Smart Health 23 (2022) 100226

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

STranGAN: Adversarially-learnt Spatial Transformer for scalable human activity recognition

Abu Zaher Md Faridee [a,*], Avijoy Chakma [a], Archan Misra [b], Nirmalya Roy [a]

[a] Information Systems, University of Maryland, Baltimore County, United States of America
[b] Computing & Information Systems, Singapore Management University, Singapore

ARTICLE INFO

Keywords:
Domain adaptation
Wearable sensing
Learnable data augmentation
Adversarial learning
Generative modeling

ABSTRACT

We tackle the problem of domain adaptation for inertial sensing-based human activity recognition (HAR) applications -i.e., in developing mechanisms that allow a classifier trained on sensor samples collected under a certain narrow context to continue to achieve high activity recognition accuracy even when applied to other contexts. This is a problem of high practical importance as the current requirement of labeled training data for adapting such classifiers to every new individual, device, or on-body location is a major roadblock to community-scale adoption of HAR-based applications. We particularly investigate the possibility of ensuring robust classifier operation, without requiring any new labeled training data, under changes to (a) the individual performing the activity, and (b) the on-body position where the sensor-embedded mobile or wearable device is placed. We propose STranGAN, a framework that explicitly decouples the domain adaptation functionality from the classification model by learning and applying a set of optimal spatial affine transformations on the target domain inertial sensor data stream by employing adversarial learning, which only requires collecting raw data samples (but no accompanying activity labels) from both source and target domains. STranGAN's uniqueness lies in its ability to perform practically useful adaptation (a) without any labeled training data and without requiring paired, synchronized generation of source and target domain samples, and (b) without requiring any changes to a pre-trained HAR classifier. Empirical results using three publicly available benchmark datasets indicate that STranGAN(a) is particularly effective in handling on-body position heterogeneity(achieving a 5% improvement in classification F1 score compared to state-of-the-art baselines), (b) offers competitive performance for handling cross-individual variations, and (c) the affine transformation parameters can be analyzed to gain interpretable insights on the domain heterogeneity.

1. Introduction

Wearable devices, embedded with inertial sensors such as accelerometers & gyroscopes, permit unobtrusive and continuous monitoring, making them ideal platforms for *human activity recognition* (HAR) for applications in cognitive (Rastegari & Ali, 2020) and mental (Boukhechba et al., 2018) health assessment, predicting depression (Ware et al., 2020), sleep (Bobovych, Sayeed, Banerjee, Robucci, & Allen, 2020) and fitness monitoring (Milanko & Jain, 2020). While state-of-the-art gesture recognition techniques, either based on shallow or deep neural network (DNN) models, provide high accuracy, they continue to face a key challenge (Nweke, Teh, Al-Garadi, & Alo, 2018): *ensuring that the pre-trained ML models are robust enough to accommodate instance-specific variations in*

* Corresponding author.
E-mail addresses: faridee1@umbc.edu (A.Z.M. Faridee), achakma1@umbc.edu (A. Chakma), archanm@smu.edu.sg (A. Misra), nroy@umbc.edu (N. Roy).

https://doi.org/10.1016/j.smhl.2021.100226
Received 27 October 2021; Accepted 1 November 2021
Available online 29 November 2021
2352-6483/Published by Elsevier Inc.

the sensing data generated due to reasons such as (a) *User Diversity:* Different users execute the same activities in markedly different ways, due to differences in demographic and physical attributes, such as gender, height, and age; (b) *Position Diversity:* Variations in on-body placement (e.g., in a trouser vs. shirt pocket) often generate significantly different sensor patterns for the same activity or gesture.

The commonplace approach for tackling such heterogeneity, by using *instance-specific* labeled data to build individual & device-specific classifiers, is clearly infeasible for practical societal-scale deployment. Instead, significant research has focused on techniques for *automated domain adaptation* – i.e., a transfer learning-based mechanism that allows a model trained on one domain to flexibly evolve and cater to data collected under a different domain/context, *while requiring modest-to-no labeled training data from the target domain*. A variety of approaches for such HAR-oriented domain adaptation have been suggested in recent years, including techniques that (a) employ transfer learning to modify a source domain model with only modest amounts of target domain labeled data (Khan, Roy, & Misra, 2018; Qin, Chen, Wang, & Yu, 2019); (b) map domain-dependent sensor values to a domain-independent, common low-dimensional latent space (Jeyakumar, Lai, Suda, & Srivastava, 2019); and (c) use adversarial learning techniques to learn a set of robust features that are invariant to data from either training (source) or test (target domains) (Ganin & Lempitsky, 2014). In general, these techniques suffer from at least one of three key limitations: (a) They often require at least modest amounts of labeled target domain data, with their performance degrading sharply in the absence of *any* labeled data; (b) They require capture of *synchronously paired data*–i.e., the simultaneous capture of target and source domain data streams, as a means of implicit labeling (e.g., Akbari and Jafari (2019) and Jeyakumar et al. (2019)); (c) They require *modification* of the gesture classification model—while not technically difficult, this requirement presents practical difficulties as many ML-based activity models are now bundled as standard executable binaries by either OS or App developers.

In our work, we propose and develop a domain adaptation framework, called *STranGAN* (*Spatial Transformation-Driven GAN*), for robust HAR that concurrently exhibits three features: (i) no requirement for labeled target domain data, (ii) no modification of the baseline HAR classifier (hence no requirement for source labels during domain adaptation) (iii) no need for paired or synchronous data. Our proposed approach works by explicitly transforming the raw data stream from the target domain, via a *Spatial Transformer* component (modeled with a convolutional neural network), to have a similar representation as to the source domain, which in turn allows a pre-existing classifier to operate unmodified on this transformed data stream. Unlike past work (Akbari & Jafari, 2019; Jeyakumar et al., 2019) that required paired or synchronous cross-domain data, *STranGAN* trains and tunes the parameters of this Spatial Transformer without any such sample correspondence. In *STranGAN*, the Spatial Transformer is trained in an adversarial fashion, in tandem with a domain Discriminator Network, to simultaneously ensure that (a) target domain sensor data samples that are similar to source domain data are left unmodified, and (b) target domain samples that differ from corresponding source domain data are modified to be distributionally similar. While *STranGAN* can be used in a variety of *classifier bootstrapping* (Abdallah, Gaber, Srinivasan, & Krishnaswamy, 2015) scenarios, we principally evaluate its effectiveness in supporting the robust use of a pre-trained classifier (trained on a group of individuals) on another individual or a different body position.

Key Contributions: We make the following key contributions-

- *Adaptation without Target Domain labels or Modification of Pre-Trained Classifier:* We motivate and propose *STranGAN*, a novel approach for automated HAR domain adaptation that requires no modification to a pre-trained HAR classifier, and dispenses with the need for either labeled target domain training data or synchronously-generated source domain data samples. *STranGAN* uses an adversarial framework to train a Spatial Transformer network that can automatically learn how to generate the affine transformation (such as translation & rotation) parameters. This affine transformation modifies the raw (target domain) inertial sensor data stream to match the discriminative characteristics needed for accurate HAR classification, thereby decoupling the adaptation mechanism from the underlying training of the HAR model.
- *Demonstration of StranGAN's Efficacy and Robustness:* We demonstrate the efficacy of *STranGAN* for both cross-person and cross-body position adaptation using activity labels and data from 3 distinct benchmark datasets: HHAR (Stisen et al., 2015), PAMAP2 (Reiss & Stricker, 2012), and OPPORTUNITY (Roggen et al., 2010). The 3 datasets capture a range of low-level human activities/gestures (sitting, standing, lying etc.) more complex short-lived transient activities (jumping, ascending and descending stairs etc.) and typical ADLs (Activities of Daily Living), and are characterized by heterogeneity across users with different body positions. We experimentally establish that *STranGAN* either outperforms or is competitive to prior baselines for cross-person and cross-body position adaptation. Moreover, in comparison to alternative data transformation approaches (Akbari & Jafari, 2019) that exhibit a steep 38%–50% performance loss in the absence of paired (synchronous) data, *STranGAN* maintains its classification performance without requiring such paired samples.
- *Rendering Interpretability During Domain Adaptation*: *STranGAN* performs domain adaptation by aligning the raw sensor feature spaces with learnable affine transformation parameters. These parameters can be directly traced to the particular type of transformation (e.g., translation/shift vs scale) and can help us find the reason for the domain discrepancies with respect to the raw features. For example, these parameters may tell us that the *most* of the x-axis of the accelerometer sensor data in the target domain has an amplitude offset of +0.2 and a magnitude factor of 1.25× with respect to source domain x-axis samples, and therefore needs to be *corrected* so that the source classifier will work without modification. In contrast, traditional domain adaptation models (Akbari & Jafari, 2019; Chen, Wang, Huang, & Yu, 2019; Pei, Cao, Long, & Wang, 2018) align the domains in latent feature space which prevent them gaining similar interpretable insights.

Our proposed approach helps plug an important gap in developing practical HAR-based wearable smart health applications, as it allows pre-trained, HAR classifiers to be re-utilized across significant individual and body position specific variations in sensor data patterns

2. Related works

We review the variety of domain adaptation approaches proposed for supporting cross-domain adaptation, both generically and specifically for HAR applications and highlight the major differences between them and *STranGAN* .

2.1. Domain adaptation via feature alignment

With roots in natural language processing and computer version (Blitzer, McDonald, & Pereira, 2006; Duan, Tsang, & Xu, 2012), domain adaptation techniques have recently received more attention for HAR applications (Cook, Feuz, & Krishnan, 2013). Based on the method of feature extraction, these approaches can be divided between *shallow* and *deep* models. For shallow models, transfer learning approaches seek to align *statistics* of selected features across the source and target domains. TCA (Pan, Tsang, Kwok, & Yang, 2010), one of the oldest approaches aligns the marginal distributions of the extracted features of both two domains by transferring them into Reproducing Kernel Hilbert Space (RKHS) manifold using Maximum Mean Discrepancy (MMD) (Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2012) as the primary distance metric. JDA (Long, Wang, Ding, Sun, & Yu, 2013) employs a pseudo-labeling technique to align conditional distributions of multiple domains, while BDA (Wang, Chen, Hao, Feng, & Shen, 2017) incrementally tunes the importance of the marginal and conditional distribution discrepancies for better performance. The state-of-the-art STL (Chen et al., 2019) method for HAR domain adaptation employs pseudo-labels in the target domain to iteratively determine the best choice among alternative source domains and then employs stratified activity transfer to tune the selected model to match the target domain's characteristics.

DNN-based approaches also seek to perform feature alignment, albeit using intermediate representations automatically learned by DNNs instead of hand-crafted features. For example, Deep Domain Confusion (DDC) (Tzeng, Hoffman, Zhang, Saenko, & Darrell, 2014), Joint Adaptation Network (JAN) (Long, Zhu, Wang, & Jordan, 2017) and Joint Distribution Adaptation (JDA) (Long et al., 2013) are popular domain adaptation architectures, originating from the computer vision literature, that aim to minimize maximum mean discrepancy (*MMD*) distance between the final deep layers. More recent work has explored this approach of latent space alignment, specifically targeted to the HAR domain adaptation problem. In HDCNN (Khan et al., 2018), the authors assume that the relative *distribution* of weights in different DNN layers remains invariant and thus seek to minimize the Kuhlbeck–Leibler distance between DNN weights. AugToAct (Faridee, Khan, Pathak, & Roy, 2019) extends the idea by employing a denoising auto-encoder on augmented (scaled, rotated) samples to extract self-supervised features, thereby minimizing the need for labeled data in the source domain. DGDA (Akbari & Jafari, 2019) proposed a generative approach using variational auto-encoders while keeping the underlying assumptions similar to the prior two discriminative approaches. While such a generative formulation may extract stochastic features that improve the models robustness, this approach requires paired, synchronized data across source and target domains.

2.2. Adversarial approaches

Various adversarial learning-based approaches have recently attempted to execute domain adaptation process by automatically extracting features that are both (i) indifferent to the domain discrepancy, and (ii) effective at classifying a given task. In particular, DANN (Ganin et al., 2016) achieves this by jointly optimizing the shared feature space between two adversarial classifiers: (i) the task-specific *label predictor* and (ii) *domain discriminator* that predicts between the source and the target domain. MADA (Pei et al., 2018) extends this approach by constructing multiple, *label/class-specific* domain discriminators that helps ensure that the class-specific alignments are also achieved. While both DANN and MADA jointly optimize the classifier and the domain discriminator, *STranGAN* differs via virtue of its decoupled design that separates the domain adaptation mechanism from the process of classifier training.

Such adversarial learning has also been combined with generative approaches to support unsupervised domain adaptation, without explicitly attempting to align the latent feature space, for vision and speech applications. CyCADA (Hoffman et al., 2018) developed an image to image translation architecture that operates without requiring paired samples, while Mic2Mic (Mathur, Isopoussu, Kawsar, Berthouze, & Lane, 2019) used a similar generative adversarial architecture to overcome microphone variability in speech systems. Both of these approaches employ cyclic consistency as a regularization mechanism to overcome the lack of paired samples in the source and target domain. Most recently, Soleimani and Nazerfard (2019) have proposed a GAN based architecture that achieves cross-subject transfer by transforming the source data to be distributionally similar to target data, while simultaneously training a classifier using the labels of these transformed source data samples. In contrast to these architectures, our proposed *STranGAN* approach (a) does not modify or retrain an existing classifier, and (b) also offers faster model learning by using *affine transformation* that avoid costly (two way) cyclic consistency computation.

2.3. Learnable transformation

Most domain adaptation methods (Chen et al., 2019; Khan et al., 2018; Long et al., 2013; Pan et al., 2010) assume that the distributions of input samples, across both source and target domains are similar under a common latent space. We hypothesize that the dissimilarity between the target and source domains within a short time window (e.g., 1–3 s) can be approximated via a series of *affine* transformations (e.g., scaling, translation, rotation) in the raw data (e.g. accelerometer, gyroscope) space. When viewed across a longer time-frame, the dissimilarity between the source and target domain samples (e.g. arising from placement in different body positions or personal variations) are mostly non-linear in nature and can be difficult to approximate. However,

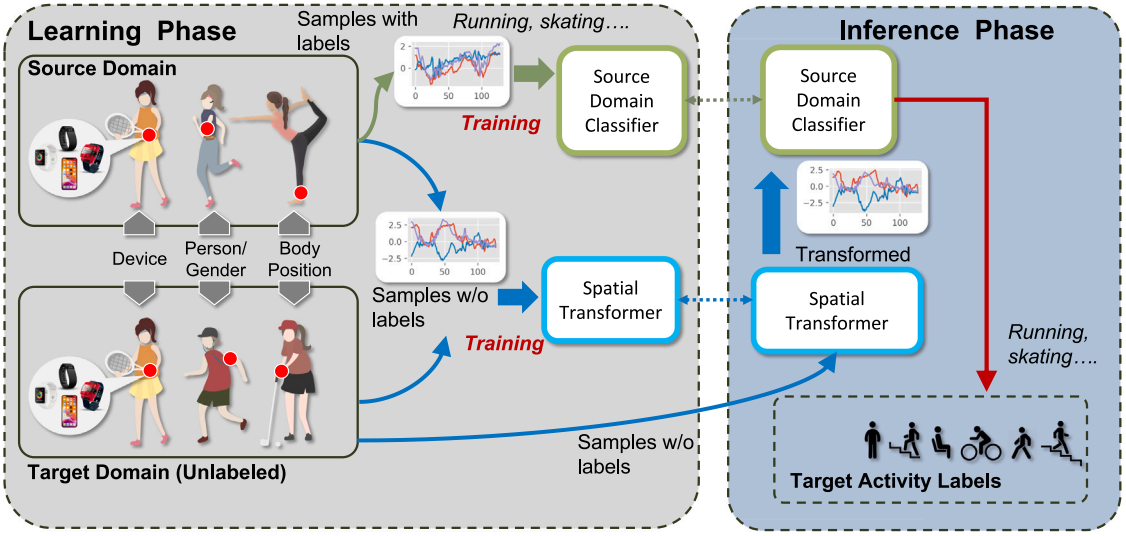


Fig. 1. Overview of *STranGAN* framework.

by choosing short (e.g. 1–3 s) and highly overlapping (80%–90%) windows, we can model these *temporally* non-linear variations with *local/piece-wise* affine-transformations. Modern convolutional architectures have been shown to be robust against moderate variances in *scale* and *translation*, but are unable to handle more pronounced (especially *rotational*) transformations that are likely to arise due to variations in on-body positions and cross-user behavior. Interestingly, such affine transformations (in the form of scaling, shifting and rotation within a short time window) have been widely used as a form of data augmentation to improve the generalizability of DNN models (Mathur et al., 2018; Qi, Su, Mo, & Guibas, 2017b; Um et al., 2017), but finding the optimum parameters for such augmentation have been shown to be non-trivial (Faridee et al., 2019). In this spirit, our work is motivated by the concept of the Spatial Transformer Network (STN) (Jaderberg, Simonyan, Zisserman, & Kavukcuoglu, 2015), a learnable module specially designed to provide CNN models the ability to learn invariance to affine transformations, including *translation, scaling, rotation, and more generic warping*. We hypothesize that such learnable transformations (e.g. STN) are particularly well suited to our HAR applications, as the activities or gestures of interest essentially involve distinctive spatial movements of one or more limbs and thus consequently manifest as differences in the spatial characteristics of the underlying sensor data. However, previous works have also reported severe overfitting (Finnveden, Jansson, & Lindeberg, 2020; Jaderberg et al., 2015) when such modules are directly embedded in a standard classification pipeline; our work is directly aimed at mitigating that effect with an adversarial learning setup.

3. Overview

Fig. 1 depicts a high-level overview of our proposed use of *STranGAN's* Spatial Transformer for domain adaptation. Our vision for *STranGAN* is to support the robust adaption of an unmodified, pre-existing HAR classifier to possible variations across individuals and body position (where the device is located). Our approach does not require any paired samples (unlike, for example, Akbari and Jafari (2019) and Jeyakumar et al. (2019)) to train and adapt across dissimilar source and target domains, and is able to operate as long as the set of class labels (i.e., the set of activities) and the relative distribution (in terms of sample count) of such labels are identical across both source and target domains. *STranGAN* approach assumes the pre-existence of an appropriate HAR classifier that has been conventionally trained in a supervised fashion, using labeled training data in a specific source domain. Core to the domain adaptation capability of *STranGAN* is the *Spatial Transformer* component, which once trained, takes input sensor data in the target domain and performs appropriate inference of spatial transformation of this data to generate a modified data stream, which is then fed to the aforementioned HAR classifier for appropriate activity recognition.

Our key innovation is on utilizing an adversarial framework to *train* this Spatial Transformer, so that it can learn the appropriate transformation to ensure the modified sensor data stream is compatible with the training data used previously for training the HAR classifier. As illustrated in Fig. 1, this training process involves the use of *unlabeled* data from the target domain, along with *unlabeled* (i.e., no activity labels) source domain data. It is key to observe that the Spatial Transformer does not require any actual activity labels on either the source or target domain data, but merely needs such data to be marked with just their "Source" or "Target" domain origin. Based on the principles of adversarial training, the Spatial Transformer is coupled with a Discriminator module and trained in tandem: the Spatial Transformer attempts to learn the right set of (spatial) transformations on the source or target domain data such that the two transformed data streams look indistinguishable to the Discriminator, while the Discriminator evolves to correctly distinguish whether the provided transformed data stream belongs to the source or target domain.

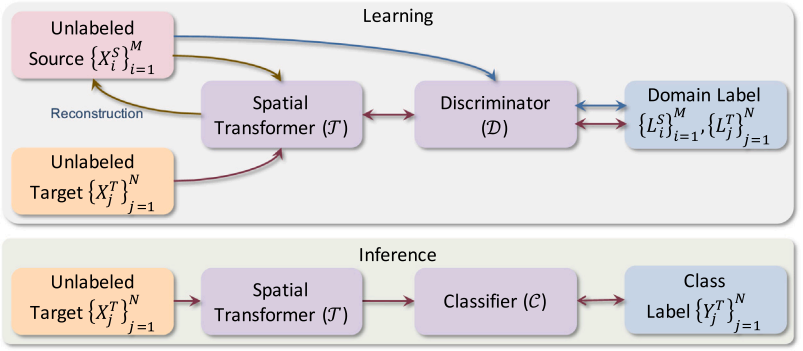


Fig. 2. Overview of the adversarial transformer architecture.

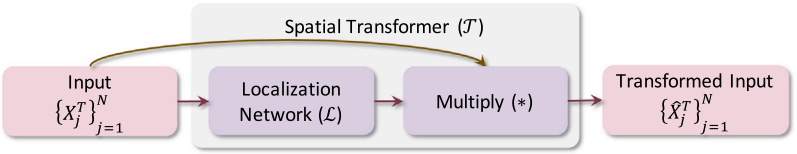


Fig. 3. Internal organization of the spatial transformer.

4. Functional components & adversarial learning

In this section, we discuss the core principles and different components of our proposed *STranGAN* framework including the **Spatial Transformer** ($\mathcal{T}$), **Discriminator** (D) and **Classifier** (C), and explain how these components are involved in both the learning and inference stages (as illustrated in Fig. 2).

We first elaborate the motivation behind our choice of the **Spatial Transformer** ($\mathcal{T}$) module. For wearable devices used to perform various locomotive and gestural activities, we assume that the sensor data undergoes different transformations, caused by changes in the on-body position of the device, the individual-specific differences in the motion trajectories associated with different activities or perhaps by variations in the characteristics of the devices used. We aim to learn such transformations between the source and target domain data, and thereby apply an inverse transform to modify the target sensor data streams to match the characteristics of the source domain.

We postulate that for the data collected via IMU (inertial) sensors (such as 3-axis accelerometer or a 3-axis gyroscope), the resulting variations can be modeled as *affine transformations*, a form of linear functional mapping between two geometric (affine) spaces that preserve points, straight and parallel lines as well as the ratios between points (Berger, 2009). Linear transformations such as *scaling, shear, reflection, and rotation* can be expressed as matrix multiplications in Cartesian coordinates. Without the loss of generalizability, we illustrate the affine transformation model with a data stream from a 3-axis accelerometer. Given an input acceleration vector in the target domain $u = (u_x, u_y, u_z)$, a transformed vector $v = (v_x, v_y, v_z)$ can be represented as the result of a matrix multiplication by a 9-element (3×3) matrix A_θ as follows:

$$\begin{pmatrix} v_x \\ v_y \\ v_x \end{pmatrix} = A_\theta \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \tag{1}$$

However, Eq. (1) cannot model or represent additional non-linear transformations, such as translation or perspective projections. Hence, we interpret u in a Homogeneous coordinate (Bloomenthal & Rokne, 1994) where $u = (wu_x, wu_y, wu_z, w)$ and $w = 1$, which allows us to add three more parameters in a 3×4 matrix A_θ and accommodate such transformations. Given such an A_θ, the final affine transformation matrix in homogeneous coordinates is modeled as:

$$\begin{pmatrix} v_x \\ v_y \\ v_x \end{pmatrix} = A_\theta \begin{pmatrix} wu_x \\ wu_y \\ wu_z \\ w \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix} \tag{2}$$

Affine transformations are associative but not commutative (House & Keyser, 2016), which allows a single transformation matrix to capture an arbitrary sequence of transformations. For example, if we want to *scale* u by a factor $s = (s_x, s_y, s_z)$ and *translate* the

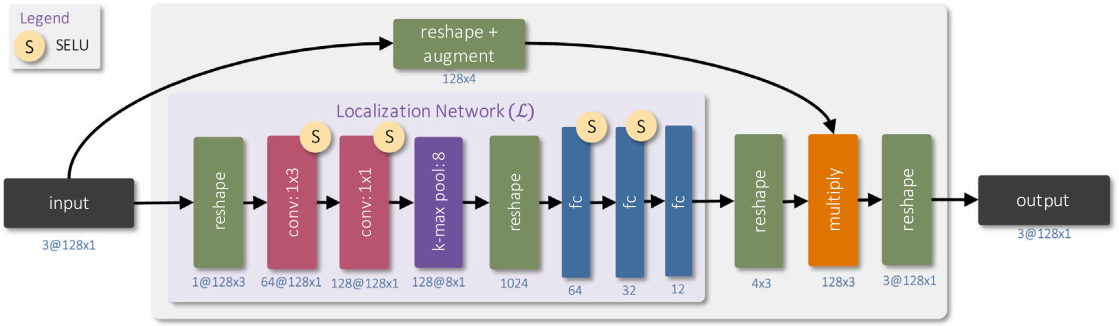


Fig. 4. Spatial transformer ($\mathcal{T}$) module (The shape of the features after each operation is shown at the bottom of each layer. Activation functions are shown with a yellow circle.).

resulting vector by amount $t = (t_x, t_y, t_z)$, then the final output is given by,

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{bmatrix} s_x & 0 & 0 & t_x \\ 0 & s_y & 0 & t_y \\ 0 & 0 & s_z & t_z \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix} = \begin{bmatrix} s_x u_x + t_x \\ s_y u_y + t_y \\ s_z u_z + t_z \end{bmatrix} \tag{3}$$

More generally, the twelve parameters of A_θ help express the resulting composite transformation (any series of operations among *rotation*, *scaling*, *sheer*, *mirror*, *translation*, or any other affine transformation) on the input data stream. In this framework, domain adaptation involves (a) first *learning* the optimal set of such parameters that provide the best inverse transformation—i.e., transform the target domain data samples to be *distributionally* identical to source domain data, and (b) subsequently, at runtime, applying this learned matrix on individual test data samples to create "source domain-equivalent" sensor data. Rather than performing manual selection of these parameters (as done in data augmentation practices Faridee et al., 2019; Um et al., 2017), we propose a *Spatial Transformer* module that automatically *learns* these parameters from the unlabeled (source and target domain) data streams.

In order to validate whether the *spatial transformer* ($\mathcal{T}$) module is able to effectively transform the target data to match the distribution of the source data, we introduce a *domain discriminator* (D) which helps predict the origin of a given sample between source (real) and target (fake) domain. The *domain discriminator* (D) and *spatial transformer* ($\mathcal{T}$), thus have an *adversarial objective*, similar to a *Generative Adversarial Network* (Goodfellow et al., 2014). A more in depth layer by layer breakdown of the $\mathcal{T}$, D and C networks are provided in the following.

4.1. Spatial transformer ($\mathcal{T}$)

Fig. 4 shows the internal details of our *spatial transformer* ($\mathcal{T}$) network, which works as the *generator* in our adversarial learning setup. The module itself consists of a small convolutional *localization network* ($\mathcal{L}$) that takes the IMU data stream of length 128 and regresses 12 transformation parameters for each windowed samples. This module consists of two convolution layers, followed by a K-Max Pooling layer (Kalchbrenner, Grefenstette, & Blunsom, 2014) layer which gives us a simple yet configurable way to modify the expressive power of the generator network. Unlike traditional convolutional architectures used in most HAR classification tasks, where the inertial axes are treated as channels, we treat them as a spatial dimension and employ a 1×3 convolution at the first layer to capture the local dependency between the axes. We also employ 1×1 convolution as channel pooling (Goodfellow, Warde-Farley, Mirza, Courville, & Bengio, 2013; Lin, Chen, & Yan, 2013; Qi, Su, Mo, & Guibas, 2017a) on the second convolution layer to further capture the interaction between the features. The output from the pooled layers is then fed through 3 fully connected layers with decreasing neurons $(64, 32, 12)$ to achieve the 12 transformation parameters as depicted in Eq. (2). At the same time, the input data window is augmented by one extra channel with a value of 1 (Eq. (2), the top part of Fig. 4). This augmented input is multiplied with the (4×3) output of the regressed transformation values to arrive at a 3 channel 128 length window output—note that this output has the same dimensions as the original input. In all convolutional and fully connected layers (except the last one), we use SELU (Klambauer, Unterthiner, Mayr, & Hochreiter, 2017) activation function for its self normalizing property, which helps reduce the need for a dedicated normalization layer (e.g., batch normalization).

4.2. Domain discriminator (D)

The *discriminator network* (D), follows a convolutional architecture as shown in Fig. 5. The input sample window is passed through 3 successive convolutional layers, each with a filter shape of (9×1), and each followed by a (2×1) max-pooling layers. The convolution feature extractor block is followed by two fully connected layers of size 64 and 1. All the convolutional layers and fully connected layers, except the last one, use a SELU (Klambauer et al., 2017) activation function. The last fully connected layer is accompanied by a `Sigmoid` activation function.

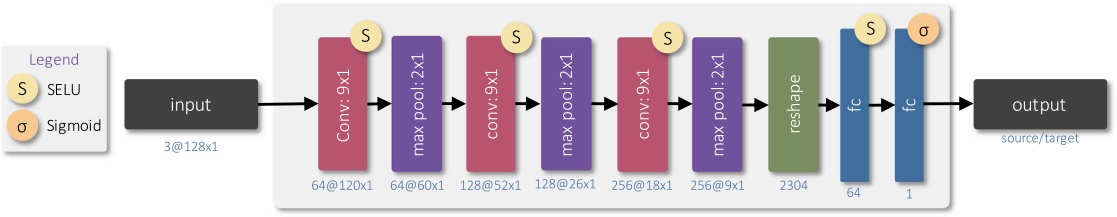


Fig. 5. Domain discriminator module.

4.3. Classifier (C)

The classifier is only utilized during the inference stage when the transformed samples are passed to the unmodified source classifier to infer the target domain class labels. This classifier itself is trained separately (outside the scope of *STranGAN*), using source domain labeled data, via a standard supervised learning process. In our experiments, we used a deep convolutional architecture which is, coincidentally, very similar to the Domain Discriminator described in Fig. 5, the only difference being the replacement of the `Sigmoid` activation function with `SoftMax` activation and the change in the number of neurons to reflect the number of activity class labels. While the recent literature (Hammerla, Halloran, & Plötz, 2016; Li, Shirahama, Nisar, Köping, & Grzegorzek, 2018; Ordóñez & Roggen, 2016; Wang, Chen, Hao, Peng, & Hu, 2019) suggest that the hybrid architectures combining both convolutional and recurrent layers (e.g., LSTM) tend to outperform pure convolutional approaches, in this work our focus was not investigating such hybrid architectures for our classifier (and gain unfair advantage) as all the state-of-the-art baseline domain adaptation models we compared against (Section 6.2) used convolutional or kernel based classifiers.

5. Learning and inference with *STranGAN*

We now describe the use of adversarial learning for training the Spatial Transformer component. Given M unlabeled source domain samples, $Data_S = \{X_i^S\}_{i=1}^M$ and N unlabeled target domain samples, $Data_T = \{X_j^T\}_{j=1}^N$, the objective of the learning algorithm for *STranGAN* is to find the optimum parameters A_θ for the *Spatial Transformer* $\mathcal{T}$, across each of the target samples $Data_T$, so that distributions of $Data_S$ and $\mathcal{T}(Data_T)$ appear similar. The adversarial learning strategy employs competition between the $\mathcal{T}$ and D networks to learn A_θ. Given a mini-batch size of m, in each iteration of our mini-batch gradient descent, we draw m samples from source domain, $B_S = \{x_i^S\}_{i=1}^m$ and target domain, $B_T = \{x_i^T\}_{i=1}^m$. The target batch B_T is passed through the $\mathcal{T}$ network to get the transformed samples, $\mathcal{T}(B_T)$.

The $\mathcal{T}$ network consists of a small convolutional *localization network* (Jaderberg et al., 2015), $\mathcal{L}$ (shown in Fig. 3) that analyses the input batch B_T and regresses 12 parameters for the matrix A_θ (Eq. (2)). The $\mathcal{T}$ network then multiplies A_θ with an augmented (1 padded) input batch $\tilde{B_T}$ to provide the its final output as shown in Eq. (4).

$$\mathcal{T}(B_T) = \tilde{B_T} \times A_\theta = \tilde{B_T} \times \mathcal{L}(B_T) \tag{4}$$

At the discriminator, we define a cross-entropy loss L_D that tries to maximize the log-likelihood of assigning the correct label (0→target, 1→source) between the source and target domain. Hence, the log-likelihood for the source domain batch becomes $\frac{1}{m}\sum_{i=1}^m \log D(B_S)$, where $D(.)$ denotes the binary classification output of the D network; the log-likelihood for the target domain is $\frac{1}{m}\sum_{i=1}^m \log(1 - D(\mathcal{T}(B_T)))$, similarly. The overall objective function for the discriminator can be defined as minimizing the total cross-entropy loss:

$$L_D = \max_D \frac{1}{m}\sum_{i=1}^m \log D(B_S) + \log(1 - D(\mathcal{T}(B_T))) \tag{5}$$

For the $\mathcal{T}$ network, the objective is to minimize the log-likelihood i.e., $\log(1 - D(\mathcal{T}(B_T)))$ for the target domain batch. However, optimizing the $\mathcal{T}$ network with solely this simple objective has a major flaw: the $\mathcal{L}$ network is then free to assign any arbitrary values to A_θ so that the resulting $\mathcal{T}(B_T)$ can fool the Discriminator (D). These arbitrary transformations can lead to the resulting samples ($\mathcal{T}(B_T)$), ending up in a completely different label boundary which would cause the classifier to perform poorly, thereby defeating our main domain adaptation objective. It is also possible for the $\mathcal{T}$ network to focus on only a small set of values of A_θ to defeat D, resulting in every $\mathcal{T}(B_T)$ looking the same (a phenomenon that is known as *mode collapse* in the GAN literature). To mitigate these issues, we take inspiration from Lee and Lee (2020), Srivastava, Valkov, Russell, Gutmann, and Sutton (2017) and introduce a *reconstruction* objective. We add a minimization objective of the $L2$ norm between the source samples B_S and transformed source samples, $\mathcal{T}(B_S)$. Such *Reconstruction* loss, defined as $\|B_S - \mathcal{T}(B_S)\|_2$, has the additional effect of training the $\mathcal{T}$ network to leave unmodified those data samples that have a very similar distribution to the source domain. This regularization forces the $\mathcal{T}$ network to apply the affine transformations selectively only to the target samples that do not share distributional similarity with the source domain; the magnitude of the transformations becomes proportional to this divergence. Accordingly, the final objective of the $\mathcal{T}$ network can be described as follows:

$$L_\mathcal{T} = \min_\mathcal{T} \frac{1}{m}\sum_{i=1}^m \log(1 - D(\mathcal{T}(B_T))) + \gamma\|B_S - \mathcal{T}(B_S)\|_2 \tag{6}$$

We define γ as a tunable hyper-parameter to control the strength of the regularization term. The full learning algorithm is described in Algorithm 1. To further ensure that $\mathcal{T}$ network learns meaningful transformations, the final layer of the $\mathcal{L}$ network is initialized with identity transformation values for A_θ, i.e.:

$$A_\theta{}^{init} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

It is also noteworthy that this training phase does not necessarily require either the source nor the target data to be scaled or normalized during a pre-processing phase, as the $\mathcal{T}$ network automatically learns such normalization parameters during the inference stage. This is in contrast to all major existing state-of-the-art techniques that require a separate and explicit initial normalization step.

Algorithm 1: Learning Algorithm for *STranGAN*

Input: Unlabeled Source dataset, $Data_S = \{X_i^S\}_{i=1}^M$
Input: Unlabeled Target dataset, $Data_T = \{X_j^T\}_{j=1}^N$
Input: Number of iterations, N_{Iter}
Input: Minibatch size, m
Output: Spatial transformer, $\mathcal{T}$
Output: Domain discriminator, $\mathcal{D}$
for k *in* $1 : N_{Iter}$ **do**
1 Get m samples from source domain, $B_S = \{x_i^S\}_{i=1}^m$
2 Get m samples from target domain, $B_T = \{x_i^T\}_{i=1}^m$
3 Update domain discriminator $\mathcal{D}$ with following objective:

$$L_D = \max_{\mathcal{D}} \frac{1}{m} \sum_{i=1}^{m} \log \mathcal{D}(B_S) + \log(1 - \mathcal{D}(\mathcal{T}(B_T)))$$

4 Update spatial transformer $\mathcal{T}$ with following objective:

$$L_{\mathcal{T}} = \min_{\mathcal{T}} \frac{1}{m} \sum_{i=1}^{m} \log(1 - \mathcal{D}(\mathcal{T}(B_T))) + \gamma \|B_S - \mathcal{T}(B_S)\|_2$$

Inference: During the inference stage, the incoming test samples $B_T = \{x_i^T\}_{i=1}^m$ from the target domain are first passed through the $\mathcal{T}$ network to infer the transformed samples $\mathcal{T}(B_T)$. These transformed samples ideally have similar marginal distribution as the labeled source samples, which we assume have been used to train a classifier C. As such, target labels $\{\hat{y}_i^T\}_{i=1}^m$ can be easily inferred by using the *unmodified* source classifier C on the transformed samples so that

$$\{\hat{y}_i^T\}_{i=1}^m = C(\mathcal{T}(B_T)) \tag{7}$$

6. Datasets, baselines & implementation details

In the following section, we discuss the details of a number of representative Activities-of-Daily-Living (ADL) datasets, which we use to demonstrate the efficacy of our proposed *STranGAN* approach. We also summarize alternative approaches that help provide competitive baselines.

6.1. Datasets

We showcase the effectiveness of our proposed framework on three publicly available datasets: (i) PAMAP2 Physical Activity Monitoring Dataset (Reiss & Stricker, 2012), (ii) OPPORTUNITY Activity Recognition Dataset (Roggen et al., 2010), and (iii) Heterogeneous Activity Recognition Dataset (HHAR) (Stisen et al., 2015). We choose these datasets as they provide a wide variety of representative data that helps us study the performance of adaptation across two major types of variations in the target domain: (i) person, and (ii) body position. A summary of the datasets is provided in Table 1 and the label distributions (full dataset) are shown in Fig. 6. Pre-processing details for the datasets are provided in Section 6.3.

6.2. Baselines

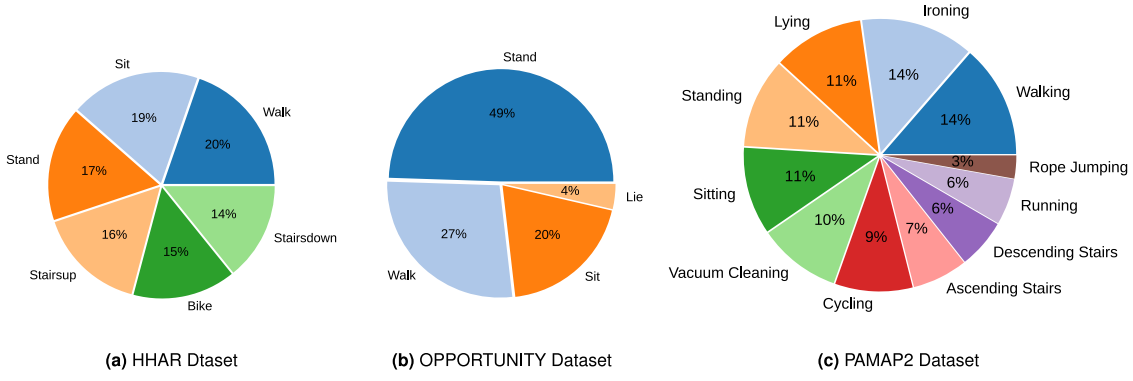
To demonstrate the effectiveness of *STranGAN*, we compare its performance with 4 different baselines: (i) STL (Chen et al., 2019), (ii) MADA (Pei et al., 2018), (iii) DGDA (Akbari & Jafari, 2019), (iv) SA-GAN (Soleimani & Nazerfard, 2019). Each of these baselines (already discussed in Section 2) were chosen carefully the reflect state-of-the-art within their respective domain adaptation

Table 1
Summary of three public datasets used in our experiments.

Dataset	Subjects	#Positions (Names)	#Labels	Device sampling frequency
PAMAP2	8	3 (Wrist, Chest, Ankle)	11	100 Hz
OPPORTUNITY	4	5 (BACK, Right Upper Arm (RUA), Right Left Arm (RLA), Left Upper Arm (LUA), Left Lower Arm (LLA))	4	30 Hz
HHAR	9	1 (Waist)	6	50 Hz, 100 Hz, 150 Hz, 200 Hz

Table 2
Feature matrix comparison of the baselines.

	Unpaired samples	Decoupled classifier	Primary mechanism	Interpretabale
STL (Chen et al., 2019)	☑	☒	Subspace manifold alignment	☒
MADA (Pei et al., 2018)	☑	☒	Adversarial	☒
DGDA (Akbari & Jafari, 2019)	☒	☑	Generative	☒
SA-GAN (Soleimani & Nazerfard, 2019)	☑	☒	Generative adversarial	☒
***STranGAN* (ours)**	☑	☑	Generative adversarial	☑



(a) HHAR Dtaset **(b)** OPPORTUNITY Dataset **(c)** PAMAP2 Dataset

Fig. 6. Label distribution of the three benchmark datasets.

mechanisms (shown in Table 2). We can also notice that, in contrast to *STranGAN*, each of the baselines requires at least one of either (a) synchronized samples or (b) classifier adaptation. Usage of learnable affine transformation to perform domain adaptation also provides *STranGAN* the ability to showcase how it performs the feature alignments, increasing its interpretability (discussed in detail in Section 7.3) which is in contrast to the other baselines. Implementation details for the baselines are provided in Section 6.3.

6.3. Implementation details

6.3.1. Runtime environment

We conducted our experiments on a Linux Server (Ubuntu 18.04) running on Intel Core i7-6850K CPU and 64 GB DDR4 RAM, with an Nvidia 1080Ti Graphics card (11 GB VRAM). Python was used for all coding tasks except for STL baseline where MATLAB was used. For the signal processing, filtering, and shallow feature extraction tasks, we used `scikit-learn`, `scipy` and `numpy` libraries. For deep learning tasks, we used `PyTorch`.

6.3.2. Baseline implementations

The STL (Chen et al., 2019) baseline involved the use of the following predefined temporal and frequency domain features: (i) mean and standard deviation of each accelerometer axes, (ii) their first and second order derivatives and magnitudes, (iii) co-variance between the axes, (iv) yaw, pitch, and roll angle derived from the axes, (v) spectral centroid, spectral entropy derived through FFT and zero-crossing rate. We directly utilized the code provided by STL authors.[1] After facing initial difficulty in running MADA (Pei et al., 2018) from the source code[2] (due to it being written in now outdated Caffe framework), we converted the codebase into `PyTorch` equivalent for our experiments. There were no publicly available implementations of DGDA (Akbari & Jafari, 2019) and

[1] https://github.com/jindongwang/activityrecognition/tree/master/code/percom18_stl.
[2] https://github.com/thuml/MADA.

Table 3
List of hyper-parameters.

Type	Parameter	Values considered	Optimal chosen value
Pre-processing	Window size	64, 128, 256, 512	128
Pre-processing	Sliding window offset	64, 32, 16	16
Discriminator	Fake/Target label	0. 0.1	0.1
Discriminator	Real/Source label	1, 0.9	0.9
Training	Batch size	32, 64	32
Classifier	Learning rate (Adam)	[0.0001–0.01]	[a]
Generator	Learning rate (SGD)	[0.0001–0.01]	[a]
Discriminator	Learning rate (Adam)	[0.0001–0.01]	[a]
Generator	γ	[0.85–1.0]	[a]

[a]Denotes the value was found with *random search* on the validation split per trial.

SA-GAN (Soleimani & Nazerfard, 2019), so we opted to re-implement each one of them following the directions provided in the respective articles. Our implementation of *STranGAN* will be made available GitHub.[3]

6.3.3. Pre-processing

We first re-sampled the data from all the sensors to 50 Hz and then divided the accelerometer data into individual windows of 128 samples, with a sliding window offset of 16; this results in 87.5% overlap between consecutive windows and a 2.56 s per window. The choice of the sampling window size and the offset was based on the observation that transformation generated by the $\mathcal{T}$ network is applied uniformly across *all* samples in a single window. With a larger window, the likelihood of a single window containing data from multiple ADLs increases, which, in turn, increases the likelihood of applying an incorrect transformation. Conversely, a smaller window is unable to capture the movement range characteristic of specific activities, making it harder to perform accurate classification. A very high overlap (2.24 s) ensures that even if successive windows receive different local affine transformation from the $\mathcal{T}$ network, their effect is smoothened out; as a result, for a longer activity sequence the effective transformation per window no longer remains uniform and helps to approximate more realistic transformations.

6.3.4. Evaluation

We followed the methodology described in Akbari and Jafari (2019) to obtain train/test/validation split: 60% of the source data is used to train source domain classifiers, and the remaining 20% source domain data along with 20% target domain data is used to train the transfer learning models (without labels). The same unlabeled data was used for testing the models. Finally, the remaining 20% data was used as a validation set to optimize the hyper-parameters.

6.3.5. Hyper-parameters

The batch sizes were set to 32 while training all of the three ($\mathcal{T}$, D, C) networks. We opted to not use Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) as we noticed its adverse effect on the stability of training the $\mathcal{T}$ network. Instead we employed *L2* regularization as the primary weight regularization method. To improve the GAN training convergence we incorporate *Label Smoothing* (Salimans et al., 2016) where we replaced the 0 (fake/target) and 1 (real/source) labels for the discriminator with 0.1 and 0.9 respectively. We also use *Spectral Normalization* (Miyato, Kataoka, Koyama, & Yoshida, 2018) on the convolution layers and *Mini-batch Discrimination* (Salimans et al., 2016) on the fully connected layers of the discriminator. We opted to use the SGD optimizer for the discriminator and the Adam optimizer for the generator as it helped improve the learning stability (Radford, Metz, & Chintala, 2015). Adam was also used as the optimizer for MADA. We would like to point out that their original implementations recommend SGD but we achieved sub-optimal results with SGD for MADA. We optimized our model hyper-parameters (especially the learning rates and the γ parameter) with *Randomized Search* on the validation sets as each dataset required a different optimal value search. The search space of the hyper-parameters are listed in Table 3.

6.4. Evaluation criteria

We use the **F1** score (harmonic mean between precision and recall) as our primary performance metric instead of the accuracy score. This is driven by our analysis of the datasets (illustrated in Fig. 6), which reveals the relatively large imbalance in class label volumes across different activity classes (especially in PAMAP2 and OPPORTUNITY datasets). According to Chen et al. (2019), accuracy scores tend to be, on average, 10% higher than the corresponding **F1** score in a number of benchmark datasets and model combinations, indicating that accuracy can be a misleadingly optimistic metric in datasets with non-uniform distribution of activity labels. To ensure that the reported F1 scores of *STranGAN* and the baselines are comparable and emphasize the statistical significance of the results, we run each experiment 10 times with different seed and report the average F1 score. We run a paired Wilcoxon signed-rank test between the 10 runs of each baseline, calculate the p value *STranGAN* and only report the result if $p < 0.01$ in Section 7, emphasizing that all the reported findings are statistically significant.

[3] https://github.com/azmfaridee/strangan-chase-2021.

Table 4
Comparison of cross body position transfer in **PAMAP2** dataset represented by average F1 scores in 10 runs. Source and target samples contain data from all 8 users.

Transfer	STL (Chen et al., 2019)	MADA (Pei et al., 2018)	DGDA (Akbari & Jafari, 2019) (Unpaired)	DGDA (Akbari & Jafari, 2019) (Paired)	SA-GAN (Soleimani & Nazerfard, 2019)	*STranGAN* (ours)
Wrist → Chest	44.34	57.04	19.19	58.3	57.33	**64.86**
Wrist → Ankle	48.17	56.75	20.86	57.66	56.97	**64.00**
Chest → Wrist	42.03	55.61	18.33	55.87	55.11	**60.76**
Chest → Ankle	40.26	54.7	18.43	55.99	55.08	**59.44**
Ankle → Wrist	41.28	52.49	18.91	55.61	54.52	**59.38**
Ankle → Chest	41.67	52.65	17.51	54.66	54.25	**59.27**
Average	42.96	54.87	18.87	56.35	55.54	**61.29**

7. Results

In this section, we compare the classification performance of *STranGAN* against the 4 baselines under the following two scenarios: (i) cross body position, and (ii) cross person. *STranGAN* and all of the baselines except DGDA (Akbari & Jafari, 2019) does not require any explicit temporal synchronization (or *pairing*) between the source and target domain samples; hence source and target domain samples are independently shuffled before training each model even if the original data collection methodology supported such synchronization.

7.1. Cross body position adaptation

To evaluate the models under cross body position, we treat the data from all the users from a single body position as the source and another body position as the target (for all the users). In below, we discuss the results for this setup for the two datasets: (i) PAMAP2 and (ii) OPPORTUNITY.

7.1.1. PAMAP2

In Table 4, we compare *STranGAN's* performance on the PAMAP2 dataset against the baselines. Given three distinct body positions (Wrist, Chest, Ankle), we have $\binom{3}{2} = 6$ distinct pair-wise transfer scenarios; each row represents the average F1 scores (across 10 independent runs) for a distinct pair. We observe that *STranGAN* consistently performs better than all the baselines; and all adversarial approaches (MADA, SA-GAN and *STranGAN*) performing on average better than the rest. SA-GAN, which represents another GAN based approach but one that requires a simultaneous update of the classifier during training, lags 5.75% behind *STranGAN*. It is also notable that in the absence of synchronized data between the domains, DGDA's performance lags the most (by average 42.42%).

7.1.2. OPPORTUNITY

In Table 5, we compare *STranGAN's* performance with the baselines for cross-body position adaptation based on the OPPORTUNITY dataset. For each of the 5 body positions (BACK, LLA, LUA, RLA, RUA) we have $\binom{5}{2} = 20$ transfer scenarios represented by each row; each entry represents average F1 scores of 10 repeated runs. We observe that *STranGAN* performs better than the other baselines in 16/20 cases and a close second in the remaining cases. On average, *STranGAN* achieves an F1 score of 67.75%, representing an improvement of 5.68% compared to SA-GAN (2nd best) and 6.16% compared to MADA (3rd best). DGDA's performance falls sharply (by average 35.65%) in the absence of synchronized data between the domains.

7.1.3. Additional observations

On closer inspection of such position-specific transfers in both datasets, we observe that transfer from arms/wrists to other body parts has higher average F1 score than transferring from ankle/back/chest positions (to other parts) for both simple (OPPORTUNITY) and diverse (PAMAP2) set of activity labels.

7.2. Cross person adaptation

To evaluate how *STranGAN* can handle variations in sensor data arising due to the behavioral variations across individuals, we divide all the subjects from each dataset into two groups: (A and B) and transfer from group A to B while keeping the body position or device fixed. We discuss next the results for this setup for the three datasets: (i) PAMAP2, (ii) OPPORTUNITY and (iii) HHAR. As none of the datasets contain any synchronization information across the user samples (i.e., the samples from different individuals are not collected simultaneously); all of the transfer techniques (including DGDA) are evaluated in an unsynchronized setting.

Table 5
Comparison of cross body position transfer in **OPPORTUNITY** dataset represented by mean F1 scores in 10 runs. Source and target samples contain data from all 4 users.

Transfer	STL (Chen et al., 2019)	MADA (Pei et al., 2018)	DGDA (Akbari & Jafari, 2019) (Unpaired)	SA-GAN (Soleimani & Nazerfard, 2019)	*STranGAN* (ours)
Back → LLA	50.02	51.59	21.76	**63.59**	61.75
Back → LUA	47.53	61.71	31.00	61.27	**64.95**
Back → RLA	**60.84**	48.73	20.85	54.02	56.39
Back → RUA	50.23	61.91	27.01	59.57	**62.66**
LLA → Back	55.61	53.48	25.47	61.93	**62.82**
LLA → LUA	58.88	71.32	36.29	67.59	**76.11**
LLA → RLA	62.03	67.66	25.44	60.49	**69.46**
LLA → RUA	60.46	**67.67**	29.45	62.74	67.39
LUA → Back	47.31	63.42	22.03	56.57	**69.34**
LUA → LLA	60.09	72.37	33.42	65.62	**73.82**
LUA → RLA	54.52	55.03	27.27	65.19	**68.62**
LUA → RUA	61.96	58.15	20.48	**67.63**	64.27
RLA → Back	60.63	49.35	34.74	50.84	**62.6**
RLA → LLA	58.08	67.84	34.66	63.58	**71.29**
RLA → LUA	52.57	66.53	21.23	65.45	**67.99**
RLA → RUA	67.26	69.7	31.14	64.15	**73.06**
RUA → Back	50.28	60.78	27L	60.91	**68.83**
RUA → LLA	63.46	59.49	29.66	64.38	**75.04**
RUA → LUA	55.02	57.6	26.62	60.47	**66.47**
RUA → RLA	62.97	67.47	33.98	65.39	**72.09**
Average	56.99	61.59	27.98	62.07	**67.75**

Table 6
Comparison of cross person transfer, $\{1,2,3,4\} \rightarrow \{5,6,7,8\}$ in PAMAP2 dataset represented by the mean F1 scores of 10 runs.

Position	STL (Chen et al., 2019)	MADA (Pei et al., 2018)	DGDA (Akbari & Jafari, 2019) (Unpaired)	SA-GAN (Soleimani & Nazerfard, 2019)	*STranGAN*
Ankle	65.14	63.26	27.87	66.06	**66.84**
Chest	65.46	73.34	28.82	66.28	**76.22**
Wrist	65.73	**69.52**	29.12	65.89	67.63
Average	65.44	68.71	28.60	66.08	**70.23**

Table 7
Comparison of cross person transfer, $\{1,2\} \rightarrow \{3,4\}$ in OPPORTUNITY dataset represented by the mean F1 scores of 10 runs.

Position	STL (Chen et al., 2019)	MADA (Pei et al., 2018)	DGDA (Akbari & Jafari, 2019) (Unpaired)	SA-GAN (Soleimani & Nazerfard, 2019)	*STranGAN*
BACK	70.66	77.91	32.66	74.56	**79.03**
LLA	70.03	79.10	33.74	**82.00**	81.16
LUA	61.06	84.98	36.72	76.76	**86.73**
RLA	77.20	74.63	35.43	73.43	**77.49**
RUA	68.44	82.71	37.11	79.27	**84.03**
Average	69.48	79.87	35.13	77.20	**81.69**

7.2.1. PAMAP2

To evaluate cross person heterogeneity in PAMAP2 dataset, we put subjects 1–4 in source group (A) and subjects 5–8 in target group (B). With this setup we perform 3 transfers for the Ankle, Chest and Wrist sensor separately and list the results (average F1 scores of 10 repeated runs) in Table 6. We observe that *STranGAN* performs on average 1.52% better for $\frac{2}{3}^{rd}$ of the cases than MADA with a mean F1 score of 70.23%. We also notice that without the use of paired (synchronized) data samples, DGDA performs significantly poorly compared to the other baselines.

7.2.2. OPPORTUNITY

To evaluate cross person heterogeneity in OPPORTUNITY dataset, we put subject 1–2 in source group (A) and 3–4 in target group (B). With this setup we perform 5 transfers for the BACK, LLA, LUA, RLA, RUA sensor separately and list the result (average F1 scores of 10 repeated runs) in Table 7 for the 4 baselines and *STranGAN's* . We observe that *STranGAN* performs on average 1.82% better than MADA with a mean F1 score of 81.69%.

Table 8
Comparison of cross person transfer, $\{1,2,3,4,5\} \rightarrow \{6,7,8,9\}$ in HHAR dataset represented by the mean F1 scores of 10 runs.

Device	STL (Chen et al., 2019)	MADA (Pei et al., 2018)	DGDA (Akbari & Jafari, 2019) (Unpaired)	SA-GAN (Soleimani & Nazerfard, 2019)	*STranGAN*
Nexus 4 (200 Hz)	60.08	84.92	27.25	78.20	**85.68**
Samsung S3 (150 Hz)	62.89	**87.13**	25.40	72.73	86.35
S3 Mini (100 Hz)	61.43	78.95	29.72	77.84	**84.88**
Galaxy S+ (50 Hz)	61.26	**88.20**	27.46	78.28	85.72
Average	61.41	84.80	27.46	76.76	**85.66**

7.2.3. HHAR

To evaluate cross person heterogeneity in HHAR dataset, we put users 1–5 in source group (A) and 6–9 in target group (B). With this setup we perform 4 transfers for the Nexus 4, Samsung S3, S3 Mini and S+ sensor separately and list the result (average F1 scores of 10 repeated runs) in Table 8. We observe that *STranGAN* performs only 2% better than MADA with a mean F1 score of 86.80%. However, *STranGAN* produces more consistent result across the devices compared to MADA; the standard deviation of the F1 scores of *STranGAN* is 0.78 which is much lower than the standard deviation of scores of MADA (4.13).

7.2.4. Additional observations

in PAMAP2 dataset, the chest sensor provides the highest F1 score in cross person transfer; but in case the of OPPORTUNITY dataset, similarly positioned BACK sensor does not seem to provide the best one. Since PAMAP2 dataset has a larger (11) set of activities (compared to only 4 simple ones in OPPORTUNITY dataset), *STranGAN* has an easier time adjusting the more 'muted' sensor readings from chest sensors than highly diverse limb movements. In HHAR dataset, all the sensors were placed in the same waist pouch, hence there is little performance difference across persons.

7.3. Interpreting the domain discrepancies with A_θ

Using learnable affine transformations to align the source and target samples provides an additional benefit — the opportunity to add interpretability into the domain adaptation pipeline (not possible with any of the baselines from Section 6.2). More specifically, each parameter of A_θ can be associated with a particular form of spatial transformation. Fig. 7 gives a simple demonstration of this idea where we plot the probability densities of 12 parameters of A_θ during LUA → BACK transfer for 4 users in OPPORTUNITY dataset (with the green vertical line marking the mean). From Eqs. (2) and (3), we know that the diagonal parameters of A_θ $(\theta_{11}, \theta_{22}, \theta_{33})$ relate to *scaling* and the parameters from the right most column $(\theta_{14}, \theta_{24}, \theta_{34})$ are relevant to the *translate* operation with respect to (x, y, z) axes. As we can notice from Fig. 7, in order to align data samples from LUA position to match the distribution of samples collected at BACK position, the $\mathcal{T}$ network has to apply an average *scaling* of $(0.8171, 1.1250, 1.1282)$ and *translation/shift* of $(-0.1287, 0.2265, 0.1067)$ around (x, y, z) axes. The effect of these transformations are noticeable when we superimpose the raw distributions of the source and target samples (shown in Fig. 8) before and after applying the transformations. Particularly, in Fig. 8b the distributions of the samples across domain have higher overlap due to the corresponding *scaling* and *translation* being applied.

The composite yet compact nature of A_θ also comes with a trade-off, as it can be difficult to discern what might have been the actual individual series of transformations (rotation, sheer, scaling) that contributes to A_θ. A possible alternative is to use a multi-stage generator model, where each stage is explicitly structured to learn the parameters of each individual operation, which can provide a more interpretable model. AutoAugment (Cubuk, Zoph, Mane, Vasudevan, & Le, 2018) is an example of such approach in computer vision literature, where the authors have utilized reinforcement learning to capture a series of augmentations that maximizes classification performance on a validation set. However, due to the discrete nature of the search space requiring RL policy optimization, AutoAugment and its derivatives (Ho, Liang, Chen, Stoica, & Abbeel, 2019) takes considerable computing resources which might be impractical for wearable sensing use cases. Nevertheless, we believe that being able to further break down A_θ to its individual components for more interpretability is a fertile ground for future investigations.

7.4. Reconstruction loss and γ

The reconstruction loss $\|B_S - \mathcal{T}(B_S)\|_2$ and its associated weight parameter γ plays a central role in making *STranGAN* learn good transformation from unlabeled target domain data. The reconstruction module forces the generator to put additional consideration when encountering new data; ideally, if the new data has some overlap with source samples in its distribution, the generator should *learn* to leave it alone and focus on only what looks different. The γ parameter gives a simple way to tune this learning ability of the generator. Here we give a simple example to demonstrate its role. We vary the γ parameter between [0 and 1.0] (0 effectively disabling reconstruction altogether) while doing activity transfer, from Wrist to Chest, across 8 persons in the PAMAP2 dataset. The associated F1 score for each γ value is shown in Table 9. Note that as soon as γ drops below 0.85, the F1 score starts falling sharply. Interestingly, setting γ to too high a value (in this case > 0.95) can also overwhelm the generator. We suspect that for each pair of domains, there exists an optimum γ value that is tied to their 'relatedness'. We observe that it resided within $[0.85 - 1.0]$ during our empirical analysis; accordingly, our hyper-parameter search for γ was confined to this range.

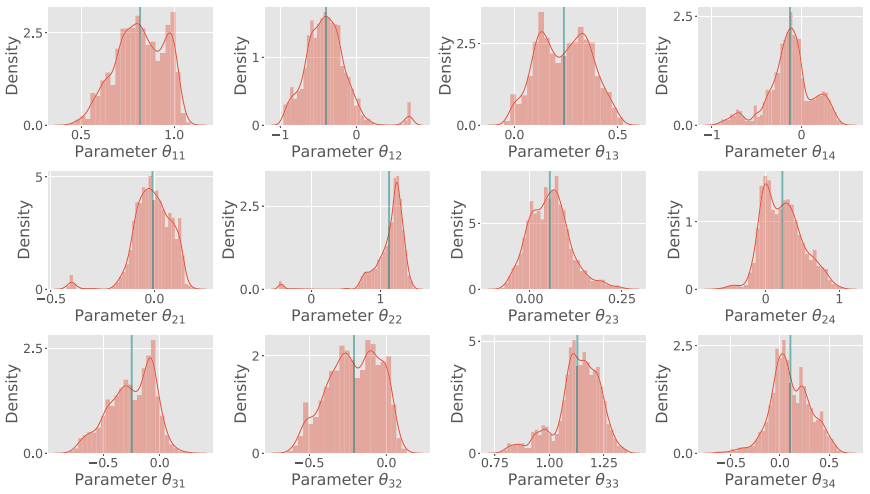


Fig. 7. Probability density plot of the 12 transformation parameters of A_θ (Eqs. (2) and (3)) during LUA → BACK transfer for 4 users in OPPORTUNITY dataset.

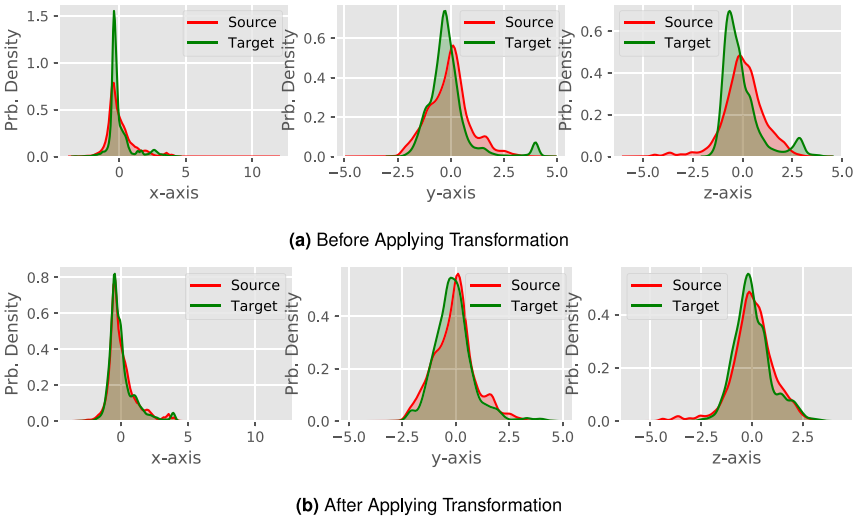


Fig. 8. Probability density plot of raw source and target data (3-axis accelerometer) in LUA → BACK transfer for 4 users in OPPORTUNITY dataset; Before and after applying transformation.

Table 9
Effect of γ parameter on final F1 score.

Gamma	0.00	0.20	0.30	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
F1 score	12.43	12.95	11.06	12.95	12.97	14.47	13.98	21.87	31.85	44.09	58.65	64.8647	62.30

7.5. Key takeaways

Our results show that *STranGAN* is capable of handling body-position heterogeneity, achieving around 5.68-5.75% improvement in human activity recognition compared to the state-of-the-art (SOTA) shallow and deep learning-based methods, without needing to modify the pre-trained HAR classifier. *STranGAN* also performs well under cross person heterogeneity, albeit achieving a relatively smaller 1.5-2% improvement over state-of-the-art techniques. It is worth reiterating that our primary objective was not to improve the accuracy of domain adaptation, but to offer a competitive approach that could support such adaptation without requiring either classifier modification or access to either labeled source or target domain data, and thus support a wider variety of practical use cases.

8. Discussion

While our work provides compelling evidence of *STranGAN's* ability to support label-free domain adaptation by learning a spatial transformation on inertial sensing streams that maintains compatibility with a pre-existing classifier, there are several open issues that warrant further investigation.

8.1. Supporting privacy objectives

While not a primary objective of our design, *STranGAN* arguably promotes privacy-preserving domain adaptation as it does not require any sharing of explicitly-labeled source or target domain data with the $\mathcal{T}$ and D network. Of course, alternative approaches, such as federated learning, also provide privacy-compliant mechanisms to evolve classifiers to handle individual/device specific activity variations. In federated learning, each individual device needs to perform local adaptation of its own model and then share *model parameters* with a central server. While there is *no exchange of data, labeled or unlabeled* across domains, such local adaptation does, however, require explicitly labeled data samples. Moreover, in this approach, (a) the update of a local HAR model does impose computational overhead, and (b) the updated model is susceptible to model *poisoning* attacks (Bhagoji, Chakraborty, Mittal, & Calo, 2019). While we do not view *STranGAN* as a fully-fledged mechanism for privacy-preserving domain adaptation, it can provide a more self-contained counterpart to alternative approaches.

8.2. Extension to more complex activities & gestures

STranGAN's current design is based on a single spatial transformer, that is identically applied to all data points in a single window. This design choice was made deliberately, as an alternative per-sample transformation matrix would result in an extremely large parameter space for $\mathcal{T}$ network, posing significant challenges in learning such a model from relatively small volumes of target domain data. To ensure that our spatial transformers do not straddle multiple activities, we thus use a relatively small window size (e.g., 128 samples, corresponding to 2.56 s of activity data @50 Hz) in this work. However, such a small window may be inappropriate for complex ADLs (e.g., cooking, cleaning house) which typically manifest themselves over much a longer time duration (e.g., 10 s-1 min). Conversely, even a 2.56 s window size may be too large for tracking transient, short-lived gestural activities (e.g., table tennis strokes or dance steps). Accordingly, we believe that, to achieve satisfactory performance, the current architecture of *STranGAN* requires a careful activity-dependent selection of the input sample window size. To extend *STranGAN's* utility across a range of activity durations, we shall need to develop a more flexible model that permits sample-specific transformation.

8.3. Extension to diverse heterogeneities

In this paper, we have primarily focused on mitigating personal and positional heterogeneity with *STranGAN* framework. Sensor heterogeneity is also a common issue HAR which we did not explicitly investigate in this work. Sensor heterogeneity primarily results from using devices of different types (smart-phone vs smart-watches) and the underlying sensitivity range, sensor bias, and gravitation component correction, and other signal processing algorithms employed in IMU hardware. We hypothesize that the affine transformation-based raw feature space alignment of *STranGAN* at its current form might not be fully suitable for mitigating these sensor heterogeneities. For example, take the example of trying to align the samples of two sensors — one with high sensitivity ($16g$) vs. another with lower one ($6g$). The lower sensitivity sensor streams will have a lot of the high-intensity signals clamped and it might be very difficult to generate equivalent signals of high sensitivity from that stream using just an affine transformation. Other avenues for further investigation are (a) the existence of multiple heterogeneities at the same time (for example transferring from a person's hand sensor to another person's ankle sensor), (b) multi-source adaptation — transferring from two or more body positions to a new position. We speculate that these can be archived by carefully modifying the discriminator to handle multiple sources and/or heterogeneity labels. Also, it would be very interesting to analyze if certain labels have a higher success rate of being transferred vs others — implying that certain activities are easier to approximate than others with affine transformations. In our future investigations, we wish to seek the answer to these questions with a cross dataset transfer experiment to represent a more challenging and realistic evaluation scenario.

8.4. Effect of label distribution mismatch between source and target domain

Without the knowledge of any activity labels from either the source and the target domain, the adversarial learning mechanism of *STranGAN* solely relies on implicit similar label distribution across the domains to learn the optimal transformation parameters. Hence, we suspect *STranGAN* might suffer from degraded performance if the source and target label distribution are widely different or in the presence of previously unseen labels (from the perspective of the classifier). This limitation stems from the design decision to decouple the classifier from *STranGAN's* training stage that helps to train it without any labeled data in either of the domain. But during this process, the classifier remains oblivious to any potential label distribution discrepancy across the domains. However, being able to handle label distribution discrepancy while keeping the source domain trained classifier intact is a nontrivial task. In the future, we would like to investigate whether integrating pseudo labeled feedback from the classifier or any other form of weak supervision (regarding the label distributions across the domains) into the adversarial learning pipeline provides any tangible benefits in handling such discrepancies.

9. Conclusion

In this work, we presented *STranGAN*, a decomposable transfer learning architecture for HAR applications that can be trained without requiring any labeled data, in both the source and target domain by re-purposing any pre-trained classifier. *STranGAN's* decomposability arises from its uses of an explicit Spatial Transformer that serves as a preprocessor on the target domain sensor data, and that effectively decouples the domain adaptation function from the subsequent classification step. Consequently, *STranGAN* can support robust domain adaptation without requiring any modifications or updates in the classifier (which we assume to be trained, a-priori, on the labeled source domain data). Key to the design of an effective Spatial Transformation is the use of an adversarial model, together with a novel reconstruction loss, that ensures that *STranGAN* modifies only those segments of target domain sensor data that are distributionally dissimilar to the corresponding source domain data samples.

Via experimental studies with 3 different and representative HAR datasets (that include a range of real-world activities), we showed that *STranGAN* provides a domain adaptation mechanism that is competitive against state-of-the-art baselines, all of which require explicit retraining of the classifier or the collection of synchronized source and target domain data samples. In particular, *STranGAN's* ability to support spatial operations such as scaling and rotation makes it effective in handling transfer across diverse individuals and different on-body positions. Moreover, the use of a spatial transformer directly on the raw sensor data samples helps support the goal of interpretable transformations, which is often elusive for DNN-based classification techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to sincerely thank all the anonymous reviewers for their thoughtful feedback and suggestions to improve the quality of this paper. This research is supported by NSF, United States CAREER grant 1750936, U.S. Army grant W911NF2120076, ONR, United States grant N00014-18-1-2462, and Alzheimer's Association, United States grant AARG-17-533039.

References

Abdallah, Z. S., Gaber, M. M., Srinivasan, B., & Krishnaswamy, S. (2015). Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, *150*, 304–317.

Akbari, A., & Jafari, R. (2019). *Transferring activity recognition models for new wearable sensors with deep generative domain adaptation* (pp. 85–96). ACM.

Berger, M. (2009). *Geometry I*. Springer Science & Business Media.

Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *Icml* (pp. 634–643).

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Emnlp* (pp. 120–128).

Bloomenthal, J., & Rokne, J. (1994). Homogeneous coordinates. *The Visual Computer*, *11*(1), 15–26.

Bobovych, S., Sayeed, F., Banerjee, N., Robucci, R., & Allen, R. P. (2020). Resteaze: low-power accurate sleep monitoring using a wearable multi-sensor ankle band. *Smart Health*, *16*, Article 100113.

Boukhechba, M., Daros, A. R., Fua, K., Chow, P. I., Teachman, B. A., & Barnes, L. E. (2018). Demonicsalmon: Monitoring mental health and social interactions of college students using smartphones. *Smart Health*, *9*, 192–203.

Chen, Y., Wang, J., Huang, M., & Yu, H. (2019). Cross-position activity recognition with stratified transfer learning. *PMC*, *57*, 1–13.

Cook, D., Feuz, K. D., & Krishnan, N. C. (2013). Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, *36*, 537–556.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501.

Duan, L., Tsang, I. W., & Xu, D. (2012). Domain transfer multiple kernel learning. *IEEE TPAMI*, *34*, 465–479.

Faridee, A. Z. M., Khan, M. A. A. H., Pathak, N., & Roy, N. (2019). AugToAct: SCaling complex human activity recognition with few labels. *EAI MobiQuitous*.

Finnveden, L., Jansson, Y., & Lindeberg, T. (2020). Understanding when spatial transformer networks do not support invariance, and what to do about it. CoRR abs/2004.11678.

Ganin, Y., & Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *JMLR*, *17*, 59:1–59:35.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Adv neural inf process syst, Vol. 3* (pp. 2672–2680).

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. arXiv preprint arXiv:1302.4389.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *JMLR*, *13*, 723–773.

Hammerla, N. Y., Halloran, S., & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880.

Ho, D., Liang, E., Chen, X., Stoica, I., & Abbeel, P. (2019). Population based augmentation: Efficient learning of augmentation policy schedules. In *International conference on machine learning* (pp. 2731–2741). PMLR.

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., et al. (2018). Cycada: Cycle-consistent adversarial domain adaptation. In *Pmlr*: *Vol. 80*, *Icml* (pp. 1994–2003). PMLR.

House, D., & Keyser, J. C. (2016). *Foundations of physically based modeling and animation*. CRC Press.

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. In *Adv neural inf process syst, Vol. 2015-Jan* (pp. 2017–2025).

Jeyakumar, J. V., Lai, L., Suda, N., & Srivastava, M. (2019). Sensehar: a robust virtual activity sensor for smartphones and wearables. In *Sensys'19* (pp. 15–28).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv:1404.2188.

Khan, M. A. H. A. H., Roy, N., & Misra, A. (2018). Scaling human activity recognition via deep learning-based domain adaptation. *PerCom*.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Adv neural inf process syst* (pp. 971–980).

Lee, G., & Lee, S. (2020). Mode penalty generative adversarial network with adapted auto-encoder. ArXiv abs/2011.07706.

Li, F., Shirahama, K., Nisar, M. A., Köping, L., & Grzegorzek, M. (2018). Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, *18*(2), 679.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.

Long, M., Wang, J., Ding, G., Sun, J., & Yu, P. S. (2013). Transfer feature learning with joint distribution adaptation. In *Iccv* (pp. 2200–2207).

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *Icml, Vol. 5* (pp. 3470–3479).

Mathur, A., Isopoussu, A., Kawsar, F., Berthouze, N., & Lane, N. D. (2019). Mic2mic: using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems. In *Ipsn* (pp. 169–180). ACM.

Mathur, A., Zhang, T., Bhattacharya, S., Veličković, P., Joffe, L., Lane, N. D., et al. (2018). Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *Ipsn 2018* (pp. 200–211). IEEE Press.

Milanko, S., & Jain, S. (2020). Liftright: quantifying strength training performance using a wearable sensor. *Smart Health*, *16*, Article 100115.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, *105*, 233–261.

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, *22*, 199–210.

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-adversarial domain adaptation. In *Thirty-second aaai conference on artificial intelligence*.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Cvpr* (pp. 652–660).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017b). PointNet: DEep learning on point sets for 3D classification and segmentation. In *Cvpr 2017, Vol. 2017-Janua* (pp. 77–85).

Qin, X., Chen, Y., Wang, J., & Yu, C. (2019). Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *ACM IMWUT*, *3*(4), 1–25.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Rastegari, E., & Ali, H. (2020). A bag-of-words feature engineering approach for assessing health conditions using accelerometer data. *Smart Health*, *16*, Article 100116.

Reiss, A., & Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *Iswc* (pp. 108–109). IEEE.

Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., et al. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In *Inss* (pp. 233–240). IEEE.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Adv Neural Inf Process Syst* (pp. 2234–2242).

Soleimani, E., & Nazerfard, E. (2019). Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. arXiv preprint arXiv:1903.12489.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, *15*, 1929–1958.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., & Sutton, C. (2017). Veegan: Reducing mode collapse in GANs using implicit variational learning. In *Nips*.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., et al. (2015). Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Sensys '15* (pp. 127–140).

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. CoRR abs/1412.3.

Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., et al. (2017). Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Icmi, Vol. 2017-Janua* (pp. 216–220). ACM.

Wang, J., Chen, Y., Hao, S., Feng, W., & Shen, Z. (2017). Balanced distribution adaptation for transfer learning. *ICDM*, *2017-Novem*, 1129–1134.

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3–11.

Ware, S., Yue, C., Morillo, R., Lu, J., Shang, C., Bi, J., et al. (2020). Predicting depressive symptoms using smartphone data. *Smart Health*, *15*, Article 100093.