This is the peer reviewd version of the followng article:

ContrasGAN: Unsupervised domain adaptation in Human Activity Recognition via adversarial and contrastive learning / Rosales Sanabria, A., Zambonelli, F., Dobson, S., Ye, J.. - In: PERVASIVE AND MOBILE COMPUTING. - ISSN 1574-1192. - 78:(2021), pp. 1-30. [10.1016/j.pmcj.2021.101477]

*Terms of use:*

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

16/11/2026 17:07

(Article begins on next page)

# ContrasGAN: Unsupervised Domain Adaptation in Human Activity Recognition via Adversarial and Contrastive Learning

Andrea Rosales Sanabria[a], Franco Zambonelli[b], Simon Dobson[a], Juan Ye[a]

[a]*School of Computer Science, University of St Andrews, UK*
[b]*Dipartimento di Scienze e Metodi dell'Ingegneria, Universita' di Modena e Reggio Emilia*

**Abstract**

Human Activity Recognition (HAR) makes it possible to drive applications directly from embedded and wearable sensors. Machine learning, and especially deep learning, has made significant progress in learning sensor features from raw sensing signals with high recognition accuracy. However, most techniques need to be trained on a large labelled dataset, which is often difficult to acquire. In this paper, we present ContrasGAN, an unsupervised domain adaptation technique that addresses this labelling challenge by transferring an activity model from one labelled domain to other unlabelled domains. ContrasGAN uses bi-directional generative adversarial networks for heterogeneous feature transfer and contrastive learning to capture distinctive features between classes. We evaluate ContrasGAN on three commonly-used HAR datasets under conditions of cross-body, cross-user, and cross-sensor transfer learning. Experimental results show a superior performance of ContrasGAN on all these tasks over a number of state-of-the-art techniques, with relatively low computational cost.

*Keywords:* Human activity recognition, unsupervised domain adaptation, GAN, contrastive loss

## 1. Introduction

Human activity recognition (HAR) is the recognition of user behaviours (*e.g.*, making a meal or performing personal hygiene tasks) from a series of observations collected from a set of sensors; for example, inferring a user is jogging or sitting based on the data collected from the acelerometer and gyroscope embedded in their smart phone. With pervasive availability of smart sensing technologies such as wearables and environmental sensors [30], we have witnessed increasing deployment of HAR systems in real-world environments [8]. A large number of machine learning techniques have been proposed to improve the accuracy of activity recognition [5, 11, 33, 49]. However, most require a

large amount of well-annotated data to train a sophisticated model. Annotations are either acquired via people's self-reporting, or video-recording [39]. Self-reporting requires a lot of time and attention from users, and is not always feasible especially with disadvantaged users; for example, the users suffering dementia. Video-recording incurs privacy risks and requires extra effort in annotating videos into activity labels. Neither approach scales to a large number of users.

Unsupervised domain adaptation (UDA) is emerging as an effective approach of tackling this annotation scarcity challenge. Domain adaptation is to use a classifier learned on one domain (called *source*) to predict labels on the other domain (called *target*) [26]. UDA is a subfield of domain adaptation, where labels on the target domain are not used during the adaptation training process. It aims to generalise and transfer a model learnt from a well-labelled source domain to a new, unlabelled target domain by mitigating the domain shift in data distribution [15]. An increasing number of UDA techniques have been applied in HAR; however, most of them work between domains that share the same feature space; for example, transferring the learn models between users [44] or between sensor wearing positions [32]. This might not be sufficient as most of HAR systems have different sensor deployments (resulting in highly heterogeneous feature spaces between the source and target domains) and host users with diverse lifestyles (leading to disparate prediction functions).

To tackle these challenges, we propose ContrasGAN, an unsupervised domain adaptation technique between domains in heterogeneous feature spaces; that is, domains with different sensing technologies or sensor deployment. For example, we imagine that the source domain is a well-controlled lab setting while the target domain is a real-world deployment. ContrasGAN will make it possible to adapt an activity model curated in the lab to a large number of real-world settings without the need of collecting any additional labels.

ContrasGAN leverages bi-directional generative adversarial networks (Bi-GAN) [56] from the field of image-to-image translation, where they are used from (for example) generating one type of image (such as sketches) from another (such as photographs). Bi-GAN does not make any assumption on the dimensions and distributions of feature spaces between source and target domains, which is desirable to tackle the heterogeneity challenge of feature spaces.

However, Bi-GAN alone is insufficient for HAR in that it focuses on the translation between individual samples and does not consider the *classes* from which these samples are drawn. This

may result in poor adaptations:

- instances of different classes may be aligned incorrectly due to the imbalanced class distribution of most HAR datasets; and
- there exist many sub-optimal solutions near the decision boundary that might overfit the source data but generalise poorly for the target domain: that is, some activities can have less distinctive patterns when they have fine difference in distributions. For example, as presented in confusion matrices in Figure 4a and 4b, it is difficult to separate sitting and standing activities in B-GAN (with accuracy of 0.54 and 0.71), while both are well recognised in ContrasGAN (with accuracy of 0.95 and 0.99).

To tackle these problems, we add contrastive learning during the adaptation with the goal to minimise the intra-class discrepancy and maximise the inter-class margin. The intra-class domain discrepancy is minimised to draw closer the feature representations of samples within a class so that the samples of the same class are close. The inter-class domain discrepancy is maximised to push the representations of each other further away from the decision boundary so that the samples from difference classes are well separated. The intra-class and inter-class discrepancies are jointly optimised to improve the adaptation performance. Contrastive learning is to learn a feature space where similar samples are put close to each other while dissimilar ones are pushed apart. With it, we hypothesise that ContrasGAN can better discriminate samples from different class labels and lead to more class-discriminative adaptation.

Looking into domain adaptation in HAR, we have identified the following three main tasks:

**Cross-body** Transferring an activity model from one wearing body position to another; *e.g.*, from left leg to right leg. It is motivated from *wearing diversity* of wearable sensors [10] in that users tend to change where to put their sensors or wearables depending on their preference and their current activities. It is desirable to transfer an activity model learnt on one body position to another; for example, being able to recognise the 'walking' activity when a sensor is worn on a user's leg from the activity model that is learned when the sensor is worn on the user's arm. This can reduce annotation cost and improve recognition accuracy and robustness to the variability of wearing positions.

**Cross-user** Transferring an activity model from one user to another. This tackles a problem where there are a large number of users for a HAR system but it is impractical or infeasible to collect

sufficient ground truth on each user to build an activity model. It is beneficial to be able to transfer the activity models between users to reduce the annotation cost on each user.

**Cross-sensor** Transferring an activity model across different sensing technologies, for example from phone to watch, or between environments deployed with different ambient sensing technologies [37]. This can be the most complicated domain adaptation task in HAR where source and target domains are in heterogeneous feature spaces. It tackles a problem where a system tries to deploy a new set of sensors for activity recognition without the need of collecting any activity labels. With the help of an existing dataset that targets a similar set of activities, the adaptation task can quickly build an activity model with new sensors.

We have performed extensive experiments on these tasks with three commonly-used HAR datasets, and have compared ContrasGAN with eight state-of-the-art domain adaptation techniques. The results have consistently demonstrated superior performance of ContrasGAN.

## 2. Related Work

We first analyse the key general techniques for UDA, the state of the art techniques for UDA in HAR, and then the Contrastive GAN approach which we have applied to the HAR problem.

### *2.1. Unsupervised Domain Adaptation*

Domain adaptation (a sub field of transfer learning) is concerned about how to adapt or transfer a model from the source domain to the target domain, where both domains may have different distributions or feature spaces. Many of these techniques have demonstrated promising results in computer vision [51] and natural language processing [34]. The main objective is to bridge the source and target domains by projecting both feature spaces into common subspaces, and by exploring the correlations between two domains. *Unsupervised* domain adaptation (UDA) addresses the situation where the source domain data are labelled and the target domain data are not [53].

Transfer component analysis (TCA) [29] is a classic transfer learning technique that learns transfer components across two domains in a Reproducing Kernel Hilbert Space (RKHS) using maximum mean discrepancy (MMD). Geodesic flow kernels (GFK) [20] explore an infinite number of subspaces to represent the geometric changes and statistical properties from the source to the target domain. CORrelation ALignment (CORAL) [42] matches the distribution of two domains

by aligning their covariance, applying a linear transformation to the original source features and use the Frobenius norm to calculate the distance between the transformed source and target covariance matrices. CORAL has been extended using a deep neural network, *DeepCORAL*, to learn a nonlinear transformation that aligns correlations of layer activation between the source and target networks [43].

Domain adaptation techniques are increasingly based on deep learning methods. Deep Adaptation Networks (DAN) [25] embed the hidden representations of the task-specific layers of a convolutional neural network (CNN) in RKHS and explicitly matches the mean embeddings of source and target domain distributions. Joint Adaptation Networks (JAN) [27] extend DANs by aligning the joint distributions of these layers based on joint maximum mean discrepancy (JMMD), which measures the Hilbert-Schmidt norm between kernel mean embedding of empirical joint distributions of source and target data.

Adversarial domain adaptation is being seen as a means of increasing the robustness of domain adaptation through adversarial training. Tzeng *at alia* [47] have proposed an unsupervised adversarial adaptation method called Adversarial Discriminative Domain Adaptation (ADDA) that learns a discriminative representation using the labels in the source domain and builds an asymmetric mapping learned through a domain-adversarial loss to map the target data to the source representations. Domain-Adversarial Neural Networks (DANN) [19] combine domain adaptation with feature learning and adversarially updates the domain classifier to learn domain-invariant features. Adversarial Domain Adaptation with Domain Mixup (ADADM) [55] advances adversarial learning by mixing transformed source and real target samples to train a more robust generator.

### *2.2. Unsupervised Domain Adaptation in HAR*

Domain adaptation has been widely adopted in sensor-based human activity recognition [15]. Adaptive Spatial-Temporal Transfer Learning (ASTTL) [32] transfers activity knowledge and selects an appropriate source domain for cross-dataset HAR problems. It extends GFK with the Markov property to learn time-adaptive features in the manifold space. Convolutional deep Domain Adaptation model for Time Series data (CoDATS) [54] also tackles the source domain selection problem and is built on the domain adversarial neural network (DANN) [19]. Similarly, Chakma et al. [9] have proposed to map relevant feature representations from multiple source domains to the target domain. Chang *et al.* [10] have developed unsupervised domain adaptation algorithms on feature

matching and confusion maximisation and performed in-depth analysis of these algorithms in wearing diversity. Suzuki *et al.* [44] have applied GAN-based style transformers as a domain adaptation technique in gesture recognition, generating one user's gesture data from that of other users. These techniques mainly look into domain adaptation on homogeneous feature spaces: source and target domains share the same feature space, while their distributions may diverge. In contrast, our focus is on a more generic UDA approach to deal both homogeneous and heterogeneous feature spaces.

In our previous work, we have proposed an unsupervised domain adaptation technique, Shift-GAN [38], that applies a bi-directional GAN with Kernel Mean Matching to transform the feature spaces between the source and target domains. The technique has achieved promising results on several HAR datasets and outperformed the state-of-the-art transfer learning techniques. The main limitation of Shift-GAN is that it mainly focuses on sample-to-sample translation while not considering classes from which samples are drawn. It might therefore not learn class-separable features during the translation. To tackle this problem, in this work we investigate contrastive learning techniques that further improve the accuracy of activity adaptation.

### *2.3. Contrastive Learning*

Contrastive learning uses both similar and dissimilar samples when developing a classifier. While discriminative and predictive models generally measure loss in the output space (how good is the classifier?), contrastive learning measures loss in the representation space (how good is the representation?).

Hadsell *et al.* [21] introduce a contrastive loss function in learning dimensionality reduction function. The loss function runs over pairs of samples with the goal to maximise the distance between dissimilar pairs and minimise the distance between similar pairs. Zou *et al.* [58] propose spectral contrastive learning in learning an interpretable generative model for the foreground dataset in contrast to the background dataset. Schroff *et al.* [40] introduce a triplet loss in feature extraction for face images. The triplet loss consists of *anchor* image of a person, *positive* images of the same person, and *negative* images of different persons. The goal is to minimise the distance between anchor and positive images and maximise the distance between anchor and negative images. Dai *et al.* [16] apply a similar loss in image captioning. A Smooth Neighbors on Teacher Graphs (SNTG) is constructed to measure similarity of neighbour points on the low-dimensional manifold. It makes the learned features more discriminative by enforcing them to be similar for neighbors and dissimilar

for those non-neighbors. Luo *et al.* [28] and Khosla et al. [24] have introduced contrastive losses for semi-supervised learning and self-supervised learning respectively.

Kang *et al.* [23] introduce a contrastive domain discrepancy (CDD) loss to minimise the intra-class discrepancy and maximise the inter-class boundary. CDD extends MMD with class information and is combined with a categorical cross-entropy loss for domain adaptation. Chang *et al* [48] proposed a manifold alignment based approach for heterogeneous domain adaptation. It reuses the labels from multiple source domains in a target domain rather than aligning cross-domain correspondence. The model learns a mapping function to project the source and target domains to a new latent space and simultaneously matching the instances with the same labels and separating instances with different labels. Chen et al. [13] have employed Siamese networks for contrastive learning. Chen et al. [12] propose *SimCLR*, a simple framework for contrastive learning of visual representations; that is, learning representations by maximising agreement between differently augmented views of the same data example via a contrastive loss in the latent space. Tang et al. [46] have applied SimCLR to human activity recognition. Haresamudram et al. [22] have employed Contrastive Predictive Coding to learn temporal structures of sensor data.

Recently, Dai *et al.* [17] apply the contrastive loss based on Siamese network [7] and it works well on a number of domain adaptation tasks in computer vision. Our work is largely based on this contrastive learning technique and the key difference from our work is that we combine contrastive learning with Bi-GAN to enable heterogeneous domain adaptation on different HAR datasets, rather than homogeneous domain adaptation between images. Also we have adapted contrastive loss with batch normalisation to balance the magnitude of different loss components, which improves the convergence of training and recognition accuracy.

## 3. Proposed Approach

### *3.1. Problem Statement and Overview*

Formally, unsupervised domain adaptation between heterogeneous feature spaces can be defined as follows. Let $\mathcal{D}_s = \{(\mathbf{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{n_s}$ be a source domain with labels $y_s \in C$, where $C$ is the class set, $n_s$ is the number of source samples, and $\mathbf{x}_s \in \mathbb{R}^{d_s}$ is a sample of dimension $d_s$. Let $\mathcal{D}_t = \{\mathbf{x}_t^{(i)}\}_{i=1}^{n_t}$ be a target domain without labels, $n_t$ is the number of target samples, and $\mathbf{x}_t \in \mathbb{R}^{d_t}$. We assume that the source and target domain share the class set $C$ but have heterogeneous feature spaces; that

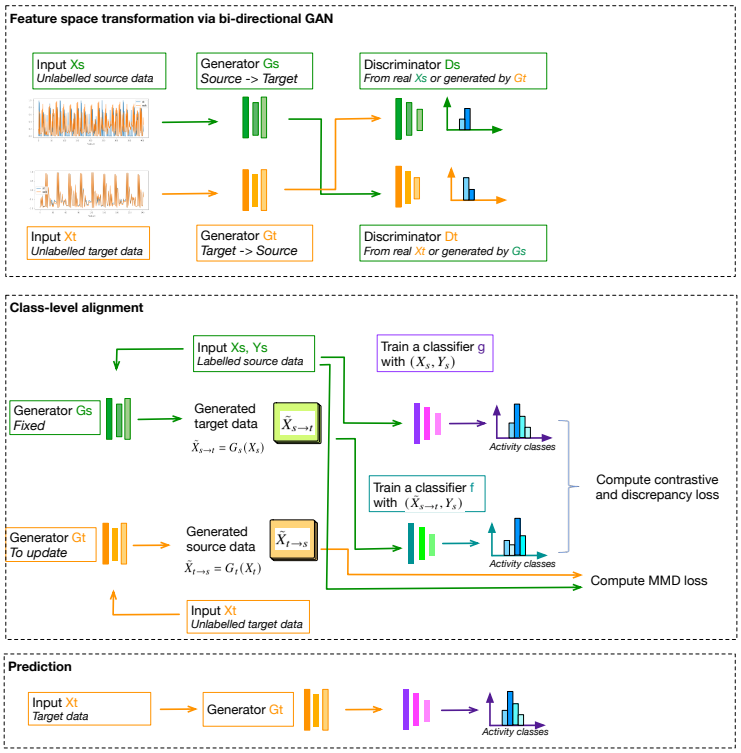



Figure 1: Workflow of ContrasGAN

is, their feature dimension $d_s$ and $d_t$ and the prior probabilities of features $P(X_s)(\mathbf{x}_s \sim X_s)$ and $P(X_t)(\mathbf{x}_t \sim X_t)$ can be different. The learning goal of UDA is to assign the correct class labels to target data points $\mathbf{x}_t$.

ContrasGAN is an UDA technique to perform activity transfer between heterogeneous datasets that have adopted different sensing technologies and sensor deployment in individual environmental settings and different activity models caused by personal routines – all of which give rise to heterogeneous feature sets. To address the high heterogeneity between domains, we need an UDA technique that does not make strong assumptions about feature spaces or probability distributions. Driven by this, we adopt a bi-directional generative adversarial network (Bi-GAN) [56, 57] to transform samples from the source domain to the target domain and *vice versa*. This type of technique has achieved success in many computer vision applications such as image-to-image translation [56, 57], and promising results in domain adaptation in HAR [38].

However, the general Bi-GAN is often class-agnostic [17, 23]: that is, it aims to generate samples between domains rather learning discriminative features for classification. This presents as a

limitation for the HAR tasks, where activities can have imbalanced distributions and overlapping class boundaries. These problems can present a challenge for Bi-GAN systems that focus only on matching feature distributions across domains while not identifying discriminating samples from different classes, and thus can mis-align them to incorrect classes. This is even worse on minor, less distinctive activities, as the alignment will be optimised to their majority counterpart activities. To address this challenge, we look into the recent advance of contrastive class alignment [17, 23] that target class-level discrimination in unsupervised domain adaptation.

ContrasGAN, as presented in Figure 1, is composed of the following components:

**Feature space transformation** Performs unsupervised feature space transformation between source and target domains via Bi-GAN.

**Class-level alignment** Performs class-level alignment via contrastive learning.

**Target label prediction** Predicts labels on the data in the target domain.

### *3.2. Bi-GAN with Expectation Loss*

In this section we will briefly introduce the details of GAN and Bi-GAN, and then describe how we extend the latter for better feature space transformation.

GAN systems consist of a generator and a discriminator. In the domain adaptation task the generator can learn to generate target samples from source samples, while the discriminator will try to tell whether a sample is generated or from the real target domain. When the discriminator is defeated, then we have a well-trained generator that bridges source and target domains.

Bi-GAN systems enable more robust adaptation by enforcing stronger, bi-directional constraints between two GANs. That is to say, Bi-GAN consists of two GANs $\{G_s, D_s\}$ and $\{G_t, D_t\}$, each consisting of a generator and a discriminator on the source and target domains respectively. $G_s$ takes a source example as input and generates an example in the target domain. This generated target example will be sent to the target discriminator $D_t$ to determine whether this example is generated from $G_s$ or from the real target data. Similarly, $G_t$ takes a target example and generates a source example, which will be sent to the source discriminator $D_s$.

Both generators are trained to generate samples as close as possible to the real samples in the other domain, and their objective function is to minimise the reconstruction losses:

$$\mathcal{L}_{G_s} = \sum_{i=1}^{n_s} ||G_t(G_s(\mathbf{x}_s^{(i)}, z), z') - \mathbf{x}_s^{(i)}|| \tag{1}$$

$$\mathcal{L}_{G_t} = \sum_{i=1}^{n_t} ||G_s(G_t(\mathbf{x}_t^{(i)}, z'), z) - \mathbf{x}_t^{(i)}|| \tag{2}$$

where $z$ and $z'$ are Gaussian noise introduced in $G_s$ and $G_t$ and they have the same input dimension as $G_s$ and $G_t$. To simplify the notation we will exclude $z$ and $z'$ in both generators. Both generators are implemented as neural networks, which take samples from each domain and output samples in the other domain. The feature space transformation is learned through two generators which do not place assumptions on the dimension or prior probabilities of the two domains. The cyclic constraints in Equation (1) and (2) make sure the original input samples can be recovered by the two generators: this is why Bi-GAN will lead to more robust domain adaptation.

However, since Bi-GAN only works on transformations between individual instances, we introduce another term – *expectation loss*, which constrain the global mapping between two feature spaces [3] to better globally align the generated space with the original space:

$$\begin{aligned}\mathcal{L}_{exp} &= \mathbb{E}[(G_t(\mathbf{X}_t) - \mathbf{X}_s)^2] + \mathbb{E}[(G_s(\mathbf{X}_s) - \mathbf{X}_t)^2] \\ &= \frac{1}{n_t}\sum_{i=1}^{n_t}(G_t(\mathbf{x}_t^{(i)}) - \mathbf{x}_s^{(i)})^2 + \frac{1}{n_s}\sum_{i=1}^{n_s}(G_s(\mathbf{x}_s^{(i)}) - \mathbf{x}_t^{(i)})^2\end{aligned} \tag{3}$$

The discriminators $D_s$ and $D_t$ are binary classifiers: $D_s$ detects whether an input is generated by $G_t$ or is a real sample from the source domain, and $D_t$ detects whether an input is generated by $G_s$ or is a real sample from the target domain. Their loss function is defined as follows:

$$\mathcal{L}_{D_s} = \sum_{i=1}^{n_s}(D_s(G_s(\mathbf{x}_s^{(i)})) - D_s(\mathbf{x}_t)) \tag{4}$$

$$\mathcal{L}_{D_t} = \sum_{i=1}^{n_t}(D_t(G_t(\mathbf{x}_t^{(i)})) - D_t(\mathbf{x}_s)) \tag{5}$$

The collaboration between the two GANs is established from their loss functions:

$$\mathcal{L} = \mathcal{L}_{G_s} + \mathcal{L}_{G_t} + \mathcal{L}_{exp} - \mathcal{L}_{D_s} - \mathcal{L}_{D_t} \tag{6}$$

The training of Bi-GAN goes as follows. We first train two discriminators $D_s$ and $D_t$ as binary classifiers on $\{\mathbf{x}_s^{(i)}\}_{i=1}^{N_s}$ and $\{\mathbf{x}_t^{(i)}\}_{i=1}^{N_s}$ using the loss functions in Equations (4) and (5). Then we train two generators $G_s$ and $G_t$ with $D_s$ and $D_t$ in an adversarial way using the loss function in Equation (6). Then we iterate the above two steps for several iterations until both generators and discriminators converge.

### *3.3. Class-level Alignment*

At the end of training Bi-GAN, we will have two generators that can transform samples from one domain to the other. Now our first task is to perform class-level alignment to learn class-discriminative feature transformation. To do so, we will first look at contrastive learning that has the strength in learning intra-class compactness *(i.e.*, grouping samples that share the same class labels) and inter-class separability (*i.e.*, pushing apart samples with different class labels).

Our second task is to further refine the source and target feature space transformation driven by the classification performance. We will focus on improving the target generator $G_t$ to transform target samples to the source domain as accurately as possible. To do so, we will train two classifiers $f$ and $g$. $f$ is trained on the transformed source data onto the target domain, $\tilde{\mathcal{D}}_{s\to t} = \{(\tilde{\mathbf{x}}_{s\to t}^{(i)}, y_s^{(i)}) \mid \tilde{\mathbf{x}}_{s\to t}^{(i)} = G_t(\mathbf{x}_s^{(i)})\}$. $g$ is trained on the original labelled source data $\mathcal{D}_s$. Then for the same target sample, we can measure the prediction discrepancy between $f$ and $g$. This discrepancy will guide the generator $G_t$ to better map target samples onto the source domain and thus minimise the discrepancy, and guide the classifier $g$ update to adapt to the transformed target samples. However, the prediction discrepancy is only for first-order moment matching [17]. To further improve the performance, we add maximum mean discrepancy (MMD) loss between the source samples and the transformed target samples to match the difference via higher-order moments. In the end, we will have improved target generator $G_t$ that can accurately transform target samples to the source domain and the classifier $g$ that can predict labels for transformed target samples.

#### *3.3.1. Contrastive Loss*

Our first step is to pre-label target samples. We transform all the source domain data to the target domain $\tilde{\mathcal{D}}_{t\to s} = \{(\tilde{\mathbf{x}}_{s\to t}^{(i)}, y_s^{(i)}) \mid \tilde{\mathbf{x}}_{s\to t}^{(i)} = G_t(\mathbf{x}_s^{(i)})\}$. We build a classifier $f$ on this transformed dataset and use it to pre-label the target data.

Once we have labelled the target data, we use contrastive loss to minimise the intra-class discrepancy and maximise the inter-class margin. The intra-class domain discrepancy is minimised so

as to compact the feature representations of samples within a class, whereas the inter-class domain discrepancy is maximised so as to push the representations further away from the decision boundary. The intra-class and inter-class discrepancies are jointly optimised to improve the adaptation performance.

The contrastive loss function is a distance-based loss function and runs over pairs of samples to ensure that semantically similar samples are embedded close together. (Here "semantically similar samples" means that the samples belong to the same class.) We define the following distance function on the source samples and the transformed source samples from the target domain.

$$con_dist((\tilde{\mathbf{x}}_{t\to s}^{(i)}, \tilde{y}_{t\to s}^{(i)}), (\mathbf{x}_s^{(j)}, y_s^{(j)})) = \begin{cases} ||\tilde{\mathbf{x}}_{t\to s}^{(i)} - \mathbf{x}_s^{(j)}||^2 & \tilde{y}_{t\to s}^{(i)} = y_s^{(i)} \\ max(0, m - ||\tilde{\mathbf{x}}_{t\to s}^{(i)}, \mathbf{x}_s^{(j)}||^2) & \tilde{y}_{t\to s}^{(i)} \neq y_s^{(j)} \end{cases} \tag{7}$$

where $y_{t\to s}$ is the predicted label of the sample $\tilde{\mathbf{x}}_{t\to s}$ and $y_s$ is the label on the real source sample $\mathbf{x}_s$. It measures the distance of a pair of *similar* samples belonging to the *same* class and constrains the distance of a pair of *dissimilar* samples belonging to *different* classes. $m$ is a pre-defined margin specifying the maximum distance between a pair of dissimilar samples over which the distance is 0, meaning that it will not contribute to the contrastive loss later. Following [21], the contrastive loss function is defined on the distance function as:

$$\mathcal{L}_{con} = \frac{1}{\epsilon} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} con_dist((\tilde{\mathbf{x}}_{t\to s}^{(i)}, \tilde{y}_{t\to s}^{(i)}), (\mathbf{x}_s^{(j)}, y_s^{(j)})). \tag{8}$$

where $\epsilon$ is a parameter used to normalise the contrastive loss. This normalisation prevents the overall loss being dominated by an individual loss with higher magnitude. We employ a batch normalisation so that the loss is normalised by the number samples in a batch, which helps balance the magnitude of different loss components.

### *3.3.2. Discrepancy Loss*

We also want to enforce consistency between two classifiers on the same target samples. The discrepancy loss represents the level of disagreement of the classifiers between the transformed and real target instances. Let $\mathbf{x}_t^{(i)}$ and $\mathbf{x}_{t\to s}^{(i)}$ be a target sample and its corresponding transformed sample in the source domain. Let $p_t^{(i)} = [p_{t,1}, ..., p_{t,C}]$ be the probability output vector on $\mathbf{x}_t^{(i)}$ from the classifier $f$, indicating the confidence of assigning the sample to each class ($\in [1, ..., C]$).

Similarly, let $p_{t\to s}^{(i)}$ be the probability output vector on $\mathbf{x}_{t\to s}^{(i)}$ from the classifier $g$. The discrepancy loss between $p_{t\to s}^{(i)}$ and $p_t^{(i)}$ is defined as:

$$\mathcal{L}_{disc} = \mathbb{E}[\sum_{i=1}^{n_t}(\frac{1}{C}\sum_{c=1}^{C}|p_{t,c}^{(i)} - p_{t\to s,c}^{(i)}|)] \tag{9}$$

where $|\cdot|$ denotes the $l_1$-norm. It measures the prediction difference between two classifiers on the same target instance, and guides the generator $G_t$ to transform better-aligned target samples to reduce this difference and the classifier $g$ to produce prediction probabilities more consistent with the other classifier $f$.

*3.3.3. MMD Loss*

MMD estimates the distance of two distributions by the mean of their projected embeddings in the reproducing kernel Hilbert space (RKHS) [6]. MMD is motivated by the fact that, if two distributions are identical, all of their statistics should be the same.

Let $X = \{\mathbf{x}_1\}_{i=1}^{n_1}$ and $X' = \{\mathbf{x}'_1\}_{i=1}^{n_2}$ be random variable sets with distributions $\mathcal{P}$ and $\mathcal{Q}$. The empirical estimated distance between $\mathcal{P}$ and $\mathcal{Q}$ defined by MMD is

$$mmd_dist(X, X') = \left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(\mathbf{x}'_i) \right\|_{\mathcal{H}} \tag{10}$$

where $\phi$ is the function mapping $X$ to $\mathcal{H}$ and $\mathcal{H}$ is a universal RKHS. Here, we define a MMD loss to minimise the distance of transformed and real feature spaces:

$$L_{MMD} = mmd_dist(G_{t\to s}(X_t), X_s) \tag{11}$$

Therefore, minimising MMD means to minimising all orders of moments. In practice, the squared value of MMD is estimated with the empirical kernel mean embeddings:

$$\mathcal{L}_{mmd}(X_s, X_{t\to s}) = \sum_{i=1}^{N_s}\sum_{j=1}^{N_{t\to s}} k\left(\phi\left(\frac{\tilde{\mathbf{x}}_{t\to s}^{(j)}}{\left\|\tilde{\mathbf{x}}_{t\to s}^{(j)}\right\|}\right), \phi\left(\frac{\mathbf{x}_s^{(i)}}{\left\|\mathbf{x}_s^{(i)}\right\|}\right)\right) \tag{12}$$

where $\phi(\cdot)$ is the kernel mapping and $\|\cdot\|$ denotes the $l_2$-norm. $k$ is the kernel to compute the inner product between two feature maps, $k(\mathbf{x}, \mathbf{x}') = \langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle_{\mathcal{H}}$ [6]. The MMD loss forces the normalised features in the two domains to be identically distributed, improving the global domain alignment.

*3.4. Algorithm and Training Regime*

Algorithm 1 describes the training process of ContrasGAN: (1) train a Bi-GAN model $G_s$ and $G_t$ to allow the transformation of instances between source and target domains; (2) perform class-level alignment. We initialise two classifiers $f$ and $g$ with the transformed source samples and the real source samples respectively; and then we fix the classifier $f$ and the source generator $G_s$, and update $G_t$ and the classifier $g$.

**ALGORITHM 1:** ContrasGAN Training

**Data:** Labelled source domain $\mathcal{D}_s = \{(\mathbf{x}_s^{(i)}, y_s^{(i)})\}_{i=1}^{N_s}$ and unlabelled target domain $\mathcal{D}_t = \{\mathbf{x}_t^{(j)}\}_{j=1}^{N_t}$

//1. Train generators $G_s$ and $G_t$ by training a Bi-GAN with $\{\mathbf{x}_s^{(i)}\}_{i=1}^{N_s}$ and $\{\mathbf{x}_t^{(j)})\}_{j=1}^{N_t}$

Initialise two generators $G_s$ and $G_t$ and two discriminators $D_s$ and $D_t$

**repeat**

  **foreach** *iteration* **do**

    sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{\mathbf{x}_s^{(j)}\}_{j=1}^{L}$ and $\{\mathbf{x}_t^{(j)}\}_{j=1}^{L}$

    update the parameters on $D_s$ to minimise $\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_s^d(\mathbf{x}_s, \mathbf{x}_t)$

    update the parameters on $D_t$ to minimise $\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_t^d(\mathbf{x}_t, \mathbf{x}_s)$

  **end**

  sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{\mathbf{x}_s^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_s$ and $\{\mathbf{x}_t^{(j)}\}_{j=1}^{L} \subseteq \mathcal{D}_t$

  update the weights on both generators to minimise $\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}$ in Eq (6)

**until** *converge*;

//2. Perform class-level alignment

Build a classifier $g$ on $\mathcal{D}_s$

Build a classifier $f$ on transformed source data $\mathcal{D}_s$

**repeat**

  **foreach** *iteration* **do**

    sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{(\mathbf{x}_s^{(j)}, y_s^{(j)})\}_{j=1}^{L}$ and $\{\mathbf{x}_t^{(j)}\}_{j=1}^{L}$

    generate transformed target samples via $G_t$: $\{\tilde{\mathbf{x}}_t^{(j)} | \tilde{\mathbf{x}}_t^{(j)} = G_t(\mathbf{x}_t^{(j)})\}_{j=1}^{L}$

    infer class labels and posterior probabilities on original target data and transformed target data

    update the parameters of $g$ to minimise $\mathcal{L}_{con}$

  **end**

  sample $L$-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$

  update the weights on $G_t$ to minimise $\mathcal{L}_{disc} + \mathcal{L}_{mmd}$

**until** *converge*;

## 4. Experimental Setup

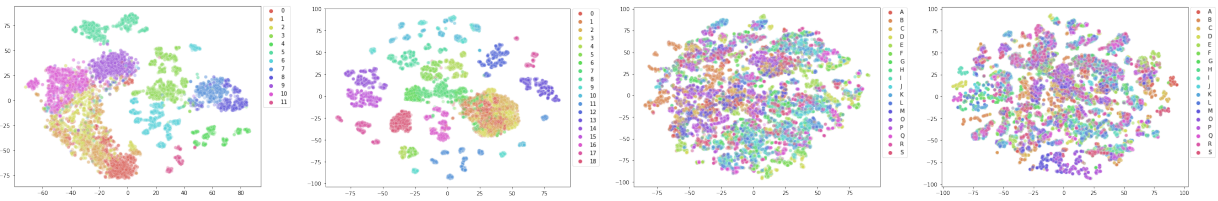
The main objective of our evaluation is to assess the performance of ContrasGAN in tackling the main types of domain adaptation tasks in human activity recognition identified in Section 1 cross-body, cross-user, and cross-sensor.

### *4.1. Datasets*

Driven by the above three types of experiments, we seek the datasets that are collected on a large number of users with diverse wearing positions, cover a variety of activities and involve different sensing technologies; for example, body-worn tri-axial accelerometers and smart phones/watches. For this purpose, we have reviewed the publicly available HAR datasets that have been widely adopted as benchmarks in recent HAR research [4, 54] and identified the following: the Smartphone and Smartwatch Activity and Biometrics dataset (WISDM) [52], Daily and Sports Activities (DSADS) [1, 2], and the Physical Activity MonitoringdDataset (PAMAP) [35]. The WISDM dataset was collected from 51 subjects performing 18 daily activities including walking, jogging, brushing teeth, eating and drinking. During the data collection each subject either wore a smartwatch (*i.e.*, an LG G Watch) on their dominant wrist or had a smartphone (*i.e.*, Samsung Galaxy or Google Nexus 5/5X) in their pocket. The accelerometry and gyroscopic data from both watches and phones were collected at a rate of 20Hz. We use the handcrafted features from the WISDM dataset including the mean, standard deviation, mel-frequency cepstrum coefficients (MFCC) of each dimension, and correlations between dimensions [52]. The DSADS dataset consists of data from 5 body-worn accelerometers from 8 subjects when they were performing 19 daily activities including running on a treadmill and exercising on a stepper. The accelerometers were placed on each subject's right arm, left arm, right leg, left leg, and torso. The PAMAP dataset was collected from 9 subjects who wore 3 accelerometers on their arm, ankle, and chest and performed 12 activities such as ironing, house cleaning, and sitting. We use the features [50, 52] generated from these datasets, including including mean, standard deviation, and correlation between sensor values for pairs of dimensions. Table 1 summarises the characteristics of each of the datasets. Figure 2 presents the t-SNE plots for the datasets, which demonstrate the diversity of samples in each class.

Table 1: Descriptions of Datasets

| Dataset | | No. of Features | No. of Samples | No. of Users | No. of Activities | Activities |
|---|---|---|---|---|---|---|
| PAMAP | | 243 | 7352 | 9 | 12 | ascending stairs (0), cycling (1), descending stairs (2), ironing (3), lying (4), nordic walking (5), rope jumping (6), running (7), sitting (8), standing (9), vacuum cleaning (10), walking (11) |
| DSADS | | 405 | 9120 | 8 | 19 | sitting (0), standing (1), lying on back (2) and on right side (3), ascending (4) and descending stairs (5), standing in an elevator still (6), moving around in an elevator (7), walking in a parking lot (8), walking on a treadmill with a speed of 4km/h in flat (9) and 15 deg inclined positions (10), running on a treadmill with a speed of 8 km/h (11), exercising on a stepper (12), exercising on a cross trainer (13), cycling on an exercise bike in horizontal (14) and vertical positions (15), rowing (16), jumping (17), and playing basketball (18) |
| WISDM | PHONE | 90 | 40553 | 51 | 18 | walking (A), jogging (B), stairs (C), sitting (D), standing (E), typing (F), brushing teeth (G), eating soup (H), eating chips (I), eating pasta (J), drinking from cup (K), eating sandwich (L), kicking (M), playing catch w/ Tennis Ball (O), dribllinlg (P), writing (Q), clapping (R), folding clothes (S) |
| | WATCH | 90 | 34942 | | | |



(a) PAMAP (b) DSADS (c) WISDM Watch (d) WISDM Phone

Figure 2: t-SNE plots for the selected datasets. The activity labels are shown in Table 1.

### *4.2. Implementation, Configuration and Hyperparameter Tuning*

In terms of the architecture of Bi-GAN, each generator contains three linear layers with leaky ReLu as the activation function. Each layer has the same dimension as the input layer. Batch normalisation is applied and between each pair of layers, with a dropout rate of 0.2. The discriminator has a similar architecture as the generator and the only difference is that the last layer of the discriminator is for classification and has a sigmoid activation function and binary output. We train Bi-GAN over 2000 epochs, and then take the models at each key epoch (i.e., 500, 1000, 1500, and 2000) for contrastive learning. We select the final model that achieves the highest accuracy on the validation data. For class-level alignment, both classifiers are implemented as a two-layered neural network where each hidden layer has the same dimension as the input layer and the output layer maps to the number of classes. We use ADAM as the optimiser with the learning rate is set to 0.0001 decaying to 0.001. We train the model for 500 epochs with a batch size of 100.

We have performed the grid search with the hyperparameters including the number of layers for generators, discriminators (i.e., 2, 3 and 4), and batch size (i.e., 25, 50, and 100), learning rate (i.e., 0.0001, 0.0005, 0.001, 0.005, and 0.01), dropout rates (i.e., 0.2, 0.4, and 0.6), and activation function (i.e., ReLu and leaky ReLu). We choose the configuration that leads to the highest F1-scores on the validation set. The code[1] for the experiments is in Python, using the PyTorch, Numpy and Pandas libraries. The experiments have been run on an Intel Core i7-9700K CPU 3.60GHz with 32GB memory.

### *4.3. Comparison Techniques and Evaluation Metrics*

We compare ContrasGAN with the state-of-the-art domain adaptation techniques and hypothesise that ContrasGAN will yield a performance gain. First of all, we choose two classic transfer learning tasks that have achieved promising performance on HAR domain adaptation tasks [14, 37, 38]: TCA and GFK. Then we choose our previous Bi-GAN based technique – ShiftGAN [38] – and the recent state-of-the-art unsupervised domain adaptation techniques: DAN, DANN, JAN, DeepCORAL, and ADADM. Beyond these 8 techniques, we add the upper- and lower-bound baselines: the upper-bound baseline trains a classifier with 80% of original target data and test on 20% of the target data, which indicates the best performance that we can achieve on the target data; while

[1]The source code will be accessible at: https://github.com/An5r3a/ContrasDGAN.

the lower-bound baseline trains a classifier with all of the source data and test on the target data, which suggests the difference between source and target domains and thus indicates the difficulty of a transfer learning task.

The performance is measured by *F1 score*, the harmonic mean of precision and recall. As the selected datasets have imbalanced activity distribution, we report both the *micro-F1 score* which aggregates the contributions of all instances, and the *macro-F1 score* which computes the metric independently for each class and then takes the average. These two metrics are often used in domain adaptation and HAR in general [4, 38], with macro-F1 score is for imbalanced datasets [18, 31]. For each experiment, we run 10 times and report the mean of macro and micro F1-scores as the final result. For each run, we follow the state-of-the-art UDA evaluation methodology [9, 44, 54] by performing stratified split on the target domain data into 60% for training, 20% for validation, and 20% for testing. Then we train ContrasGAN or a comparison technique with all the source domain data with labels and the training data of target domain data without labels, and then predict the labels on the test data of the target domain for evaluation.

## 5. Results and Discussion

This section presents and discusses our results in the three experiments.

### *5.1. Cross-Body experiment*

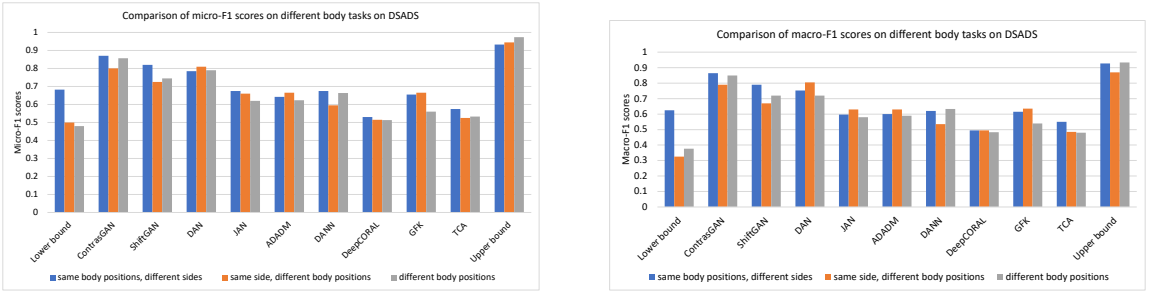



Figure 3: Comparison of micro- and macro-F1 scores on different body tasks on DSADS

The cross-body experiment measures the accuracy of transferring an activity model learnt on one body position (*e.g.*, left arm) to another position (*e.g.*, right arm). Tables 2a and 2b reports the micro- and macro-F1 scores of ContrasGAN and the baseline techniques on *cross-body* experiments. On the DSADS dataset, we perform 9 tasks, (1) between the sides of the same position including

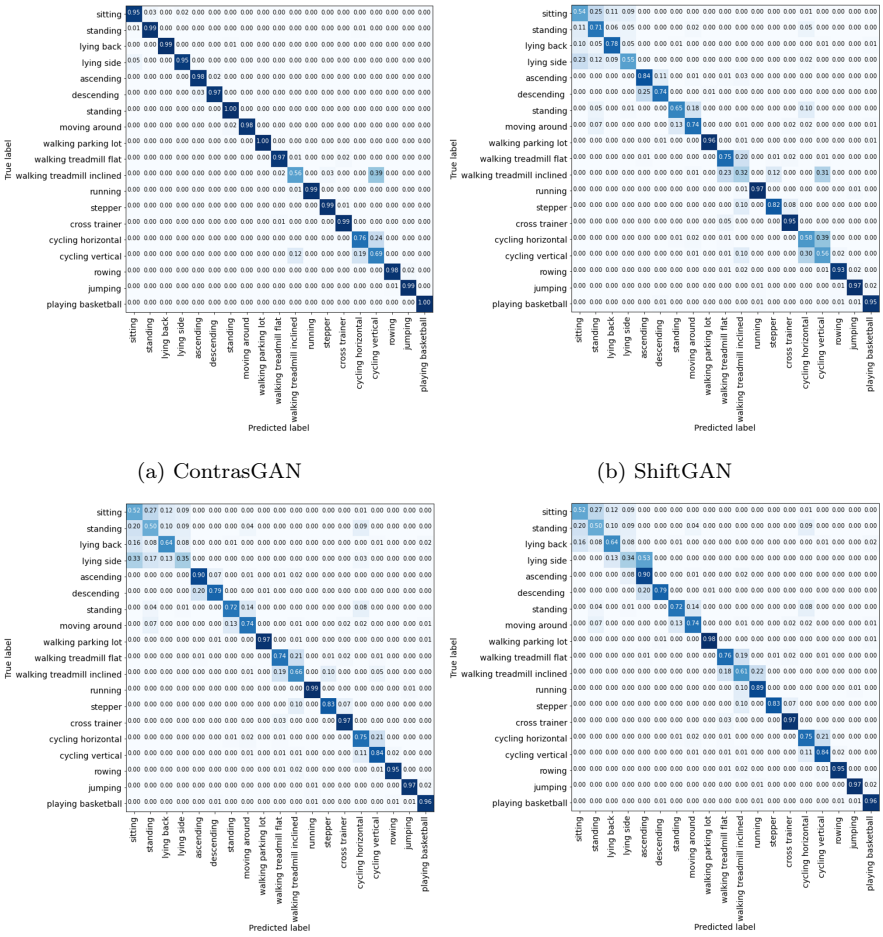
Table 2: Comparison of F1 scores between ContrasGAN and baseline techniques on *cross-body* experiments on DSADS and PAMAP datasets.

(a) Micro-F1

| | Tasks | Lower bound | ContrasGAN | ShiftGAN (2021) | DAN (2015) | JAN (2017) | ADADM (2020) | DANN (2016) | DeepCORAL (2016) | GFK (2012) | TCA (2011) | Upper bound |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSADS | RA-LA | 0.68 | 0.9 (+0.22) | **0.91 (+0.23)** | 0.72 (+0.04) | 0.7 (+0.02) | 0.74 (+0.06) | 0.78 (+0.10) | 0.55 (-0.13) | 0.74 (+0.06) | 0.67 (-0.01) | 0.94 |
| | LA-RA | 0.67 | **0.81 (+0.14)** | 0.71 (+0.04) | 0.8 (+0.13) | 0.69 (+0.02) | 0.58 (-0.09) | 0.64 (-0.03) | 0.51 (-0.16) | 0.58 (-0.09) | 0.53 (-0.14) | 0.93 |
| | RL-LL | 0.8 | 0.9 (+0.10) | **0.95 (+0.15)** | 0.78 (-0.02) | 0.63 (-0.17) | 0.6 (-0.20) | 0.73 (-0.07) | 0.55 (-0.25) | 0.69 (-0.11) | 0.6 (-0.20) | 0.97 |
| | LL-RL | 0.58 | **0.87 (+0.29)** | 0.71 (+0.13) | 0.84 (+0.26) | 0.68 (+0.10) | 0.65 (+0.07) | 0.55 (-0.03) | 0.51 (-0.07) | 0.61 (+0.03) | 0.5 (-0.08) | 0.89 |
| | RA-RL | 0.43 | **0.8 (+0.37)** | 0.68 (+0.25) | 0.8 (+0.37) | 0.67 (+0.24) | 0.65 (+0.22) | 0.54 (+0.11) | 0.51 (+0.08) | 0.66 (+0.23) | 0.52 (+0.09) | 0.92 |
| | LA-LL | 0.57 | 0.8 (+0.23) | 0.77 (+0.20) | **0.82 (+0.25)** | 0.65 (+0.08) | 0.68 (+0.11) | 0.65 (+0.08) | 0.52 (-0.05) | 0.67 (+0.10) | 0.53 (-0.04) | 0.97 |
| | RA-T | 0.47 | 0.89 (+0.42) | **0.9 (+0.43)** | 0.72 (+0.25) | 0.56 (+0.09) | 0.6 (+0.13) | 0.78 (+0.31) | 0.55 (+0.08) | 0.49 (+0.02) | 0.51 (+0.04) | 0.95 |
| | T-RA | 0.43 | **0.86 (+0.43)** | 0.63 (+0.20) | 0.82 (+0.39) | 0.62 (+0.19) | 0.6 (+0.17) | 0.54 (+0.11) | 0.49 (+0.06) | 0.56 (+0.13) | 0.54 (+0.11) | 0.98 |
| | LA-T | 0.54 | 0.82 (+0.28) | 0.71 (+0.17) | **0.83 (+0.29)** | 0.68 (+0.14) | 0.67 (+0.13) | 0.67 (+0.13) | 0.5 (-0.04) | 0.63 (+0.09) | 0.55 (+0.01) | 0.99 |
| PAMAP | H-C | 0.41 | 0.65 (+0.24) | 0.59 (+0.18) | **0.72 (+0.31)** | 0.44 (+0.03) | 0.58 (+0.17) | 0.64 (+0.23) | 0.62 (+0.21) | 0.7 (+0.29) | 0.56 (+0.15) | 0.88 |
| | C-H | 0.49 | **0.83 (+0.34)** | 0.52 (+0.03) | 0.82 (+0.33) | 0.65 (+0.16) | 0.64 (+0.15) | 0.55 (+0.06) | 0.51 (+0.02) | 0.62 (+0.13) | 0.52 (+0.03) | 0.94 |
| | A-C | 0.36 | **0.82 (+0.46)** | 0.74 (+0.38) | 0.79 (+0.43) | 0.7 (+0.34) | 0.7 (+0.34) | 0.63 (+0.27) | 0.53 (+0.17) | 0.68 (+0.32) | 0.57 (+0.21) | 0.96 |
| | C-A | 0.46 | **0.84 (+0.38)** | 0.53 (+0.07) | 0.75 (+0.29) | 0.61 (+0.15) | 0.61 (+0.15) | 0.52 (+0.06) | 0.51 (+0.05) | 0.7 (+0.24) | 0.56 (+0.10) | 0.96 |
| | H-A | 0.56 | **0.8 (+0.24)** | 0.51 (-0.05) | 0.7 (+0.14) | 0.62 (+0.06) | 0.62 (+0.06) | 0.42 (-0.14) | 0.53 (-0.03) | 0.66 (+0.10) | 0.58 (+0.02) | 0.96 |
| | A-H | 0.34 | **0.87 (+0.53)** | 0.73 (+0.39) | 0.85 (+0.51) | 0.8 (+0.46) | 0.8 (+0.46) | 0.48 (+0.14) | 0.5 (+0.16) | 0.66 (+0.32) | 0.59 (+0.25) | 0.95 |
| Avg | | 0.52 | **0.83 (+0.31)** | 0.71 (+0.19) | 0.78 (+0.26) | 0.65 (+0.13) | 0.65 (+0.13) | 0.61 (+0.09) | 0.53 (+0.01) | 0.64 (+0.12) | 0.56 (+0.04) | 0.95 |

(b) Macro-F1

| | Tasks | Lower bound | ContrasGAN | ShiftGAN (2021) | DAN (2015) | JAN (2017) | ADADM (2020) | DANN (2016) | DeepCORAL (2016) | GFK (2012) | TCA (2011) | Upper bound |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSADS | RA-LA | 0.67 | **0.9 (+0.23)** | 0.89 (+0.22) | 0.64 (-0.03) | 0.53 (-0.14) | 0.67 (0.00) | 0.77 (+0.10) | 0.5 (-0.17) | 0.7 (+0.03) | 0.66 (-0.01) | 0.94 |
| | LA-RA | 0.52 | **0.8 (+0.28)** | 0.65 (+0.13) | 0.79 (+0.27) | 0.66 (+0.14) | 0.55 (+0.03) | 0.49 (-0.03) | 0.48 (-0.04) | 0.53 (+0.01) | 0.46 (-0.06) | 0.91 |
| | RL-LL | 0.79 | 0.89 (+0.10) | **0.94 (+0.15)** | 0.74 (-0.05) | 0.52 (-0.27) | 0.56 (-0.23) | 0.71 (-0.08) | 0.5 (-0.29) | 0.65 (-0.14) | 0.59 (-0.20) | 0.97 |
| | LL-RL | 0.52 | **0.87 (+0.35)** | 0.70 (+0.18) | 0.84 (+0.32) | 0.68 (+0.16) | 0.62 (+0.10) | 0.51 (-0.01) | 0.5 (-0.02) | 0.58 (+0.06) | 0.49 (-0.03) | 0.89 |
| | RA-RL | 0.25 | **0.8 (+0.55)** | 0.62 (+0.37) | 0.79 (+0.54) | 0.64 (+0.39) | 0.62 (+0.37) | 0.52 (+0.27) | 0.48 (+0.23) | 0.63 (+0.38) | 0.49 (+0.24) | 0.88 |
| | LA-LL | 0.4 | 0.78 (+0.38) | 0.73 (+0.33) | **0.82 (+0.42)** | 0.62 (+0.22) | 0.64 (+0.24) | 0.55 (+0.15) | 0.51 (+0.11) | 0.64 (+0.24) | 0.48 (+0.08) | 0.86 |
| | RA-T | 0.44 | **0.89 (+0.45)** | 0.89 (+0.45) | 0.52 (+0.08) | 0.5 (+0.06) | 0.55 (+0.11) | 0.77 (+0.33) | 0.5 (+0.06) | 0.48 (+0.04) | 0.5 (+0.06) | 0.95 |
| | T-RA | 0.26 | **0.85 (+0.59)** | 0.61 (+0.35) | 0.82 (+0.56) | 0.59 (+0.33) | 0.57 (+0.31) | 0.52 (+0.26) | 0.49 (+0.23) | 0.53 (+0.27) | 0.44 (+0.18) | 0.97 |
| | LA-T | 0.43 | 0.81 (+0.38) | 0.67 (+0.24) | **0.82 (+0.39)** | 0.65 (+0.22) | 0.65 (+0.22) | 0.61 (+0.18) | 0.46 (+0.03) | 0.61 (+0.18) | 0.5 (+0.07) | 0.88 |
| PAMAP | H-C | 0.36 | **0.64 (+0.28)** | 0.55 (+0.19) | 0.49 (+0.13) | 0.32 (-0.04) | 0.56 (+0.20) | 0.64 (+0.28) | 0.52 (+0.16) | 0.65 (+0.29) | 0.52 (+0.16) | 0.88 |
| | C-H | 0.28 | **0.82 (+0.54)** | 0.48 (+0.20) | **0.82 (+0.54)** | 0.63 (+0.35) | 0.61 (+0.33) | 0.53 (+0.25) | 0.49 (+0.21) | 0.6 (+0.32) | 0.47 (+0.19) | 0.91 |
| | A-C | 0.3 | **0.81 (+0.51)** | 0.69 (+0.39) | 0.77 (+0.47) | 0.6 (+0.30) | 0.66 (+0.36) | 0.48 (+0.18) | 0.47 (+0.17) | 0.64 (+0.34) | 0.55 (+0.25) | 0.72 |
| | C-A | 0.41 | **0.84 (+0.43)** | 0.43 (+0.02) | 0.74 (+0.33) | 0.61 (+0.20) | 0.59 (+0.18) | 0.38 (-0.03) | 0.46 (+0.05) | 0.64 (+0.23) | 0.48 (+0.07) | 0.75 |
| | H-A | 0.5 | **0.73 (+0.23)** | 0.41 (-0.09) | 0.67 (+0.17) | 0.62 (+0.12) | 0.6 (+0.10) | 0.38 (-0.12) | 0.48 (-0.02) | 0.64 (+0.14) | 0.56 (+0.06) | 0.84 |
| | A-H | 0.28 | 0.73 (+0.45) | 0.61 (+0.33) | **0.84 (+0.56)** | 0.63 (+0.35) | 0.79 (+0.51) | 0.37 (+0.09) | 0.5 (+0.22) | 0.66 (+0.38) | 0.54 (+0.26) | 0.83 |
| Avg | | 0.43 | **0.81 (+0.38)** | 0.66 (+0.23) | 0.74 (+0.31) | 0.59 (+0.16) | 0.62 (+0.19) | 0.55 (+0.12) | 0.49 (+0.06) | 0.61 (+0.18) | 0.52 (+0.09) | 0.88 |




(a) ContrasGAN (b) ShiftGAN

(c) DAN (d) DANN

Figure 4: Confusion matrices on the RA-T task

right arm to left arm (RA-LA), left arm to right arm (LA-RA), right leg to left leg (RL-LL), left leg to right leg (LL-RL); (2) between different positions including right arm to torso (RA-T), torso to right arm (T-RA), left arm to torso (LA-T); and (3) between different positions on the same side right arm to right leg (RA-RL), left arm to left leg (LA-LL). On the PAMAP dataset, we perform 6 tasks between three body positions: hand (H), chest (C), and Ankle (A).

Tables 2a and 2b report the averaged F1 scores over 10 runs and the increased micro-F1 scores from the lower-bound performance (in brackets). We highlight the best accuracy in bold and the negative transfers in red. ContrasGAN achieves the best macro-F1 scores on 11 out of 15 tasks without any negative transfer. It obtains the highest averaged micro- and macro-F1 scores: 0.83 and 0.81, which is 0.05 and 0.07 higher than DAN, the second best performing technique. It is

also worth noting that with class-level alignment ContrasGAN can achieve more balanced accuracy on each class, leading to much higher macro-F1 scores than all the comparison techniques. In addition, ContrasGAN outperforms ShiftGAN with 0.12 and 0.15 in micro- and macro-F1 scores, which demonstrates the effectiveness of contrastive learning in capturing discriminative transfer features between classes.

Looking more closely at different body tasks on the DSADS dataset, we have plotted the averaged micro- and macro-F1 scores on each type of task in Figure 3. The lower-bound accuracy indicates the difficulty level of these adaptation tasks: the easiest one is the same body part between different sides such as left arm to right arm, while the tasks between different body parts are more difficult. The results show that 6 out of 9 techniques perform better on the tasks of same body part and different sides, and have achieved higher F1 scores than the other two types of tasks. The difference in F1-scores between the tasks of different body parts and same side but different body parts is very small: 0.01 in both micro- and macro-F1 scores averaged on all the comparison techniques. This suggests that the difficulty of transferring activity models depends on different body parts more than different sides between the same body part: body parts, even though on different sides, have similar motion patterns. For example, legs have different frequencies of kinematic and EMG oscillations from arms [45]. It is therefore easier to transfer activity models between sides of legs than between legs and arms.

Figure 4 presents confusion matrices on ContrasGAN, ShiftGAN, DANN, and DAN. ShiftGAN, DANN, and DAN struggle to distinguish similar activities, for example sitting, standing, lying back and lying side, and walking on treadmill with different speeds. ContrasGAN with contrastive learning has demonstrated superior performance on separating these groups of activities, even though it also faces the challenge of differentiating subtle activities like different cycling modes.

Among the comparison techniques, DeepCORAL and JAN do not work well on the accelerometer data and have produced worse accuracy than the classic techniques TCA and GFK. ShiftGAN, DAN, JAN (a variation of DAN), and ADADM have achieved better overall performance. ShiftGAN is also built on Bi-GAN and extends it with kernel mean matching technique to align transformed target data with the real target data. Both ContrasGAN and ShiftGAN are better than the others, including a single GAN based approach – ADADM, which shows that Bi-GAN is effective for domain adaptation. DAN and JAN that combine domain adaptation with feature learning; i.e.,

minimises the distance of hidden representations of source and target domains at the task-specific layers, have shown their promise in achieving global adaptation.

Looking across the tasks, we can see that the lower-bound accuracy on the tasks of RA-T and H-C are the lowest, suggesting that the source and target domains have very different distributions. Also the upper bound accuracy on H-C is lower than the others, implying the difficulty of classifying on the target domain. On this task, most of techniques do not perform well. DANN has achieved the highest micro-F1 score 0.72, which is 0.08 higher than ContrasGAN. However, its macro-F1 score is only 0.49, 0.14 lower than ContrasGAN. This suggests that its learning prioritises the majority classes. On the other hand, ContrasGAN, DANN and GFK are able to achieve more balanced accuracy across all the classes.

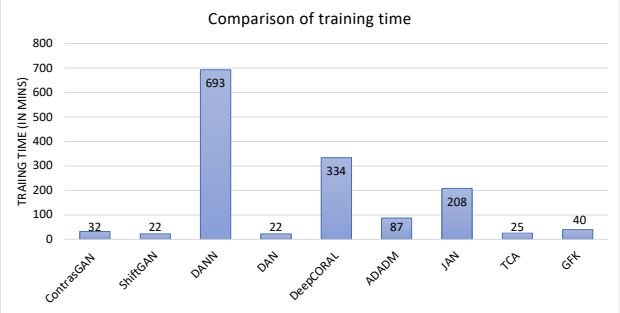
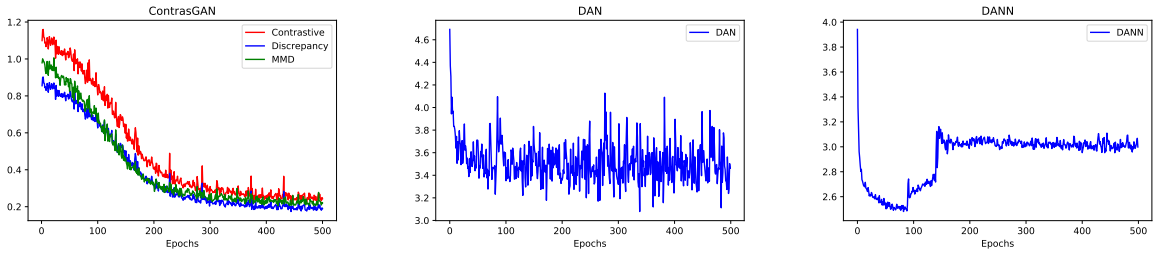



Figure 5: Comparison of training time between ContrasGAN and other techniques. ContrasGAN has the smallest training time averaged across 15 transfer learning tasks.

### *5.2. Computation Cost*

Figure 5 compares the training time between ContrasGAN and the other techniques. ShiftGAN and DAN have the least training time and ContrasGAN takes 10 more minutes (45% more time) on average when training each task. DANN is the most expensive as it performs adaptation with feature learning and employs an end-to-end training regime, which takes longer to converge. This becomes a particular problem when the two domains are highly heterogeneous. The training time of DANN on the W-C task is 1784 mins, nearly 50 times of the training time on ContrasGAN and significantly higher than all the other techniques.

Figure 6 presents the loss plots of ContrasGAN for class-level alignment, DNN, and DANN. This shows that ContrasGAN can converge smoothly and stably.




(a) ContrasGAN (b) DAN (c) DANN

Figure 6: Comparison of loss between ContrasGAN, DAN, and DANN on the task of RA-LA

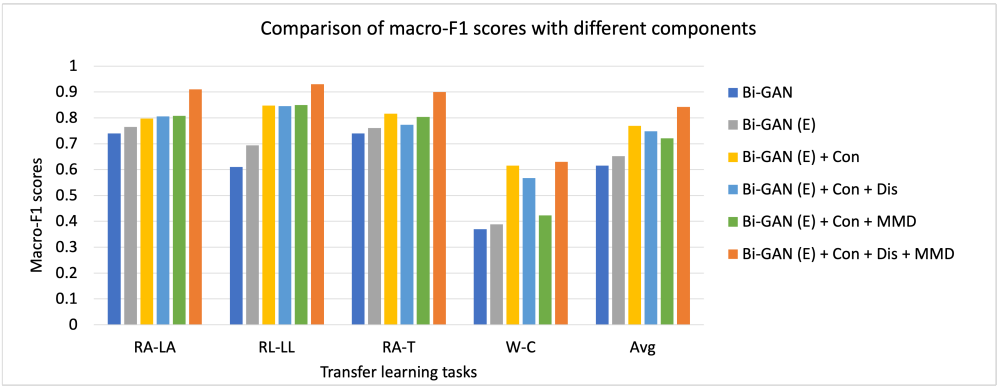


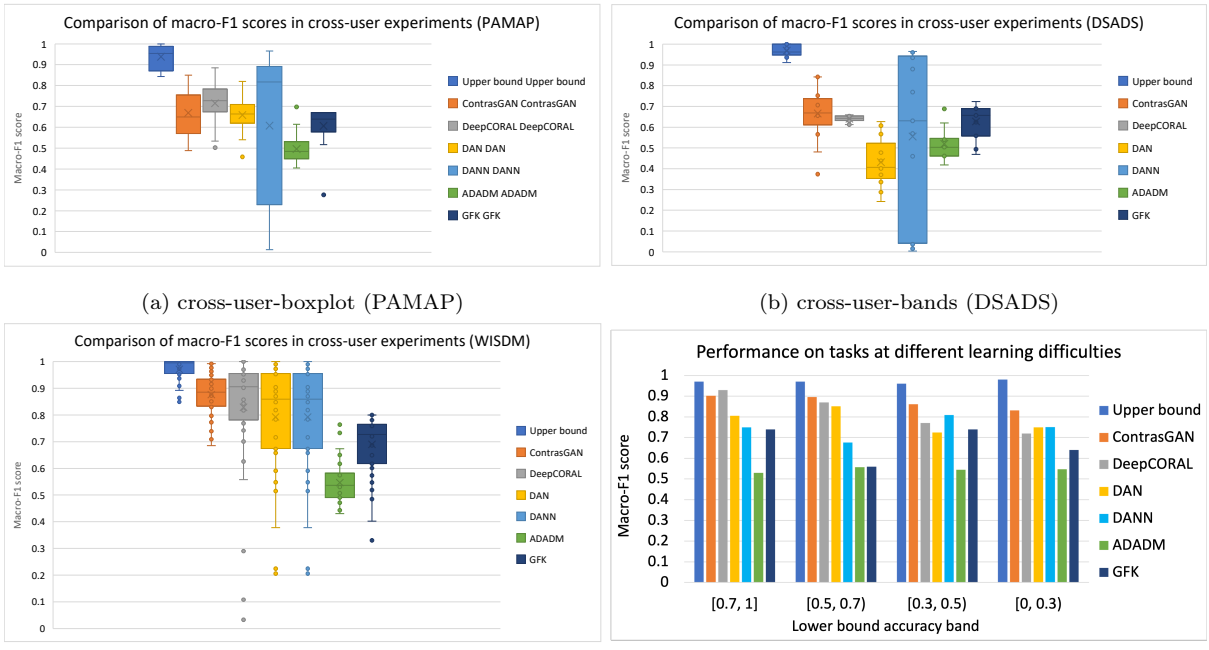

Figure 7: Ablation study of ContrasGAN

### 5.3. Ablation Study

ContrasGAN extends the original Bi-GAN model with two components, introducing the expectation loss and adding the class-level alignment with MMD, discrepancy and contrastive loss. Here, we evaluate the importance of each type of new loss *via* an ablation study. Figure 7 presents the macro-F1 scores of the ablation experiments on a subset of tasks.

Based on these results, we can make the following observations. Firstly, contrastive loss significantly boosts performance, especially on task W-C, where it outperforms Bi-GAN with 0.25 in macro-F1 scores. This indicates that the contrastive loss help learn discriminative features and improve class-level recognition accuracy. Secondly, the combination of contrastive, discrepancy, and MMD loss makes significant improvement, from 0.65 on Bi-GAN with expectation loss to 0.84 on ContrasGAN in the averaged macro-F1 scores. It suggests that the discrepancy and MMD loss makes contribution to transform the global feature spaces during the fine alignment stage. Thirdly, the expectation loss, discrepancy, and MMD loss each individually improves performance on these transfer learning tasks only to a certain degree.

### 5.4. Cross-User Experiment




(a) cross-user-boxplot (PAMAP)

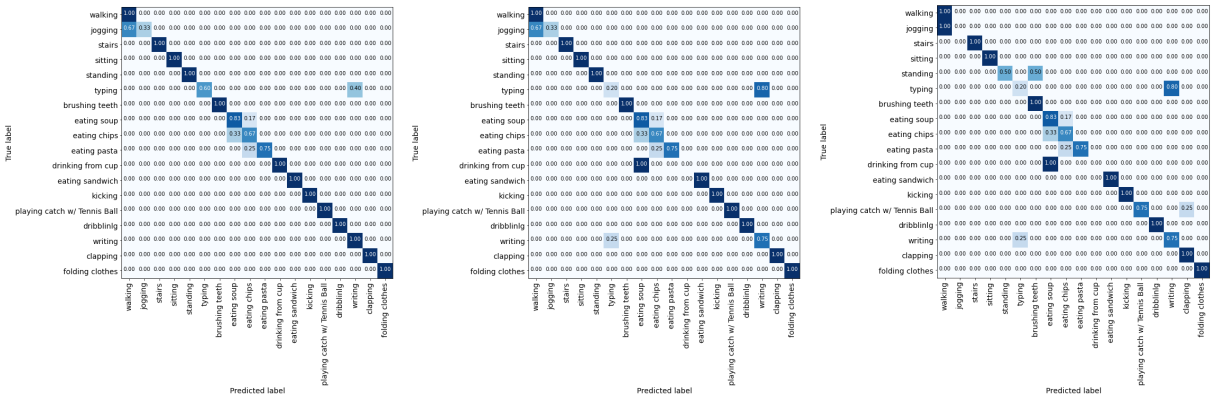
(b) cross-user-bands (DSADS)

(c) cross-user-boxplot (WISDM)

(d) cross-user-bands (WISDM)

Figure 8: Comparison of performance on the cross-user experiments

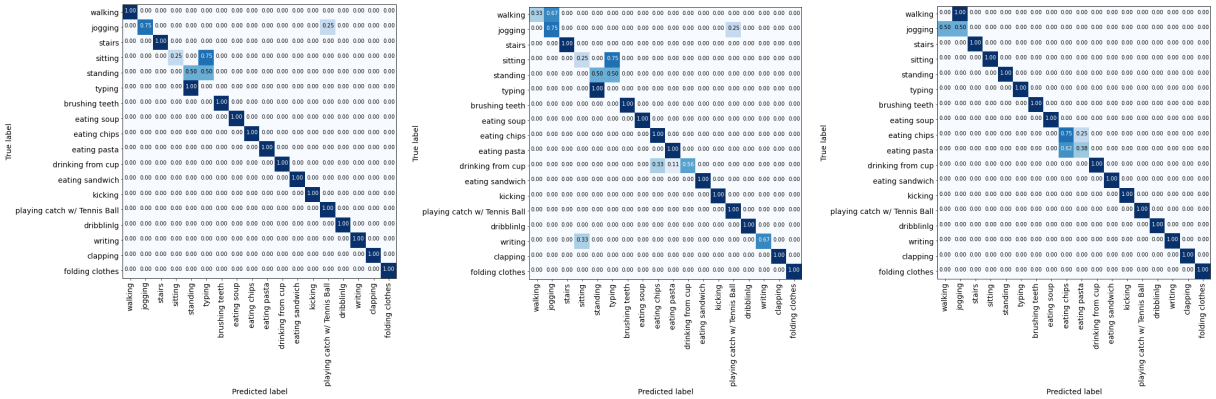
The cross-user experiment measures the accuracy of transferring an activity model from one user (source) to another user (target). We have randomly sampled 40 pairs of users on each dataset, and evaluated our techniques by taking one of them as the source domain and the other as the target domain. Figure 8 presents a boxplot of macro-F1 scores between ContrasGAN, the upper-bound baseline, and the comparison techniques. Overall, DeepCORAL performs the best among all the comparison techniques and is very comparable to ContrasGAN. On PAMAP in Figure 8a, ContrasGAN has achieved the averaged macro-F1 scores of 0.67 (0.05 lower than DeepCORAL), on DSADS in Figure 8b, ContrasGAN has 0.67 (0.03 higher), and on WISDM in Figure 8c, ContrasGAN has 0.88 (0.05 higher). On PAMAP and WISDM datasets, ContrasGAN has smaller variance compared to the other techniques, indicating that it is less sensitive to different types of data. On DSADS, DANN suffers high fluctuation in consistent accuracy, but slightly lower than ContrasGAN.




(a) ContrasGAN (b) DeepCORAL (c) DAN

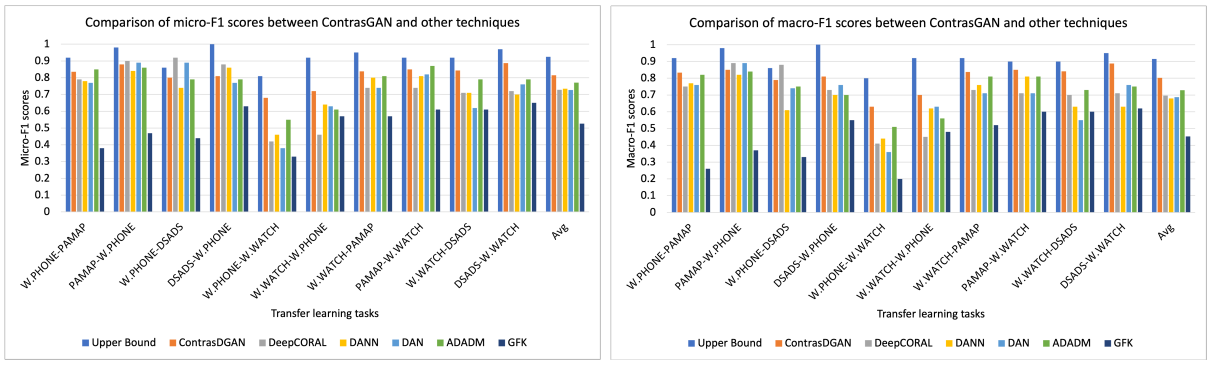
Figure 9: Confusion matrices on the user pair 1605-1602 task

To look more closely at the results, we put the accuracy of lower-bound baseline into different bands showing different difficulty levels of learning tasks on WISDM dataset. For example, a user pair with a high lower-bound accuracy suggests that these users' data have similar distributions, and thus that the difficulty of transfer learning will be lower. Based on this intuition, we define 4 bands, with the band of [0.7, 1] grouping all the easiest tasks and the band of [0, 0.3) grouping all the most difficult tasks. The results on these different bands are shown in Figure 8d. ContrasGAN consistently outperforms the other techniques on tasks at different difficulty levels. All the techniques seem insensitive to the learning difficulty. ADADM seems suffer from negative




(a) ContrasGAN (b) DANN (c) GFK

Figure 10: Confusion matrices on the 1602-1642 task




(a) Micro-F1 (b) Macro-F1

Figure 11: Comparison of F1 scores on the cross-sensor experiments

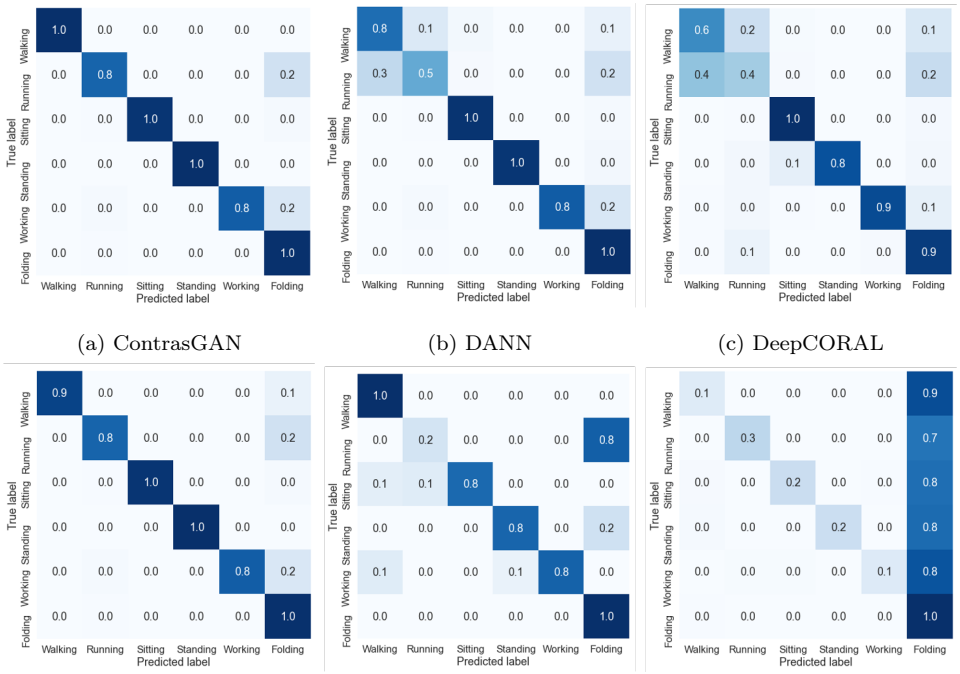
transfer, which sometimes produces lower accuracy than the lower-bound baseline.

Figures 9 and 10 present confusion matrices of ContrasGAN and the best performing techniques between two pairs of users. Most of the techniques can produce high accuracy as the transfer learning challenge is less significant, and ContrasGAN has consistently gained better accuracy on individual classes.

### *5.5. Cross-Sensor Experiment*

The cross-sensor experiment measures the accuracy of transferring an activity model from different types of sensors. This can be the most challenging transfer learning task in HAR as the source and target domains can have highly heterogeneous feature spaces, with different dimensions and different distributions. Here we perform the experiments on all the three datasets: W.PHONE-

PAMAP, W.PHONE-DSADS, W.WATCH-PAMAP, W.WATCH-DSADS, and PHONE-WATCH within WISDM. For each pair of datasets, we use the common set of activities between them as the prediction target. Figures 11a and 11b compare the micro- and macro-F1 scores between




(a) ContrasGAN (b) DANN (c) DeepCORAL

(d) DAN (e) ADADM (f) GFK

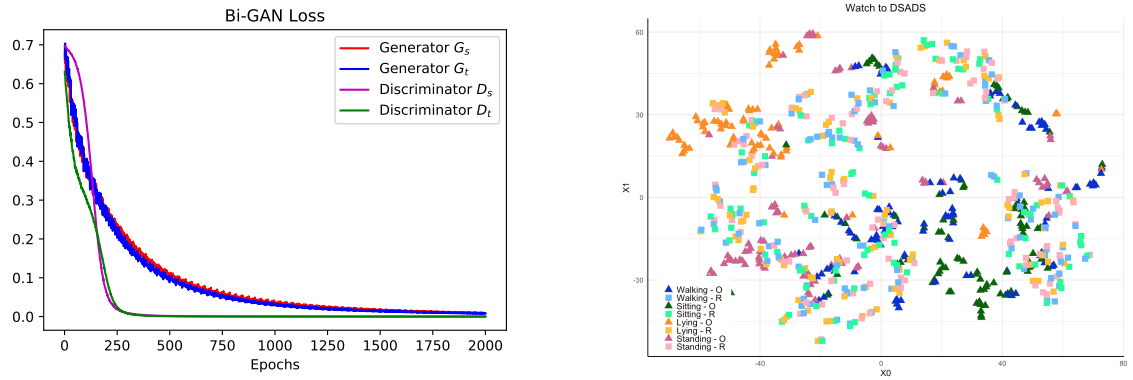
Figure 12: Confusion matrices on the PAMAP-W.PHONE task

ContrasGAN and the comparison techniques. ContrasGAN outperforms all these techniques and achieves the averaged micro- and macro-F1 scores as 0.81 and 0.80, which is higher than the second-best technique (ADADM) by 0.04 and 0.07 respectively.

Due to the heterogeneity in the feature space we cannot run the lower-bound baseline, and so we use upper-bound baseline to indicate the challenge on the target data. As shown in Figure 11a, PHONE-WATCH produces the lowest upper-bound accuracy (0.81). Most of the comparison techniques seem to struggle: for example, GFK and DAN only reaches 0.2 and 0.36 in macro-F1 scores. ContrasGAN achieves the macro-F1 scores of 0.63, 0.12 higher than ADADM, the second best technique. When observing the difference between micro- and macro-F1 scores, ContrasGAN has the smallest difference, indicating that it can better balance the majority and minority classes.

The cross-sensor experiment presents as the most challenging task as it deals with heterogeneous feature space transfer. Figure 13a presents the losses of Bi-GAN, which shows that both generators

and discriminators converge around 2000 epochs. However, it suffers overfitting as we employ Bi-GANs learnt at the epochs of 500, 1000, 1500, and 2000 for contrastive learning and the validation accuracy drops from 0.98 to 0.89 gradually. Thus, we take Bi-GAN at the epoch of 500. Figure 13b presents the t-SNE plot to project the original samples in DSADS and the generated samples from W.Watch to DSADS; that is, from the source feature space with a dimension of 90 to the target space with a dimension of 405. As we can see, Bi-GAN does not suffer much from mode collapse as it can generate samples from the same class in different clusters. However, the distribution of generated samples does not exactly match to the real samples. That's why we need to use contrastive learning to better align the samples, which is the key contribution of our work.




(a) Loss plot of Bi-GAN, showing the convergence of two generators and discriminators.

(b) t-SNE plot for original (O) samples in DSADS and reconstructed (R) samples from W.Watch to DSADS.

Figure 13: Loss and t-SNE plots on the task of W.WATCH-DSADS

Figure 12 presents the confusion matrices on the PAMAP-W.PHONE task. First of all, ContrasGANand DAN are better than DANN and DeepCORAL at differentiating similar activities, such as walking and running. Secondly, ADADM seems not stable and the performance might be affected by the mixup ratio. Thirdly, GFK cannot cope with high heterogeneity well and only predicts one activity.

### *5.6. Summary*

Here we summarise the following highlights yielded from the above experiments:

- ContrasGAN outperforms the state-of-the-art techniques on all the transfer learning tasks and over all the datasets with low computational footprint.

- With different components, contrastive loss makes the most significant contribution in ContrasGAN in improving recognition accuracy over Bi-GAN, suggesting that contrastive learning helps capture distinctive class-level features.
- On cross-body experiments, the task difficulty is more related to body positions rather than sides of sensors being worn on. ContrasGAN has outperformed all the comparison techniques on tasks at different difficulty levels.
- On cross-user experiments, the difference between source and target data is small, ContrasGAN has achieved the highest macro-F1 scores. DeepCORAL that under-performed on the other experiments has obtained the best median macro-F1 scores.
- On cross-sensor experiments, when dealing with the most challenging tasks in heterogeneous feature spaces and imbalanced class distribution, ContrasGAN can still produce consistently good performance, while some of the existing techniques have less stable performance and are more sensitive to the difficulty level of tasks.

## 6. Conclusion and Future Work

In this paper we have presented ContrasGAN, an unsupervised domain adaptation technique for heterogeneous feature spaces. Our technique requires only one well-annotated domain and can transfer the activity model from this domain to many other unlabelled domains. Thus, it can significantly reduce the labelling effort. We have evaluated ContrasGAN on three main transferring tasks. The results have shown that ContrasGAN can outperform state-of-the-art techniques on all these tasks and over various HAR datasets.

Our work suggests the potential of combining bi-directional GAN and contrastive learning in learning distinctive feature transfer and thus leading to effective activity model transfer. In the future, we will look into improving the robustness of this technique in the face of sensor noise. One limitation of our work is that we have only validated our approach on three datasets and we will assess its generality on more HAR datasets; for example, OPPORTUNITY [36] and Heterogeneity Activity Recognition datasets [41].

It is also interesting to consider the implications for this work outside HAR. Transfer learning is needed in many applications where there are differences between training and deployment scenarios, as well as when there is substantial variation in deployments. We hypothesise that approaches such

as ContrasGAN could be useful in applying machine learning to such diverse and challenging contexts, making classifiers more robust to the minor differences that often defeat current techniques. It will be interesting to explore these ideas more in the future.

## References


[1] Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. In *Proceedings of the First International Conference on Human Behavior Understanding*, HBU'10, pages 38–51, Berlin, Heidelberg, 2010. Springer-Verlag.

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

[3] Laila Bashmal, Yakoub Bazi, Haikel Alhichri, Mohamad Alrahhal, Nassim Ammour, and Naif Alajlan. Siamese-gan: Learning invariant representations for aerial vehicle image categorization. *Remote Sensing*, 10, 02 2018.

[4] Claudio Bettini, Gabriele Civitarese, and Riccardo Presotto. Personalized semi-supervised federated learning for human activity recognition. *ACM Trans. Intell. Syst. Technol*, 37(4), 2021.

[5] Nicola Bicocchi, Marco Mamei, and Franco Zambonelli. Detecting activities from body-worn accelerometers via instance-based algorithms. *Pervasive and Mobile Computing*, 6(4):482–495, 2010.

[6] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, 07 2006.

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[8] Chloë Brown, Jagmohan Chauhan, Andreas Grammenos, Jing Han, Apinan Hasthanasombat, Dimitris Spathis, Tong Xia, Pietro Cicuta, and Cecilia Mascolo. Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3474–3484, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Avijoy Chakma, Abu Zaher Md Faridee, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. Activity recognition in wearables using adversarial multi-source domain adaptation. *Smart Health*, 19:100174, 2021.

[10] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. A systematic study of unsupervised domain adaptation for robust human-activity recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(1), March 2020.

[11] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Comput. Surv.*, 54(4), May 2021.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th Inter-*

*national Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021.

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

[15] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. Transfer learning for activity recognition: A survey. *Knowl. Inf. Syst.*, 36(3):537–556, September 2013.

[16] Bo Dai and Dahua Lin. Contrastive learning for image captioning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 898–907, 2017.

[17] Shuyang Dai, Yu Cheng, Yizhe Zhang, Zhe Gan, Jingjing Liu, and Lawrence Carin. Contrastively smoothed class alignment for unsupervised domain adaptation, 2019.

[18] Kyle Feuz and Diane Cook. Modeling skewed class distributions by reshaping the concept space. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[20] Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, page 2066–2073, USA, 2012. IEEE Computer Society.

[21] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[22] Harish Haresamudram, Irfan Essa, and Thomas Plötz. Contrastive predictive coding for human activity recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(2), June 2021.

[23] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021.

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105. JMLR.org, 2015.

[26] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2208–2217. JMLR.org, 2017.

[28] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8896–8905. IEEE Computer Society, 2018.

[29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *2011 IEEE Transactions on Neural Networks*, number 2, pages 199–210, 2011.

[30] Ignacio Perez-Pozuelo, Dimitris Spathis, Emma A.D. Clifton, and Cecilia Mascolo. Chapter 3 - wearables, smartphones, and artificial intelligence for digital phenotyping and health. In Shabbir Syed-Abdul, Xinxin Zhu, and Luis Fernandez-Luque, editors, *Digital Health*, pages 33–54. Elsevier, 2021.

[31] Thomas PlÖtz. Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them. 54(6), 2021.

[32] Xin Qin, Yiqiang Chen, Jindong Wang, and Chaohui Yu. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), December 2019.

[33] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. Recent trends in machine learning for human activity recognition—a survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1254, 2018.

[34] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey, 2020.

[35] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109, June 2012.

[36] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 233–240, 2010.

[37] Andrea Rosales and Juan Ye. Unsupervised domain adaptation for activity recognition across heterogeneous datasets. *Pervasive and Mobile Computing*, 2020.

[38] A. R. Sanabria, F. Zambonelli, and J. Ye. Unsupervised domain adaptation in activity recognition: A gan-based approach. *IEEE Access*, 9:19421–19438, 2021.

[39] Max Schröder, Kristina Yordanova, Sebastian Bader, and Thomas Kirste. Tool support for the online annotation of sensor data. In *Proceedings of the 3rd International Workshop on Sensor-based Activity Recognition and Interaction*, iWOAR '16, pages 9:1–9:7, New York, NY, USA, 2016. ACM.

[40] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[41] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, page 127–140, New York, NY, USA, 2015. Association for Computing Machinery.

[42] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2058–2065. AAAI Press, 2016.

[43] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham, 2016. Springer International Publishing.

[44] Noeru Suzuki, Yuki Watanabe, and Atsushi Nakazawa. Gan-based style transformation to improve gesture-recognition accuracy. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), December 2020.

[45] Francesca Sylos-Labini, Yuri P. Ivanenko, Michael J. MacLellan, Germana Cappellini, and Richard E. Poppele. Locomotor-like leg movements evoked by rhythmic arm movements in humans. *PLOS ONE*, 2014.

[46] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. Exploring contrastive learning in human activity recognition for healthcare, 2021.

[47] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1541–1546. AAAI Press, 2011.

[49] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019. Deep Learning for Pattern Recognition.

[50] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S. Yu. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, Mar 2018.

[51] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[52] Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

[53] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), July 2020.

[54] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. *Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data*, page 1768–1778. Association for Computing Machinery, New York, NY, USA, 2020.

[55] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6502–6509. AAAI Press, 2020.

[56] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV 17*, 2017.

[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[58] James Y Zou, Daniel J Hsu, David C Parkes, and Ryan P Adams. Contrastive learning using spectral methods.

In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2238–2246. Curran Associates, Inc., 2013.