In preprocessing steps, we divide all individual activities into several high-level abstract activity clusters. The abstract activity cluster is a subset of activities that shares an identical abstract feature and distinguishes from other clusters. For example, all types of actions can be divided into dynamic actions (including walking, running) and static actions (including sitting, standing) according to whether this action involves movement. An abstract activity cluster consists of several individual classes, and an individual class belongs to only one cluster. We use "cluster of classes" to describe this relationship, so that one sample will have two labels, one is *class label* (such as sitting), the other is *cluster label* (such as a static activity). In this article, we use y_i and z_i to represent class labels and cluster labels for a sample x_i, respectively. The clusters can be artificially defined, or automatically generated based on the geometric distances between individual activities.

Why do we introduce the abstract activity cluster? Direct cross-person HAR at the individual activity level cannot get good accuracy (as shown in Fig. 2), but it can achieve acceptable performance at the activity cluster level because the distance between different clusters of the same person is larger than that between the same clusters of a different person [29]. If samples in both domains have the cluster label, we can make an intracluster knowledge transfer to improve the classification accuracy.

The next key question is how to predict the cluster label for the target domain. For the source domain, we can use the class labels to categorize samples into the corresponding abstract clusters. For the target domain, we introduce a voting technique to predict the cluster label. For the cross-position HAR, we use multiple base classifiers with the same input features to vote. For the cross-person HAR, we not only change the classifier but also change the sample features. Specifically, we build multiple classifiers for every IMU in different body positions. Therefore, every classifier can make a prediction individually and combine their predictions via voting. For example, if we build KNN and random forecast two classifiers for three IMUs on the torso, right arm, and right leg, there will be six classifiers participating in the voting.

First, we train M base classifiers $f_i(\cdot), i = 1, \ldots, M$ on the source domain $\{x_s, z_s\}$ to predict the cluster labels $\{\hat{z}_j\}$ for the target domain. M classifiers give M predicted cluster labels $[f_i(x_j)]_{i=1}^{M}$ for each target sample x_j. Then, we compute the most frequent value $\text{Mode}([f_i(x_t)]_{i=1}^{M})$. If $\text{Mode}([f_i(x_t)]_{i=1}^{M})$ is larger than $M/2$, the final result of majority voting on x_j will be $\text{Mode}([f_i(x_t)]_{i=1}^{M})$. Otherwise, x_j will be viewed as a residual sample without a credibly voted cluster label. Specifically, the target samples those have majority consistent voting results are called candidates x_{can}, and they are divided into predefined abstract clusters. Other samples are called residuals x_{res} and do not have cluster labels. We give the residuals a pseudolabel -1. Therefore, the predicted cluster label of a target sample x_t is

$$\hat{z}_j = \begin{cases} \text{Mode}([f_i(x_j)]_{i=1}^{M}), & \text{if major holds} \\ -1, & \text{orthewise.} \end{cases} \tag{1}$$

It is worth noting that the base classifiers can be any kind of classifier and the feature subset can be changed to adapt to different numbers of IMUs. In the ideal case, several IMUs are placing in different positions of the human body, so that we can perform major voting method on different classifiers of different positions. However, when there is only one IMU, we can do a little modification to make our methods sill work. We can separate the signals of the three sensors (accelerometer, gyroscope, and magnetometer) in one IMU, and create multiple classifiers for every sensor data.

D. Local Transfer

Via dividing the abstract action cluster in Section III-C, the similar activities in the source domain and the target domain are categorized in the same cluster, and each sample in the target domain either has a cluster label or becomes a residual sample. In this section, we transfer the knowledge between the same clusters in two domains, as called local transfer. In particular, we try to find a feature mapping that projects the original samples into a latent space where the data distributions in different domains are close to each other. The basic assumption of our method is that, in the same cluster, the samples have more common features and distribute in a similar subspace. Therefore, the local transfer can reduce the distribution difference more effectively.

To present our method, we first define some notations. $\mathcal{D}_s = \{x_1, \ldots, x_m\} \sim P$ and $\mathcal{D}_t = \{x_1, \ldots, x_n\} \sim Q$ denote the source and target domains, respectively, following the marginal distributions P and Q. Then, we define a kernel function $k(x_i, x_j)$ and the corresponding reproducing kernel Hilbert space (RKHS) is $\mathcal{H}$. The project function from the sample space to the RKHS is denoted by $\phi(x) = k(x, \cdot)$. We use maximum mean discrepancy (MMD) to measure the distribution distance of two distinct domains. The MMD between two domains can be presented as

$$\text{MMD}\left(\mathcal{D}_s, \mathcal{D}_t\right) = \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(x_j) \right\|_{\mathcal{H}}^{2}. \tag{2}$$

Different from [11], [30], we aim to minimize the sum of MMD distances between all the same clusters in the $\mathcal{D}_s$ and $\mathcal{D}_t$ instead of just minimizing the global MMD between two domains. Assuming that we predefine $n_{\mathcal{C}}$ clusters, the MMD between two clusters with the same cluster label from different domains will be

$$\text{MMD}\left(\mathcal{C}_s^k, \mathcal{C}_t^k\right) = \left\| \frac{1}{\left|\mathcal{C}_s^k\right|} \sum_{x_i \in \mathcal{C}_s^k} \phi(x_i) - \frac{1}{\left|\mathcal{C}_t^k\right|} \sum_{x_j \in \mathcal{C}_t^k} \phi(x_j) \right\|_{\mathcal{H}}^{2} \tag{3}$$

where $\mathcal{C}_s^k$ and $\mathcal{C}_t^k$ mean the kth cluster in the source domain and the target domain, respectively. It can also be written as

$$\text{MMD}\left(\mathcal{C}_s^k, \mathcal{C}_t^k\right) = \text{Tr}(K^k L^k) \tag{4}$$

where Tr means matrix trace. The computation is shown in

$$\begin{aligned} \text{Tr}(K^k L^k) &= \sum_{i,j=1}^{\left|\mathcal{C}_s^k\right| + \left|\mathcal{C}_t^k\right|} K_{ij}^k L_{ji}^k \\ &= \frac{1}{\left|\mathcal{C}_s^k\right|^2} \sum_{x_i, x_j \in \mathcal{C}_s^k} \left\langle \phi\left(x_i\right), \phi\left(x_j\right) \right\rangle \\ &\quad - \frac{1}{\left|\mathcal{C}_s^k\right| \left|\mathcal{C}_t^k\right|} \sum_{x_i \in \mathcal{C}_s^k, x_j \in \mathcal{C}_t^k} \left\langle \phi\left(x_i\right), \phi\left(x_j\right) \right\rangle \\ &\quad - \frac{1}{\left|\mathcal{C}_s^k\right| \left|\mathcal{C}_t^k\right|} \sum_{x_i \in \mathcal{C}_t^k, x_j \in \mathcal{C}_s^k} \left\langle \phi\left(x_i\right), \phi\left(x_j\right) \right\rangle \\ &\quad + \frac{1}{\left|\mathcal{C}_t^k\right|^2} \sum_{x_i, x_j \in \mathcal{C}_t^k} \left\langle \phi\left(x_i\right), \phi\left(x_j\right) \right\rangle \\ &= \left\| \frac{1}{\left|\mathcal{C}_s^k\right|} \sum_{x_i \in \mathcal{C}_s^k} \phi\left(x_i\right) - \frac{1}{\left|\mathcal{C}_t^k\right|} \sum_{x_j \in \mathcal{C}_t^k} \phi\left(x_j\right) \right\|_{\mathcal{H}}^{2} \end{aligned} \tag{5}$$

where K^k is the kernel matrix on $\mathcal{C}_s^k$ and $\mathcal{C}_t^k$, L^k is the coefficient matrix, i.e.,

$$L_{ij}^k = \begin{cases} \frac{1}{\left|\mathcal{C}_s^k\right|^2} \text{ if } x_i, x_j \in \mathcal{C}_s^k \\ \frac{1}{\left|\mathcal{C}_t^k\right|^2} \text{ if } x_i, x_j \in \mathcal{C}_t^k \\ -\frac{2}{\left|\mathcal{C}_s^k\right|\left|\mathcal{C}_t^k\right|} \text{ Otherwise} \end{cases}. \tag{6}$$

To this end, minimizing MMD distance is to find a kernel matrix that minimize $\mathrm{Tr}(K^k L^k)$, but solving K is an expensive semidefinite programming problem. Therefore, a mapping matrix $W \in \mathbb{R}^{D\times(|\mathcal{C}_s^k|+|\mathcal{C}_t^k|)}$ is introduced to present kernel matrix in a low-rank form as

$$\tilde{K}^k = K^k W W^T K^k \tag{7}$$

where D is the dimension of low-dimensional subspace. In the low-dimension kernel matrix, the kernel evaluation between two samples x_i, x_j is $\tilde{k}_{ij} = k_{x_i} W W^T k_{x_j}$, where k_{x_j} is the jth column of K. The W maps the empirical kernel presentation in the RKHS into a low-dimensional space, the embedding of a sample in the latent space is $W^T k_{x_j}$.

In the latent space, the MMD distance between two clusters can be rewritten as

$$\begin{aligned} \mathrm{MMD}\left(\mathcal{C}_s^k, \mathcal{C}_t^k\right) &= \mathrm{Tr}\left(\tilde{K}^k L^k\right) \\ &= \mathrm{Tr}(W^T K^k L^k K^k W). \end{aligned} \tag{8}$$

While minimizing the MMD distance in (8), we also need to preserve the data variance of the target domain to reduce overtransfer. The explicit variance can be formulated as $W^T K^k H K^k W$, where $H \in \mathbb{R}^{(|\mathcal{C}_s^k|+|\mathcal{C}_t^k|)\times(|\mathcal{C}_s^k|+|\mathcal{C}_t^k|)}$ is the centering matrix as defined in [31]. Now the objective function can be formulated as

$$\begin{aligned} &\min_{W^k} \sum_{k=1}^{K} \mathrm{Tr}((W^k)^T K^k L^k K^k W^k) + \mu^k \ \mathrm{Tr}((W^k)^T W^k) \\ &\text{s.t.} \ \ (W^k)^T K^k H^k K^k W^k = I \end{aligned} \tag{9}$$

where $\mathrm{Tr}((W^k)^T K^k L^k K^k W^k)$ indicates the MMD distance, the regular term $\mu^k \ \mathrm{Tr}((W^k)^T W^k)$ penalizes the complexity of mapping function, and μ^k is a tradeoff parameter. The constraint $(W^k)^T K^k H^k K^k W^k = I$ can preserve the inherent variance of the data. Ws are independent with each other in 9, so we can solve each W separately. Similar to the transfer component analysis and Fisher discriminant analysis, the solution of W is the D largest eignvectors of $(K^k L^k K^k + \mu I)^{-1} K^k H^k K^k$.

Algorithm 1 shows the overall steps of LDA. Line 1 and Line 2 are the abstract activity clustering step in LDA and ACLDA. Line 3 predicts the cluster label for the candidate samples in the target domain. Line 4 to Line 8 performs the local transfer for each cluster and predict the class label for the candidate samples. Line 9 predicts the class label for the residual samples. Finally, Line 10 concatenates the predicted labels for $\{\hat{y}_{\text{can}}^k\}$ and $\{\hat{y}_{\text{res}}^k\}$ and gets all predicted labels for $\mathcal{D}_t$.

Algorithm 1: (Autoclustering) Local Domain Adaptation.

```
  Input: Labeled data from source person $\mathcal{D}_s = \{x_s, y_s\}$,
   unlabeled data from target person $\mathcal{D}_t = \{x_t\}$, the
   number of dimensions $D$, the number of clusters $n_{\mathcal{C}}$.
  Output: The labels for the target person $\{y_t\}$.
 1: (LDA) Define the abstract activity clusters based on
    the source domain data using the expert knowledge,
    and get $\mathcal{C}_s^k = \{x_s^k, z_s^k\}, k = 1, ..., n_{\mathcal{C}}$.
 2: (ACLDA) Group the classes into $n_{\mathcal{C}}$ clusters with
    Algorithm 2.
 3: Perform multi-positions multi-classifiers voting on $\mathcal{D}_s$
    using Eq. (1) and get $\mathcal{C}_t^k = \{x_{\text{can}}^k, \hat{z}_{\text{can}}^k\}, k = 1, ..., n_{\mathcal{C}}$
    and $\{x_{\text{res}}\}$.
 4: for $k$=1:$n_{\mathcal{C}}$ do
 5:     Construct kernel matrix $K^k$ on $x_s^k$ and $x_{\text{can}}^k$, and
        compute the coefficient matrix $L^k$ and centering
        matrix $H^k$.
 6:     Eigen-decompose $(K^k L^k K^k + \mu I)^{-1} K^k H^k K^k$
        and take the $d$ largest eign-vectors to construct the
        transformation matrix $W^k$.
 7:     Map $x_s^k$ and $x_{\text{can}}^k$ into the same subspace using $W^k$.
 8:     Predict the class labels for $\{x_{\text{can}}\}$ to get $\{\hat{y}_{\text{can}}^k\}$.
 9: Predict the class labels for $x_{\text{res}}$ to get $\{\hat{y}_{\text{res}}^k\}$.
10: Concatenate $\{\hat{y}_{\text{can}}^k\}$ and $\{\hat{y}_{\text{res}}^k\}$ to get all predicted
    labels for $\mathcal{D}_t$
     Return: $\{\hat{y}_t\}$
```

E. Autoclustering LDA

To overcome the shortcoming that LDA needs human knowledge, we introduce a clustering technique to generate the activity clusters automatically. We refer this version of LDA to as ACLDA. Algorithm 2 presents the details of the clustering step of ACLDA.

In the first step, we compute the distance between two categories using the average distance between the samples in two categories as

$$\mathrm{Dist}(\mathcal{S}_i, \mathcal{S}_j) = \frac{2\times\sum\left\|x_{\mathcal{S}_i} - x_{\mathcal{S}_j}\right\|_2}{|\mathcal{S}_i| + |\mathcal{S}_j|} \tag{10}$$

where $\mathcal{S}_i$ and $\mathcal{S}_j$ are the sample set of the ith and jth class, $x_{\mathcal{S}_i}$ and $x_{\mathcal{S}_j}$ are samples in $\mathcal{S}_i$ and $\mathcal{S}_j$.

In the second step, we use aggregative clustering to group the classes. The aggregative clustering aims to contract a hierarchy of clusters from bottom to up [32]. It only takes the closet pairs of objects or low-level clusters into a higher level group, until all the objects are linked together into a single tree, and then cuts the tree into the specified clusters. Note that other clustering methods can take replace of aggregative clustering here, such as spectral clustering. Once the activity clusters are generated using Algorithm 2, the subsequent algorithm process of ACLDA is the same as LDA.

Unlike LDA that involves human knowledge, ACLDA generates the clusters according to the geometric distance between samples, so each cluster may not have a clear abstract meaning. The expert can only use his knowledge to define the cluster, so

Algorithm 2: Group the Classes Into Clusters.

```
Input: Labeled data from the source persons
  $\mathcal{D}_s = \{x_s, y_s\}$, The number of clusters $n_\mathcal{C}$.
Output: The cluster labels for the source persons $z_s$.
Compute the mean of the pairwise distance between
  any two classes of samples in the source domain using
  Eq. (10), and get the distance matrix $D$ between
  classes.
Perform aggregative clustering or spectral clustering
  based on $D$ to get the cluster labels $z_s$ for the source
  domain.
Return: $\mathcal{C}_s = \{x_s, \hat{z}_s\}$
```

the clustering results keep the same for all persons. However, different persons usually have their action characteristics. Besides, the best number of clusters for each person may also be different. To this end, the clustering process is person-dependent and dataset-dependent. Beyond defining the stationary cluster, ACLDA can automatically identify suitable clusters and search for the best number of clusters. The number of cluster $n_\mathcal{C}$ is a key parameter for ACLDA. However, the range of $n_\mathcal{C}$ is small, using a validation dataset and optimizing classification accuracy on it can suffice to find good $n_\mathcal{C}$.

IV. Experiments

In this section, we evaluate LDA and ACLDA on two public datasets: the daily and sports activities dataset (DSADS) [8] and the physical activity monitoring dataset (PAMAP2) [33]. First, we present details of the experimental setting, then discuss the experimental results, and analyze the parameter sensitivity at last.

A. *Experiment Setting*

Following the related works [9], [20], [34], we conduct two types of cross-domain HAR tasks: 1) cross-person HAR, where different domains contain data collected from different persons, 2) cross-position HAR, which means that different sensor positions belong to different domains.

The DSADS and PAMAP2 are two widely-used HAR datasets. DSADS involves 19 activities, each performed by eight objects. The data are collected from five IMUs mounted on the torso, right arm, left arm, right leg, and left leg. PAMAP2 contains 18 different physical activities performed by nine subjects wearing three IMUs (arm, chest, and leg). In both datasets, the subjects perform the activities in their style without specific restrictions, so both datasets are suitable to evaluate a cross-person HAR algorithm. Additionally, both datasets contain more than one sensor position, so we can use it to do cross-position HAR experiments. Note that for the PAMAP2 dataset, we select a subset containing six persons and nine activities because of the data missing.

Similar to existing literatures [14], [35], we classify the activity data based on manual features instead of the raw time-series data. Specifically, we standardize the data using z-score and combine the data from three axes of one sensor together using $\sqrt{x^2+y^2+z^2}$ to reduce the interference caused by different installation attitudes of sensors. Then, we segment the data with a 5s-window-size, 3s-step-length moving window. Finally, 27 features are extracted for each data window with the toolbox *Seglearn* [36].

Here are some implementation details of LDA and ACLDA. Table I shows the definitions of abstract activity clusters used in LDA. According to the motion and environment contexts, nineteen classes in DSADS are grouped into five clusters, while nine classes in PAMAP2 are grouped into three clusters. For ACLDA, the number of clusters $n_\mathcal{C}$ is chosen by grid search method, the best $n_\mathcal{C}$ for DSADS dataset and PAMAP2 dataset is 13 and 6. In the multiclassifier voting step, we use three classifiers, 1-NN, SVM, and random forest. 1-NN classifier is used as the final classifier to predict the class label of candidates and residuals.

We construct a comparative study to illustrate the effectiveness of LDA. The comparison methods include baseline methods, i.e., 1-nearest neighbors (1-NN), support vector machine (SVM), and random forests (RF); dimension-reduction-based method, i.e., principal component analysis (PCA); domain adaptation methods, i.e., transfer component analysis (TCA), which aims to align the marginal distribution of both domains [30], subspace alignment (SA), which maps both domains into a subspace spanned by eignvectors [37], and stratified transfer learning (STL), which performs intraclass marginal distribution [38].

For the random forest, the number of trees is 20. PCA, TCA, SA, and STL are all dimensionality reduction-based methods, so we test and report their performances under different dimension settings, and 1-NN serves the final classifier. To provide robust experiment results, we repeat all the experiments 20 times by randomly selecting the persons in both domains and report the average results. We use the classification accuracy and F1 score of the target domain as the evaluation metric.

B. *Results and Discussion*

Tables II and III report the classification results from the proposed methods and comparison methods on the cross-person and cross-position HAR tasks. The tasks are named by "source domain → target domain". For example, "One → One" means a cross-person HAR task that the source domain is from one person, and the target domain is from another person; "RA → LA" means a cross-position HAR task that the source domain is the data from the right arm and the target domain is from the left arm. All the tasks are repeated 20 times with randomly selecting source persons and target persons. We also report the P-values between the results of ACLDA and STL to show the statistical significance.

Table II shows that LDA and ACLDA outperform other methods in most cases on cross-person HAR experiments. Meanwhile, ACLDA achieves better accuracy than LDA. Compared with the direct classification methods (1-NN, 3-NN, and RF), LDA and ACLDA obtain higher accuracy in most cases, which indicates that the proposed methods can perform a stable positive

TABLE I
DEFINITIONS OF ABSTRACT ACTIVITY CLUSTERS IN LDA

Data set	Cluster	# classes	Class labels in the cluster (Class label ID)
PAMAP2	Static activities	3	Lying (A1); Sitting (A2); Standing (A3).
	Locomotion activities	4	Walking (A4); Running (A5); Cycling (A6); Nordic walking (A7).
	Upstairs/Downstairs	2	Ascending stairs (A8); Descending stairs (A9).
DSADS	Static activities	5	Sitting (A1); Standing (A2); Lying on back (A3); Lying on right side (A4); Standing in an elevator still (A7).
	Upstairs/Downstairs	3	Ascending stairs (A5); Descending stairs (A6); Moving around in an elevator (A8).
	Locomotion activities	4	Walking in a parking lot (A9); Walking on a treadmill with a speed of 4 km/h in flat (A10); Walking on a treadmill with a speed of 4 km/h in 15 deg inclined positions (A11); Running on a treadmill with a speed of 8 km/h (A12).
	Sports with fitness equipment	4	Exercising on a stepper (A13); Exercising on a cross trainer (A14); Cycling on an exercise bike in horizontal position (A15); Cycling on an exercise bike in vertical position (A16).
	Sports without fitness equipment	3	Rowing (A17); Jumping (A18); Playing basketball (A19).

TABLE II
ACCURACY AND F1 SCORES FROM LDA, ACLDA AND THE BASELINE METHODS ON CROSS-PERSON HAR TASKS

Method		1NN		SVM		RF		PCA		TCA		SA		STL		LDA		ACLDA		P-value	
Data set	Task	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DSADS	One→One	57.91	55.89	58.15	55.98	57.09	53.19	57.88	55.89	58.59	56.76	57.83	55.83	60.63	58.45	63.11	60.67	**64.27**	**61.94**	<0.01	<0.01
	Two→Two	62.65	61.66	64.23	63.23	66.98	64.87	62.63	61.64	62.59	61.07	62.66	61.68	68.43	66.95	71.48	70.60	**74.23**	**72.30**	<0.01	<0.01
	Three→Three	68.02	67.70	69.87	69.56	68.04	67.75	68.01	67.79	68.70	67.72	68.02	67.80	71.29	70.60	74.98	74.73	**76.67**	**75.72**	<0.01	<0.01
	One→Two	56.92	55.10	57.90	56.83	52.94	50.13	57.28	56.33	57.54	56.69	57.27	56.32	62.10	60.18	64.05	62.78	**64.59**	**63.02**	<0.01	<0.01
	One→Four	58.30	57.48	59.44	58.78	59.61	57.35	58.32	57.48	58.64	57.97	58.35	47.49	**63.29**	**61.77**	63.15	61.69	62.89	60.27	<0.05	<0.05
	One→Six	60.31	59.77	61.32	60.79	59.50	57.73	60.29	59.76	60.33	59.78	60.45	59.90	66.31	65.65	67.06	66.41	**68.12**	**67.33**	<0.01	<0.01
PAMAP2	One→One	68.44	66.81	70.23	**69.39**	66.82	63.10	68.54	65.26	69.15	66.22	68.51	65.23	67.91	62.85	69.30	63.91	**72.23**	69.23	<0.01	<0.01
	Two→Two	72.91	71.08	73.93	72.72	74.55	73.76	72.97	71.10	73.27	71.36	72.92	71.10	76.89	74.23	**77.11**	**76.96**	75.45	74.50	<0.05	<0.05
	Three→Three	75.10	73.58	76.66	75.00	73.10	72.98	75.11	73.67	75.29	73.57	75.11	73.59	76.61	74.76	78.75	**77.88**	**79.79**	77.68	<0.01	<0.01
	One→Two	69.15	65.96	70.77	67.07	68.11	65.81	69.71	65.99	69.37	66.36	69.18	66.02	68.25	65.85	72.98	71.34	**73.38**	**72.66**	<0.01	<0.01
	One→Four	68.45	68.42	69.58	66.59	67.41	64.89	68.47	65.98	68.98	68.09	69.43	66.78	68.97	65.69	70.67	67.63	**73.44**	**72.02**	<0.01	<0.01
	One→Six	68.76	66.78	70.34	67.23	68.22	67.02	68.23	66.03	69.02	67.93	68.78	66.73	69.34	67.80	72.72	71.21	**74.07**	**73.11**	<0.01	<0.01
Average		65.57	64.18	66.86	65.26	65.19	63.21	65.62	63.91	65.96	64.46	65.71	63.20	68.33	66.23	70.44	68.81	**71.60**	**69.98**	-	-

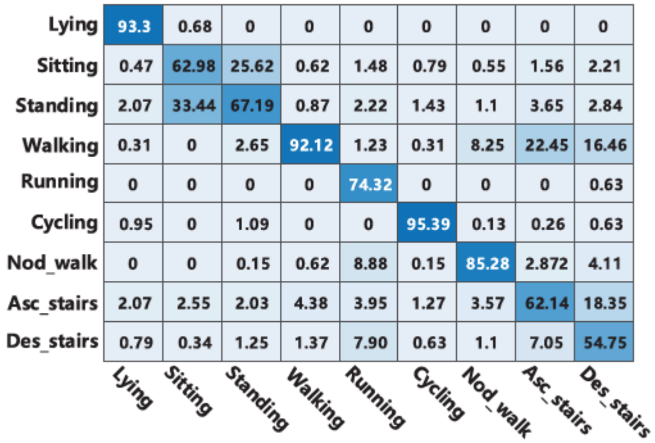
TABLE III
ACCURACY AND F1 SCORES FROM LDA, ACLDA AND THE BASELINE METHODS ON CROSS-POSITION HAR TASKS[1]

Method		1NN		SVM		RF		PCA		TCA		SA		STL		LDA		ACLDA		P-value	
Data set	Task	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DSADS	RA→LA	57.27	55.95	59.25	57.08	62.19	60.82	56.30	55.61	57.61	56.15	57.73	55.09	64.70	63.03	65.32	**64.75**	**65.77**	64.42	<0.05	<0.05
	RL→LL	63.36	61.79	64.37	63.81	62.98	61.30	63.35	61.77	64.21	62.06	63.40	61.65	**69.86**	**68.26**	67.25	67.06	68.54	66.70	<0.05	<0.05
	RA→T	42.56	41.01	43.65	42.66	44.67	43.81	42.35	40.52	42.75	40.24	42.59	40.20	44.02	42.61	45.36	44.39	**46.77**	**44.04**	<0.01	<0.01
PAMAP2	L→C	38.49	37.41	41.25	39.36	40.67	39.70	37.57	36.54	38.88	37.18	38.72	37.29	41.39	41.18	43.25	42.51	**44.04**	**42.69**	<0.01	<0.01
	C→A	35.71	33.89	34.25	31.49	36.73	35.51	36.32	34.42	37.45	36.29	37.48	36.32	39.48	37.59	39.05	37.15	**40.86**	**38.94**	<0.05	<0.05
	A→L	34.18	32.32	35.08	34.72	37.21	36.39	34.57	33.48	37.86	35.65	35.13	32.45	39.02	37.32	40.78	39.26	**41.64**	**40.01**	<0.01	<0.01
Average		45.26	43.72	46.30	44.85	47.40	46.25	45.07	43.72	46.46	44.59	45.84	43.83	49.74	48.33	50.16	49.18	**51.27**	**49.46**	-	-

transfer in the cross-domain HAR problem. On the cross-person HAR problem, Compared with the best comparison method STL, LDA shows 2.11% improvement on the accuracy and 2.58% improvement on the F1 score. This indicates that transfer learning at a cluster level is effective for the cross-person HAR problem. Moreover, ACLDA performs 1.16% better than LDA on the accuracy, and 1.17% better on the F1 score. This verifies the effectiveness of automatic clustering step of ACLDA. Table III also shows the similar results on the cross-position HAR. Another interesting finding is that the performances of all methods on cross-position HAR are lower than those on the cross-person HAR. This is because less sensor data are used in the cross-position HAR.

TCA, SA, STL, LDA, and ACLDA are all based on subspace learning, which aim to minimize the distribution discrepancy between domains by projecting the original features into a subspace. TCA and SA aim to minimize the set-to-set distribution discrepancy, but their performances may decline when the cross-domain divergence is large. Conversely, STL performs based

[1] For the task names on the DSADS, RA, LA, RL, LL, and T stand for right arm, left arm, right leg, left leg, and torso, respectively. Similarly, L, C, A stand for leg, chest and arm on the PAMAP2.



	Lying	Sitting	Standing	Walking	Running	Cycling	Nod_walk	Asc_stairs	Des_stairs
Lying	93.3	0.68	0	0	0	0	0	0	0
Sitting	0.47	62.98	25.62	0.62	1.48	0.79	0.55	1.56	2.21
Standing	2.07	33.44	67.19	0.87	2.22	1.43	1.1	3.65	2.84
Walking	0.31	0	2.65	92.12	1.23	0.31	8.25	22.45	16.46
Running	0	0	0	0	74.32	0	0	0	0.63
Cycling	0.95	0	1.09	0	0	95.39	0.13	0.26	0.63
Nod_walk	0	0	0.15	0.62	8.88	0.15	85.28	2.872	4.11
Asc_stairs	2.07	2.55	2.03	4.38	3.95	1.27	3.57	62.14	18.35
Des_stairs	0.79	0.34	1.25	1.37	7.90	0.63	1.1	7.05	54.75

Fig. 4. Confusion matrix (%) of ACLDA on One → One tasks on the PAMAP2 dataset. Row labels refer to the real classes and column labels refer to the predicted classes.

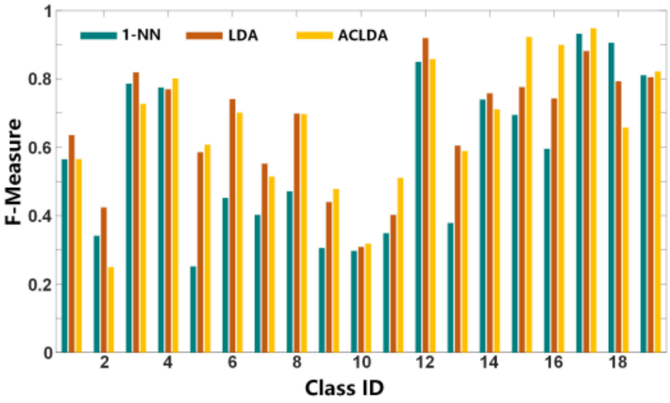


Fig. 5. F-Measure of One → One HAR tasks on the DSADS dataset. Class ID can be found in Table I.

on the predicted soft labels for the target domain and achieves better results than TCA and SA. However, STL heavily relies on the accuracy of soft labels. Differently, LDA and ACLDA perform cluster-to-cluster distribution alignment. Compared with set-to-set alignment, LDA reduces the distribution discrepancy at a finer granularity; compared with class-to-class alignment, LDA gives more accurate soft labels and guarantees the number of candidates at the cluster level. These characteristics lead to the best performance among compared subspace-based DA methods.

To analyze the performance of LDA and ACLDA further, we present the confusion matrix and F1 score (F measure) bars in Fig. 4 and Fig. 5, respectively. Fig. 4 shows the ACLDA confusion matrix of One → One tasks on the PAMAP2 dataset. We can find that ACLDA can achieve an accuracy more than 90% for lying, walking, and cycling. However, there is some misclassification between sitting and standing. Besides, some ascending and descending samples are classified as walking. These errors are the primary source of overall errors. In addition, there are some samples from all kinds of dynamic activities that are misclassified as standing, which can be explained by the fact that standing often happens during dynamic activities. The source domain selection technique or a more discriminate feature extraction method can reduce the misclassification. Fig. 5 shows the F1 score of each class in the DSADS dataset. The average F1 score of 1-NN, LDA, and ACLDA are 0.574, 0.666, and 0.662, respectively. LDA and ACLDA improve the classification accuracy for most classes. For example, for activities A5, A6, A8, A15, and A16, the F1 scores of LDA and ACLDA are 0.24 and 0.26 higher than that of the baseline. However, LDA and ACLDA may also perform badly for a few activities, such as activity A18.

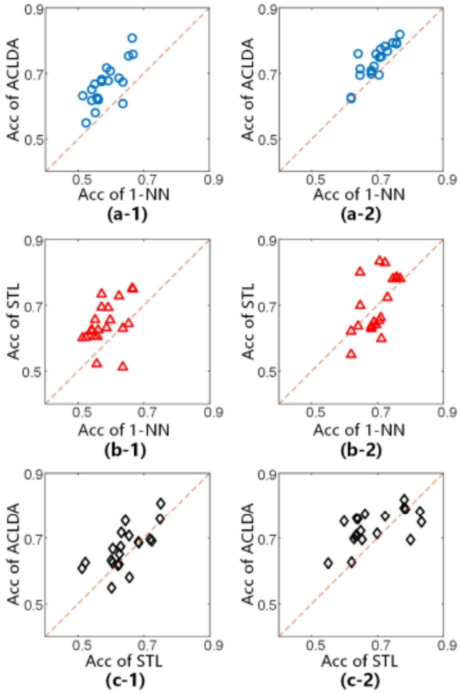


Fig. 6. Comparison of the accuracy of ACLDA (proposed), STL (best comparison) and 1-NN (baseline) on 20 random experiments on two datasets. The left column and right column show the accuracy on DASDS and PAMAP2. A marker at the top left of the dashed line indicates an experiment that the method of vertical axis outperforms the method of horizontal axis.

To verify the empirical robustness of proposed methods, we compare the accuracy of ACLDA (proposed), STL (best comparison) and 1-NN (baseline) on 20 random One → One experiments. Fig. 6 shows the result. Fig. 6(a-1) and (a-2) shows that, among 20 experiments on each dataset, there is only one experiment in which the accuracy of ACLDA is lower than 1-NN. However, as shown in Fig. 6(b-1) and (b-2), STL encounters 4 and 13 negative transfer experiments on the DSADS and PAMAP2. Fig. 6(c-1) and (c-2) compare the performance of ACLDA and STL directly, we can see that ACLDA outperforms STL in 31 experiments among 40 experiments. These comparisons indicate that ACLDA can achieve stable positive transfer and improve the accuracy of cross-position HAR in most cases.

C. Parameter Analysis

The number of reduction dimensions is a free parameter for LDA and other comparison DA methods, so we evaluate the impact of reduction dimensions by varying it from 20 to 200.

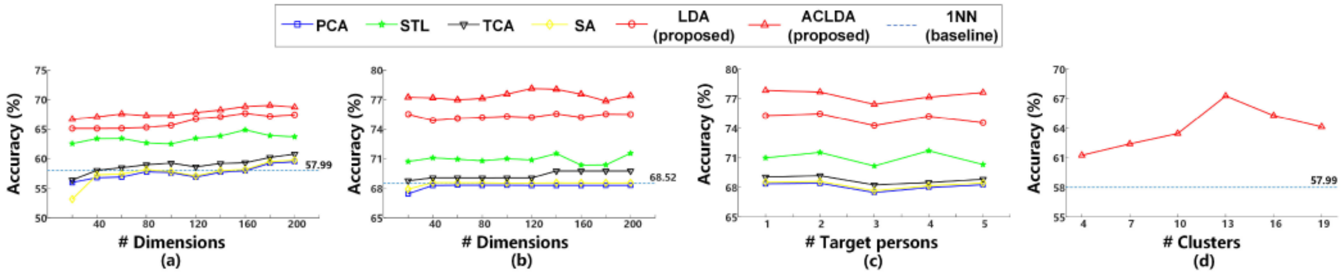


Fig. 7. Parameters analysis. (a) and (b) Accuracy of proposed methods and comparison methods under different dimensions on the DSADS dataset and the PAMAP2 dataset. (c) Accuracy under different numbers of target persons on the PAMAP2 dataset. (d) Accuracy of ACLDA on the DSADS dataset with different numbers of clusters $n_{\mathcal{C}}$.

Fig. 7(a) and (b) show the accuracy with different dimensions on the DSADS and PAMAP2. On both datasets, ACLDA achieves the best accuracy under every dimension setting, which indicates that our methods are robust to the dimension for the cross-person HAR tasks. However, TCA, SA, and PCA may be worse than baseline methods when the dimension is low. For example, SA and PCA need more than 180 dimensions to realize positive transfer, which indicates that PCA cannot find "the cross-person feature". Another interesting finding is that the performance of PCA and SA is very similar because SA essentially performs PCA on both domains to implement knowledge transfer.

To show the impact of the target sample size, we plot the accuracy curves under different target persons on the PAMAP2 dataset in Fig. 7(c). We choose the data of one person as a source, and to predict more than one persons' activities separately. We find that ACLDA still achieves the best performance among all methods, with an average accuracy of 77.34%. Besides, ACLDA's worst accuracy is only 1.41% worse than its best, and the accuracy does not fall when the number of target persons increases. These results indicate that ACLDA is not sensitive to the size of the target domain.

The number of clusters $n_{\mathcal{C}}$ is another free parameter of ACLDA. Fig. 7(d) shows the accuracy curve against the number of clusters $n_{\mathcal{C}}$ on the DSADS dataset. ACLDA achieves positive transfer in all cases when $n_{\mathcal{C}}$ is varying. It achieves the highest accuracy 67.28% when $n_{\mathcal{C}} = 13$, but when $n_{\mathcal{C}}$ is small as 3 or large as 19, the accuracy falls down to 61.28% and 64.14%, respectively. The experiments on the PAMAP2 dataset also obtain similar results. This can be explained by the fact that, if $n_{\mathcal{C}}$ is small, the cluster label prediction will be accurate, but the number of classes and samples in each cluster will be largely resulting in a poor local transfer. On the contrary, when $n_{\mathcal{C}}$ is equal to the number of classes, it results in a bad prediction for the candidate samples, so that the subsequent local transfer may be performed on between wrong classes.

To assess the time complexity of our methods, we compare the running time on a random Three → Three cross-person HAR tasks on PAMAP2 dataset. The running platform is MATLAB on a personal laptop with Intel Core i7-4710MQ CPU @ 2.50 GHz and 8.00 GB RAM. The running time is shown in Table IV. STL, LDA (ours), and ACLDA (ours) have significantly greater time complexity than other methods, because these three methods involve ensemble process. ACLDA takes more time than LDA is because ACLDA need to predict the cluster labels for source domain. To reduce the time complexity of our method, we can use simpler and fewer base classifiers.

TABLE IV
TIME COMPLEXITY COMPARISON (S)

Data set	KNN	SVM	RF	PCA	TCA	SA	STL	LDA	ACLDA
PAMAP2	0.24	4.27	1.63	0.19	0.58	0.30	10.66	12.71	13.67
DSADS	2.67	36.38	3.97	1.22	4.00	1.78	53.68	57.18	63.88

V. CONCLUSION

This article proposes LDA and ACLDA, two new approaches for cross-domain HAR. LDA(ACLDA) first groups the activities into abstract clusters, then maps the original features into a low-dimensional subspace where the MMD distance between two clusters with the same label from different domains is minimized. Finally, LDA(ACLDA) predicts the target class labels in the subspace. Compared with set-to-set DA, LDA(ACLDA) takes advantage of the soft labels of candidates from the target domain, so align the distribution in finer granularity. Compared with class-to-class DA, LDA(ACLDA) gives more confident soft labels, so is more robust. Cross-person and cross-position HAR experiments on two public datasets demonstrate the superior of LDA(ACLDA). Here are two possible future research directions. On the one hand, we will combine our method with deep neural networks into a united framework, which can take the advantage of neural networks to perform feature extraction. On the other hand, it is a good direction to generalize our method beyond HAR problems to other domain adaptation problems.

REFERENCES

[1] Z. Wang, N. Yang, M. Guo, and H. Zhao, "Human-human interactional synchrony analysis based on body sensor networks," *IEEE Trans. Affect. Comput.*, vol. 10, no. 3, pp. 407–416, Jul.–Sep. 2019.

[2] S. Gaglio, G. L. Re, and M. Morana, "Human activity recognition process using 3-d posture data," *IEEE Trans. Hum.-Mach. Syst.*, vol. 45, no. 5, pp. 586–597, 2014.

[3] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From activity recognition to intention recognition for assisted living within smart homes," *IEEE Trans. Hum.-Mach. Syst.*, vol. 47, no. 3, pp. 368–379, Jun. 2017.

[4] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, "Enabling effective programming and flexible management of efficient body sensor network applications," *IEEE Trans. Hum.-Mach. Syst.*, vol. 43, no. 1, pp. 115–133, Jan. 2013.

[5] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, "Multi-user activity recognition: Challenges and opportunities," *Inf. Fusion*, vol. 63, pp. 121–135, Nov. 2020.

[6] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *Proc. 3rd Int. Conf. Crowd Sci. Eng*. ACM, 2018, Art. no. 16.

[7] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognitiona survey," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discov.*, vol. 8, no. 4, 2018, Paper e1254.

[8] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, 2010.

[9] J.-H. Hong, J. Ramos, and A. K. Dey, "Toward personalized activity recognition systems with a semipopulation approach," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 1, pp. 101–112, Feb. 2016.

[10] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, and K. Tang, "Kernel fusion based extreme learning machine for cross-location activity recognition," *Inf. Fus.*, vol. 37, no. 9, pp. 1–9, 2017.

[11] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction." in *Proc. AAAI*, 2008, vol. 8, pp. 677–682.

[12] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervasive Mobile Comput.*, vol. 57, no. 7, pp. 1–13, 2019.

[13] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit.*, vol. 108, 2020, Art. no. 107561.

[14] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, 2014, Art. no. 33.

[15] 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 *Proc. Int. Workshop Ambient Assist. Living*, Springer, 2012, pp. 216–223.

[16] A. Jordao, L. A. B. Torres, and W. R. Schwartz, "Novel approaches to human activity recognition based on accelerometer data," *Signal, Image Video Process.*, vol. 12, no. 7, pp. 1–8, 2018.

[17] K. Nakano and B. Chakraborty, "Effect of dynamic feature for human activity recognition using smartphone sensors," in *Proc. IEEE 8th Int. Conf. Awareness Sci. Technol.,* 2017, pp. 539–543.

[18] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE Trans. Cybern.*, to be published.

[19] L. Li, H. He, J. Li, and G. Yang, "Adversarial domain adaptation via category transfer," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[20] F. J. O. Morales and D. Roggen, "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations," in *Proc. ACM Int. Symp. Wearable Comput.*, 2016, pp. 92–99.

[21] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, and G. Tröster, "Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system," *Mach. Learn.*, vol. 1, no. 7, pp. 73–78, 2011.

[22] Y.-T. Chiang, C.-H. Lu, and J. Y.-j. Hsu, "A feature-based knowledge transfer framework for cross-environment activity recognition toward smart home applications," *IEEE Trans. Hum.-Mach. Syst.*, vol. 47, no. 3, pp. 310–322, Jun. 2017.

[23] D. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Catalonia, Spain, 2011, pp. 1962–1967.

[24] C.-H. Lu, Y.-C. Ho, Y.-H. Chen, and L.-C. Fu, "Hybrid user-assisted incremental model adaptation for activity recognition in a dynamic smart-home environment," *IEEE Trans. Hum.-Mach. Syst.*, vol. 43, no. 5, pp. 421–436, Sep. 2013.

[25] V. S. Handiru and V. A. Prasad, "Optimized bi-objective EEG channel selection and cross-subject generalization with brain–computer interfaces," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 6, pp. 777–786, Dec. 2016.

[26] J. Zhao, L. Li, F. Deng, H. He, and J. Chen, "Discriminant geometrical and statistical alignment with density peaks for domain adaptation," *IEEE Trans. Cybern.*, to be published.

[27] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Netw.*, vol. 53, pp. 1–7, 2014.

[28] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 2545–2250.

[29] F. Azhar and C.-T. Li, "Hierarchical relaxed partitioning system for activity recognition," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 784–795, Mar. 2017.

[30] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[31] J. I. Marden, *Analyzing and Modeling Rank Data*. Boca Raton, FL, USA: CRC Press, 2014.

[32] R. Sibson, "Slink: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.

[33] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proc. 5th Int. Conf. PErvasive Technol. Related to Assistive Environ.*, ACM, 2012, Art. no. 40.

[34] D. H. Hu, V. W. Zheng, and Q. Yang, "Cross-domain activity recognition via transfer learning," *Pervasive Mobile Comput.*, vol. 7, no. 3, pp. 344–358, 2011.

[35] L. Hu, Y. Chen, J. Wang, C. Hu, and X. Jiang, "Okrelm: Online kernelized and regularized extreme learning machine for wearable-based activity recognition," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 9, pp. 1577–1590, 2018.

[36] D. M. Burns and C. M. Whyne, "Seglearn: A python package for learning sequences and time series," *J. Mach. Learn. Res.*, vol. 19, no. 11, 2018.

[37] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2960–2967.

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