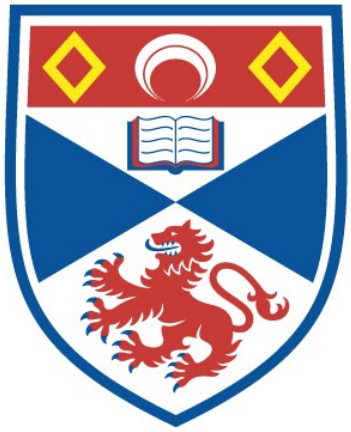
Unsupervised domain adaptation in sensor-based human activity recognition

Andrea Rosales Sanabria

A thesis submitted for the degree of PhD
At the
University of St Andrews



2022

Candidate's declaration

I, Andrea Rosales Sanabria, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 25,701 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree. I confirm that any appendices included in my thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

I was admitted as a research student at the University of St Andrews in October 2017.

I confirm that no funding was received for this work.

Date 20/02/2022

Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree. I confirm that any appendices included in the thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

Date 20 Feb. 2022

Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Andrea Rosales Sanabria, confirm that my thesis does not contain any third-party material that requires copyright clearance.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Printed copy

No embargo on print copy.

Electronic copy

No embargo on electronic copy.

Date 22/02/2022 Signature of candidate

Date 20 Feb. 2022 Signature of supervisor

Underpinning Research Data or Digital Outputs

Candidate's declaration

I, Andrea Rosales Sanabria, hereby certify that no requirements to deposit original research data or digital outputs apply to this thesis and that, where appropriate, secondary data used have been referenced in the full text of my thesis.

Date 20/02/2022 Signature of candidate

Abstract

Sensor-based human activity recognition (HAR) is to recognise human daily activities through a collection of ambient and wearable sensors. Sensor-based human activity recognition is having a significant impact in a wide range of applications in smart city, smart home, and personal healthcare. Such wide deployment of HAR systems often faces the annotation-scarcity challenge; that is, most of the HAR techniques, especially the deep learning techniques, require a large number of training data while annotating sensor data is very time- and effort-consuming. Unsupervised domain adaptation has been successfully applied to tackle this challenge, where the activity knowledge from a well-annotated domain can be transferred to a new, unlabelled domain. However, existing techniques do not perform well on highly heterogeneous domains.

To address this problem, this thesis proposes unsupervised domain adaptation models for human activity recognition. The first model presented is a new knowledge- and data-driven technique to achieve coarse- and fine-grained feature alignment using variational autoencoders. This proposed approach demonstrates high recognition accuracy and robustness against sensor noise, compared to the state-of-the-art domain adaptation techniques. However, the limitations with this approach are that knowledge-driven annotation can be inaccurate and also the model incurs extra knowledge engineering effort to map the source and target domain. This limits the application of the model.

To tackle the above limitation, we then present another two data-driven unsupervised domain adaptation techniques. The first method is based on bidirectional generative adversarial networks (Bi-GAN) to perform domain adaptation. In order to improve the matching between the source and target domain, we employ Kernel Mean Matching (KMM) to enable covariate shift correction between transformed source data and original target data so that they can be better aligned. This technique works well but it does not separate classes that have similar patterns. To tackle

this problem, our second method includes contrastive learning during the adaptation process to minimise the intra-class discrepancy and maximise the inter-class margin. Both methods are validated with high accuracy results on various experiments using three HAR datasets and multiple transfer learning tasks in comparison with 12 state-of-the-art techniques.

Acknowledgments

> If people never did silly things, nothing intelligent would ever get done.
>
> *Ludwig Wittgenstein*

This thesis concludes my PhD journey.

Firstly, I would like to thank and to express my deep gratitude to my supervisor and mentor Dr. Juan Ye for supporting my research directions, which allowed me to explore new ideas in the field of human activity recognition and deep learning, who challenged me and always trusted in me, even when my ideas sound crazy, and without whom there would be no single page of this thesis. I feel very fortunate of having her as my supervisor. She is a deep thinker, very smart and hardworking researcher.

I give special thanks to Dr. Simon Dobson, Head of School of Computing Science, for his support during my PhD. I had the chance to collaborate with him in two research papers. I thank Dr. Tom Kelsey, my second supervisor, for his help with my research.

I would like to acknowledge Santander 600 Scholarship for providing me the scholarship from School of Computer Science. I am honoured to be one of the recipients of this Scholarship which provided me with the opportunity to continue my academic success while contributing to their outstanding program. Thanks for your support and for investing in my future.

I would like to express my deepest gratitude to my family; the source of my life energy. Mom, Dad and Diana thank you for your endless support and your unconditional confidence in me. It is difficult to be away from home and it is even harder without seeing you for such a long time. You are my source of inspiration and my reason to keep going and working hard every day.

Behind this thesis, my PhD life has been very enjoyable thanks to all the nice people I met which opened another door for me to learn about different cultures. Fahrurrozi, Juanjo, and Lei who became very good friend and supported me in my ups and downs during these four years. Thanks Esma for the random chats and support. Looking forward to our new adventure. It is so great to have so many talented colleagues and friends.

List of Publications

Journals

- **Andrea Rosales Sanabria**, Franco Zambonelli, Simon Dobson and Juan Ye. *ContrasGAN: Unsupervised Domain Adaptation in Human Activity Recognition via Adversarial and Contrastive Learning*. Submitted to Pervasive and Mobile Computing (PMC). [Accepted for publication].

- **Andrea Rosales Sanabria**, F. Zambonelli and J. Ye. *Unsupervised Domain Adaptation in Activity Recognition: A GAN-Based Approach*. IEEE Access, volume 9, pages 19421-19438, year 2021.

- **Andrea Rosales Sanabria** and Juan Ye. *Unsupervised domain adaptation for activity recognition across heterogeneous datasets*. Pervasive and Mobile Computing, volume 64, pages 101147, year 2020.

- **Rosales Sanabria, Andrea**, Kelsey Thomas W., Dobson Simon, Ye Juan. *Representation learning for minority and subtle activities in a smart home environment*. Journal of Ambient Intelligence and Smart Environments, vol. 11, no. 6, pp. 495-513, 2019.

Conference

- **Rosales Sanabria, Andrea**, Kelsey Thomas W., Ye Juan. *Representation learning for minority and subtle activities in a smart home environment*. IEEE International Conference on Pervasive Computing and Communications, 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the advancement in medical science and technology, life expectancy is increasing. The European Commission has predicted that by 2025, the UK alone will see a rise of 44% in people over 60 [116]. However, with the ageing population comes the problem of medical costs and caring for older people. This motivates the development of new solutions to improve and guarantee an adequate quality of life and independence of elderly people.

Sensor-based human activity recognition aims to develop methods to understand human behaviour from a series of observations derived from motion, location, physiological signals and environmental information. Recent advances in data mining, machine learning, and deep learning [129] have demonstrated promising results in learning complex correlations between human activities and sensor features. With the support of these intelligent algorithms, we can infer current activities (e.g., making a meal or performing personal hygiene) and further detect changes over time.

Human activity recognition has been an active field for more than a decade. The first works on human activity recognition (HAR) date back to the late $'90s$ [91]. Since then, it has drawn much attention to researches due to its essential applications to real-world problems, especially for medical, military, and security applications.

1.1 Challenges

Today we have witnessed an increasing number of smart environment applications in our everyday life, such as health assessment (*e.g.*, stress and depression detection, and clinical assessment on cognition and mobility) [134], activity-driven behaviour changing applications (*e.g.*, smoking cessation) [139], home automation (*e.g.*, automatic heating configurations) [35], and so on. Activity recognition lies at the heart of these applications, which is the ability to recognise and predict users' current and future activities from data collected on a wide range of sensors that are embedded in an environment such as RFID, infra-red positioning sensors, and that are worn on the users such as smartwatches, glasses, and phones. Applications are designed to deliver intended services automatically and unobtrusively Based on inferred user activities.

Significant progress has been made in activity recognition over the past few years with the support of a large number of modern data-driven techniques, including Hidden Markov Models, Conditional Random Fields, Support Vector Machine [185], and the recent deep learning techniques [129]. To build a robust activity recognition model, most of these existing techniques require a large number of training data, annotated sensor data with activity labels. However, the key challenge faced in the current activity recognition community is the lack of sufficient training data. It often requires a lot of time and effort to annotate sensor data, either relying on users' constant self-report on what they are doing or recording users' activities via videos, which are annotated later by the users. In addition, there are never enough training data, as users might behave differently in the actual system running period from the training phase; that is, they can perform new types of activities or the same activity in different manners. Therefore, the current annotation approaches that require highly intensive effort or commitment are only suitable for lab- or test bed-based experiments on a small number of users over a short period of time. It is difficult, if not impossible, to apply them on hundreds or thousands of users over 1, 2, or 5 years. Furthermore, some HAR systems require the user to wear sensors and annotate the activities performed. These can be tedious, and data quality can be compromised. Reducing manual annotation and minimizing the number of sensors is beneficial and reduces the complexity of HAR systems. Therefore, the main challenge would be to train a classifier in a dataset with labelled instances and transfer that knowledge to recognise activities from an unlabelled dataset.

Unsupervised domain adaptation (UDA) is emerging as a practical approach to tackling

this annotation scarcity challenge. 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 [31]. However, HAR brings extra challenges in UDA. Most 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).

1.2 Aims and Objectives

This thesis aims to develop novel unsupervised domain adaptation techniques to achieve high activity recognition accuracy without collecting labels on target domain, in the face of the source and target domains in heterogeneous feature spaces. The hypothesis is these techniques will improve adaptation accuracy with heterogeneous feature transferring via domain knowledge or bi-directional GAN and with class-distinguishable feature transferring via contrastive learning.

- **Q1** Is it possible to relieve the annotation burden on individual users but still be able to build a robust activity recognition model by sharing and transferring activity models across users, even though the sensor deployments and operating environments are different?
- **Q2** The amount of training data can affect the model's performance. Can the domain adaptation model achieve high accuracy with little training data?
- **Q3** The performance of the sensors can vary over time, affecting the sensor features drastically. Is it possible to develop a system that performs robustly in the face of sensor noise?
- **Q4** Is it possible to discriminate better samples from different class labels leading to more class-discriminative adaptation?

The main goal of this thesis is to demonstrate that the proposed methods can perform accurate activity recognition across heterogeneous domains. To do so, we conduct a series of experiments using real-world datasets commonly used in HAR and we define various transfer learning tasks to examine our research questions.

To answer Q1, we implemented a knowledge-driven approach and a deep learning-based model and we compare the performance of the proposed methods with baseline domain adaptation techniques.

To answer Q2, we vary the percentage of training data in the target domain from 20% to 80% and assess the impact of the training data on the accuracy of domain adaptation.

To answer Q3, we systematically inject noise to sensor features and compare the recognition accuracy with baseline domain adaptation techniques.

To answer Q4, we add contrastive learning during the domain adaptation process 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. In contrast, the inter-class domain discrepancy is maximised to push the representations further away from each other to enable a suitable decision boundary.

We also aim to comprehensively review human activity recognition and domain adaptation techniques. Finally, we provide an overview of the main challenges faced in domain adaptation for HAR and future research directions.

1.3 Main Contributions

The main contributions of this thesis are:

- **An in-depth understanding of transfer learning and domain adaptation.** This thesis provides a comprehensive review, comparison and evaluation of non-deep learning and deep learning-based domain adaptation techniques. In addition to this, we present challenges and essential applications of HAR to provide further insights and future research directions.

- **A knowledge-driven model for unsupervised domain adaptation.** This method combines knowledge- and data-driven techniques in performing domain adaptation at different stages. We build on a general ontology for smart home datasets and achieve coarse-grained feature space remapping to link heterogeneous datasets without the need for labelled data in the target domain. We apply Variational Autoencoder (VAE) to perform fine-grained feature space alignment. This contribution has been published [140].

- **Two deep-learning based models for unsupervised domain adaptation.** We propose an unsupervised domain adaptation to tackle the scarcity of labelled datasets while not relying on predefined knowledge. Domain adaptation techniques have been increasingly applied to HAR applications. These techniques often work well when the source and target domains share feature space and they only need to tackle the difference in distributions. However, this assumption does not hold for many other HAR applications where two domains have heterogeneous feature spaces. In this direction, we propose two innovative domain adaptation methods using Bi-directional Generative Adversarial Network (Bi-GAN). The first model uses Kernel Mean Matching (KMM) to improve the matching between source and target domain. This contribution has been published [141]. The second model introduces contrastive learning to better discriminate samples from different class labels during the training process. This contribution has been accepted for publication.

- **Validation of the models by using third-party real-world datasets.** The validation process is aimed at assessing the effectiveness of the proposed methods. In this regard, experiments have been conducted using three publicly available datasets commonly used in HAR.

  We are also the first to go beyond domain adaptation and design and perform other HAR-specific experiments on sensor noise and sensitivity to training data. These experiments matter in HAR and other real-world applications where noisy data are pervasive and training data is scarce.

1.4 Thesis Outline

This thesis consists of 8 chapters, including the introduction chapter. In the following, we will describe each of them.

Chapter 2 presents essential background and basic human activity recognition and sensor technology concepts. It then reviews existing work in sensor-based human activity recognition and focuses on data-driven, knowledge-driven and deep learning techniques. Finally, it discusses applications and current challenges that form the motivations of this thesis.

Chapter 3 presents a comprehensive review of transfer learning and domain adaptation techniques for human activity recognition. First, the chapter introduces basic concepts of transfer learning and reviews existing work in supervised and unsupervised domain adaptation for accelerometer and binary data. Finally, it discusses the main challenges in domain adaptation for HAR.

Chapter 4 presents the first contribution of this thesis: Unsupervised Domain Adaptation for Human Activity Recognition (UDAR), a knowledge- and data-driven technique for feature alignment using variational autoencoders. This model addresses unsupervised domain adaptation between heterogeneous datasets. The chapter explains its configuration and the experimental results are presented in Chapter 7.

Chapter 5 presents two GAN-based models. The first model is called *shift*-GAN, which integrates bidirectional generative adversarial networks (Bi-GAN) and kernel mean matching (KMM) in an innovative way to learn intrinsic, robust feature transfer between two heterogeneous domains. Bi-GAN consists of two GANs bounded by a cyclic constraint, enabling more effective feature transfer than a classic, single GAN model. KMM is a powerful non-parametric technique to correct covariate shift, improving feature space alignment.

The chapter then discusses the ContrasGAN algorithm, our final contribution, which performs unsupervised alignment between source and target domain via Bi-GAN and introduces contrastive learning to minimise the intra-class discrepancy and maximise the inter-class margin. The evaluation results for both methods are presented in Chapter 7.

Chapter 6 presents the experimental setup and evaluation methodologies. In addition, the chapter provides the implementation and configuration details for UDAR, *shift*-GAN and ContrasGAN and provides a detailed explanation of the datasets used for evaluation and the baseline domain adaptation comparison techniques.

Chapter 7 discusses the evaluation results between UDAR, *shift*-GAN and ContrasGAN and the baseline domain adaptation techniques. The evaluation process is divided into two parts. The first one uses binary sensor datasets and the second one is for accelerometer datasets. For each type of dataset, we have defined specific transfer learning tasks. The results are then presented and analysed.

Chapter 8 summarises the thesis and draws overall conclusions. Finally, several future

research directions in human activity and domain adaptation are discussed.

Chapter 2

Background and Literature Review

In this chapter, we provide an overview of recent approaches to human activity recognition problem. Firstly, it discusses basic and high-level concepts related to Human Activity Recognition (HAR). Following this, it provides an extensive review of sensor technology. Towards the end of this chapter, we discuss more specific literature in human activity recognition, including research challenges and applications.

2.1 Introduction of Human Activity Recognition

Human activity recognition, often referred to simply by its acronym HAR, plays a vital role in people's daily life. Human activity recognition is the problem of identifying and classifying different human actions in real-life environments [83]. Human activities can vary from simple actions such as walking or jumping, to interactions between humans or objects such as drinking water or shaking hands. In this sense, the complexity and definition of activity can vary considerably. Activities can be simple using one part of the body or more complex using the whole body. According to [196], human actions can be categorised into three levels:

1. **Action primitives**, consist of actions where one part of the body is performing the action; for example clapping.

2. **Activities** are actions where the whole body is involved in performing the action, for example running.

3. **Interactions** are actions that involve objects or other persons, for example, shaking hands or throwing a ball.

Turaga et al. [156] differentiate between *action* and *activity*, where the former refers to a simple motion pattern usually executed by a single person and for a short period of time. For example, walking, swimming, running, etc. On the other hand, activity refers to a sequence of actions performed by several people interacting with each other. For example, a football team plays a game. Since the range of the complexity of activity is huge, Vrigkas et al. [164] classified human activities in six categories:(i) posture estimation are primitive movements that may correspond to a particular action of a person [182]; (ii) atomic actions are movements of a person describing a certain motion that may be part of more complex activities [113]; (iii) human-to-human or human-to-object are human activities that involve two or more persons or objects, (iv) group actions are activities performed by a group of persons [101]; (v) human behaviours refer to physical actions that are associated with the emotions, personality, and psychological state of the individual [101]; and (vi) events that are high-level activities describing social actions between individuals and indicate the intention of a person [89].

Hussain et al. [72] provided a more concrete classification of human activity recognition. They divided activities into three main categories:

1. **Action-based activities** are activities that involve some movements of the human body. This action can involve either the whole body or a specific part. They further classified these activities in six sub-categories: (i) gesture recognition such as waving a hand to control the TV; (ii) posture recognition such as standing up; (iii) behaviour recognition aims to infer the behaviour of a person; (iv) activities of daily living are daily activities in an indoor environment such as a home; (v) fall detection occurs when the position of the human body changes from the normal state (e.g., standing, sitting or walking) to reclining; and (vi) ambient assisted living intends to develop systems to assist humans in their daily lives.

2. **Motion-based activities** are related to the motion of a human being. They are not only related to performing a specific action but also related to the presence or absence of

motion. Examples of motion-based activities are tracking, motion detection and counting or estimating the number of people in a specific area.

3. **Interaction-based activities** that activities involve using or interacting with objects. A user can perform gestures or activities either by using their own body or using some object.

As seen from the different definitions stated above, human activities can vary in many ways. In this thesis, we propose the classification given in Table 2.1. We divide human activities into two main categories: (i) *action-based activities* are activities where one part of the body or the whole body is used to perform an action; and (ii) *interaction-based activities* are activities performed with a group of persons or activities that involve interacting with objects. We further divide action-based activities into gesture/ posture recognition, behaviour recognition and activities of daily living. Interaction-based activities are divided into two subcategories for human-to-human and human-to-object activities. We will focus mainly on posture recognition and activities of daily living. Table 2.1 describes each sub-category along with examples and potential applications.

Category	Subcategory	Description	Examples	Applications
Action-based activities	Gesture/ posture recognition	Simple actions that involve a specific body part or the whole body	Hand movement, standing, sitting, lying, walking.	Sign language interpretation, controlling an audio player, game consoles.
	Behaviour recognition	Aims to infer/ recognise the behaviour of a person.	Elderly care centres and smart homes. Analyse customer behaviour in a shopping centre.	Monitor patients remotely. Improve shopping experience.
	Activities of daily living	Identifying daily activities in an indoor environment such as a smart home	Eating, cooking, sleeping, sitting, bathing.	Ambient assisted living, healthcare remote tracking
Interaction-based activities	Human-to-human	Activities that involve two or more persons	An accelerometer sensor attached to a cup.	To monitor the performance of a team during a training session.
	Human-to-object	Activities that involve the interaction of two or more persons and objects	People living in a smart home and their interaction with different objects.	Smart refrigerator to monitor food usage.

Table 2.1: Classification of human activity recognition.

2.2 Sensor Technologies

Human activity recognition is a composite process. Chen et al. [24] proposed to decompose it into four primary tasks:

- To choose and deploy appropriate sensors to objects and environments to monitor and capture a user's behaviour along with the state change of the environment.
- To collect, store, and process received information through data analysis techniques and/or knowledge representation formalisms at appropriate levels of abstraction.
- To create computational activity models in a way that allow software systems to conduct reasoning and manipulation.
- To select or develop intelligent algorithms to infer activities from sensor data.

According to this classification, the first task is deploying the sensors to capture the users' behaviour. Some of the most common sensors used for activity recognition are [72]:

- **Accelerometer**. An accelerometer is an electromechanical device that measures acceleration in multiple directions (x, y, and z-direction) simultaneously.
- **Motion sensors**. Motion sensors are used to detect the motion or the presence of a subject in a particular environment.
- **Proximity sensors**. Proximity sensors can detect the presence of nearby objects without making any physical contact.
- **Gyroscope**. Sensors can measure and maintain the orientation and angular velocity of an object.
- **Radio-based**. Sensors use electromagnetic fields to identify and track objects automatically. The most common ones are RFID sensors.
- **Depth cameras**. Cameras can retrieve depth information about a scene either using a particular sensor or by running a stereo algorithm on the colour frames.

Given the diversity in sensor technology, human activity can also be classified in terms of the type of sensors that are used for activity monitoring. Chen et al. [167] classified human activity recognition in two categories: *video-based* HAR and *sensor-based* HAR. Video-based HAR analyses motions and behaviours of humans from videos or images, while sensor-based HAR

analyses motion data from various types of sensing devices such as accelerometers, gyroscopes, Bluetooth, etc.

The video-based approach is one of the pioneer approaches in HAR. It is easy to use and can provide good results in capturing information about the activities. However, there are some issues related to this approach [72]. For example, the use of cameras in private environments such as smart homes for healthcare purposes raises privacy concerns. Zhang et al. [196] outlined some domain-specific problems. For example, depending on the camera and scene, the image's background can be highly dynamic. This means that the amount of irrelevant information for identifying an action can be very high. Also, cameras can fail during nighttime if there is no proper light. For that reason, most of the research in HAR has shifted towards sensor-based approaches due to the low cost and advances in sensor technology. We are particularly interested in sensor-based HAR because they have shown excellent results in HAR applications and its application is rising rapidly [83]. Therefore, for the remainder of this chapter, we will focus on sensor-based HAR, which is the main topic of this thesis.

2.3 Sensor-based Human Activity Recognition

Sensor-based human activity recognition is to extract high-level descriptions (i.e., activities) from low-level sensor data [144]. There are two main categories in terms of sensor deployment strategies. The first approach is deploying sensors in the environment (ambient sensors). In the second approach, the users carry the sensors (wearable sensors). However, Wang J. et al. [167] extended these classifications as follow:

- **Body-worn Sensors**. In this approach, accelerometer, magnetometer or gyroscope sensors are attached to a user as they perform an activity. Accelerometers can often be found on smartphones, watches, bands, glasses and helmets. Gyroscopes and magnetometers are also frequently used together with accelerometers to recognise activities of daily living (ADL) and sports. Despite their wide use, a major problem with this approach is that sometimes wearing a tag is not feasible [21]. For instance, a user can forget to wear the bands or glasses or a user can place the phone in different parts of the body; for example, in their trouser pocket or shirt pocket, which will give different measures of the activity.

- **Object Sensors**. Sensors are attached to objects of daily use and are used to detect the movements of those objects to infer human activities. For example, the accelerometer attached to a cup can be used to detect *drinking water* activity. Another example is Radio Frequency Identifier (RFID) tags, which are deployed in smart home environments; for example, Jayatilaka et al. [74] employed a smart cup tagged with passive RFID tags to recognise fluid.

- **Ambient Sensors**. Sensors in the environment are deployed when a user performs any activity. Ambient sensors capture the interaction between humans and the environment. Examples of ambient sensors are radar, sound, pressure, and temperature sensors. They capture changes in the environment where they are deployed to infer activities. This approach does not require the user to carry any device while doing any activity. However, the deployment is also tricky and is easily affected by the environment.

- **Hybrid Sensors**. It has been shown that combining different types of sensors can improve the accuracy of HAR [167]. For example, ambient sensors can be used together with object sensors to record object movements and environment state.

The diversity of sensors leads to high complexity. Different sensors produce different data types, including binary, continuous numeric, and featured values [185]. The use and application can vary significantly due to their different modalities, output signal, size, and costs. The first attempts in activity recognition are related to home automation, and various location-based applications aim to adapt systems to users' locations [21]. Several researchers used RFID to detect the environment's interactions through object use [123, 20, 50, 121, 67]. Gaddam et al. [51] presented a smart home system for assisted living. The system monitors the use of electronic appliances, the water usage with water flow sensors and the bed usage for determining the sleeping pattern of the elderly. Many researchers have investigated gesture recognition and activity recognition from still images and video in stationary settings. To mention some examples, Turaga et al. [156] presented several approaches to analyse human activities in videos and classified them according to their ability to handle different degrees of complexity. Some other researchers employed state-of-the-art techniques for gesture recognition, hand gestures and facial expressions [106, 1]. However, interests in recognising activities in unconstrained daily

life settings caused a shift toward using inertial sensors worn on the body, such as accelerometers or gyroscopes.

Sensors have their own technical advantages and limitations. For example, ambient sensing has fewer privacy-related problems compared to vision-based systems. This caused studies to focus more on device-free (dense sensing) technologies. Dense sensing-based monitoring makes it more suitable to create ambient intelligent applications such as smart environments because of its low-cost, low-power characteristics [167]. In the early 2000s, activity recognition and tracking of postures and gestures were done with motion sensors attached to the user [21]. Tapia et al. [155] used environmental state-change sensors to collect information about interaction with objects and recognise activities of interest to medical professionals. Wilson et al. [173] used binary sensors, motion detectors, break-beam sensors, pressure mats, and contact switches for simultaneous tracking and recognising activities. Wren et al. [176] used passive infrared motion sensors networks to identify low-level and mid-level activities such as walking and visiting, respectively.

In more recent studies, sensors have been used in several real-world applications such as monitoring daily activity to support medical diagnosis, rehabilitation, or to assist patients [72]. For example, healthcare support in smart homes can be used to provide remote healthcare services or emergency support to elderly and disabled people. It can offer patient-monitoring services to identify health conditions, ensure assisted services, and generate local warnings [72]. In the industrial sector, wearable computing is used in assembly line operations of blue collar workers [102]. In sports and the entertainment sector, wearable gyroscopes and acceleration sensors are used to capture people's movements while doing sports. The information provided by these sensors is of special interest for clinical studies and performance tracking [88].

Other important research projects began in 2000s, including the Gator-Tech [66] smart house built by the University of Florida for research on ambient assisted living. The Aware-home project was developed by the Georgia Institute of Technology [82]. They used ceiling-mounted cameras and RFID sensors for localization purposes. In terms of activity recognition purposes, one of the pioneering studies is the House_n project developed by the Massachusetts Institute of Technology. Tapia et al. [155] installed reed switches and piezoelectric switches in different parts of the house and appliances such as microwaves, refrigerators, stoves, etc. to detect more than

20 activities. In The Center for Advanced Studies in Adaptive Systems (CASAS) project [34], 15 different activities were monitored using a smart home testbed, which was equipped with motion and temperature sensors, as well as analogue sensors that monitor water and stove burner use. Gaddam et al. [51] introduced a smart home monitoring application for assisted living. The sensors provide information used to monitor elderly people by detecting any abnormality pattern in their daily activities. Other examples of living laboratories for human activities recognition are inHaus [63], DOMUS [57], and iDorm [125].

There is also an increasing interest in acoustic sensing of the activities in smart environments [68]. Understanding speech and ambient sound can be beneficial for many healthcare applications. For example, AuditHIS system performs real-time sound analysis from microphones placed in a smart home [158]. Another example is a multi-modal system proposed by Karpov et al. [77]. They collected an audio corpus containing five spoken commands and 12 non-speech acoustic events for different activities. They defined alarming speech and audio events such as "Help", "Problem", "Crying" and "Key/ object drop".

In the wearable sensing category, the increased number of smartphones with sensing capability has made it possible to use them for human activity recognition purposes [21]. It is possible to recognise human activities by collecting data through GPS sensors, microphones, cameras, light, proximity, etc. Many authors developed wellbeing applications to recognise activities automatically [21]. For example, Pinky et al. [76] proposed an activity recognition system working on Android platforms that supports online training and classification. The system can recognise four main activities: walking, running, standing and sitting.

More recently, activity recognition is a crucial component in many consumer products such as game consoles which rely on the recognition of body movements to change the game experience. This led to a vast number of applications, such as personal fitness products. For example, the Philips DirectLife or the Nike+ running shoes integrate motion sensors to offer athletes feedback on their performance [21].

In Figure 2.1, we propose human activity recognition classification. First, we classify human activity recognition in terms of the sensor type used for activity monitoring [167]. The first category is vision-based, which uses visual sensing technology and the second category is sensor-based, which employs sensor network technologies. We further classify sensor-based

human activity recognition in two main branches:

- **Wearable sensing** refers to sensors carried by the users. These can be sensors positioned directly or indirectly on the user. For example, the most popular wearable sensor for activity recognition is the accelerometer. However, the increased use of smartphones has made it possible to collect data and automatically recognise activities. In this sense, we can differentiate between sensors positioned directly on the user (body-worn) such as accelerometers and those indirectly positioned on the user (object-worn sensors) such as smartphones or smartwatches.

- **Ambient sensing** refers to sensors deployed in an environment. Ambient sensors capture the interaction between humans and the environment. This interaction can be with other humans or with objects. We classify ambient sensing into three sub-categories: interaction-based sensors placed on specific objects, device-free such as WiFi and acoustic sensors such as microphones.

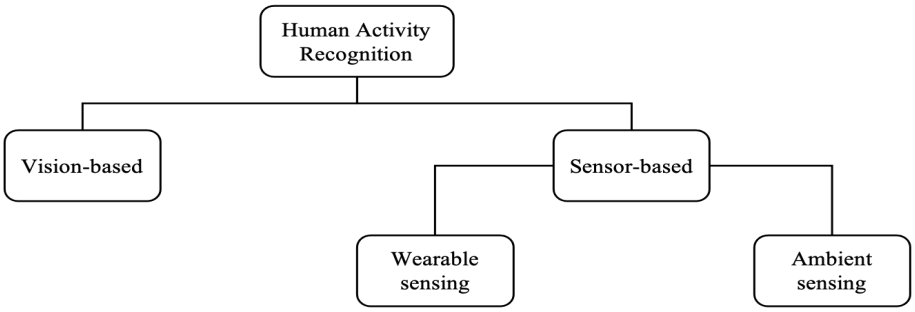


Figure 2.1: Human activity recognition classification

Table 2.2 provides the advantages and disadvantages of each sub-category along with examples of sensor technologies. As we can see, sensor-based approaches use different kinds of sensors such as accelerometers, motion sensors, pressure sensors, and RFID tags, for recognising daily activities. However, it is difficult to have a unique classification because of the different types of sensors and their output signals, size, weight, cost, etc. Nevertheless, extensive research has been undertaken to investigate their use to accurately recognises human activity.

In the following section, we present the evolution of sensor-based human activity in terms of literature review and applications.

Approach	Category	Subcategories	Technologies	Advantages	Disadvantages
Vision-Based	Video	Video	Camaras	High Accuracy	High cost, privacy issue
Sensor-based	Wearable sensing	Body-worn sensors	Accelerometers, magnetometer, gyroscope	Low cost	Constraint to wear the device
		Object-worn sensors	Smart Phones	Low cost	Constraint to have the device
	Ambient sensing	Interaction-based	Accelerometer, RFID	Low cost	Environmental interference and constraint to use the tagged objects
		Device free	WiFi, RFID	Low cost	Environmental interference
		Acoustic	Video, radar, microphones	Low cost	Environmental interference

Table 2.2: Technology, advantages and disadvantages of sensor classification for human activity recognition.

2.4 Sensor-based Activity Recognition Evolution

A considerable amount of work has been done in human activity recognition for the last decade. Demrozi et al. [40] identified 293 published papers, of which 46 are survey papers published since 2015. These papers can be categorised based on the data sources and the activity recognition algorithm. In the following, we describe the most relevant surveys related to sensor-based, we excluded papers related to video-based HAR.

Chen et al. [167] presented a survey where they classified sensor-based approaches in two main categories: 1) vision-based vs sensor-based, and 2) data-driven based vs. knowledge-driven based. In the former, different techniques are discussed, which use wearable sensors and dense sensing. In the latter, the authors discuss generative modelling and discriminative modelling to categorise data-driven approaches and for knowledge-driven approaches, techniques are further divided into logic-based, ontology-based, and mining based methods.

Wang J. et al. [167] have introduced the first survey to describe different deep learning approaches for human activity recognition using sensors. They classify the literature in activity recognition in three categories: sensor modality, deep model, and application area. They further classify modality literature in four categories: body-worn sensors, object sensors, ambient sensors, and hybrid sensors. For deep models, the authors categorised the models in discriminative deep architecture, generative deep architecture, and hybrid deep architecture. Finally, with respect to the application area, the literature is classified as activities of daily living, sleep, sports, and health.

Wang Y. et al. [171] presented the state-of-the-art of wearable sensor modality. They focus on

the techniques associated with each step of HAR in terms of sensors, data preprocessing, feature learning, and classification. They discussed the pros and cons of hand-crafted features and the use of conventional and deep learning methods for recognition tasks. They also summarised applications of HAR in healthcare and proposed some research challenges.

Sousa et al. [146] have presented a complete historical review and evolution of HAR based on smartphones. They also described each step of the methodology commonly used to recognise human activities with smartphones equipped with inertial sensors. They presented two approaches to extract features based on shallow and deep learning algorithms.

Nweke et al. [114] have provided in-depth analysis of data fusion and multiple classifier techniques for human activity recognition, emphasising mobile and wearable devices. They reviewed different deep learning algorithms for HAR, empathising in strengths and weaknesses of these methods and they provide open research challenges related to data collection.

Finally, Ramasamy et al. [131] reviewed recent machine learning algorithms such as deep learning, transfer learning, and active learning. They discuss the state-of-the-art techniques and highlight fundamental problems and challenges as a guide for future research directions.

Various data- and knowledge-driven techniques have been applied to human activity recognition, including ontological reasoning, Naïve Bayes, Decision Trees, Hidden Markov Models (HMM), Conditional Random Fields (CRF), Neural Networks, and Support Vector Machines (SVM) [24, 185]. Data-driven approaches rely on large datasets from which a model for a specific problem is learnt [136]. However, their performance depends on the number of training samples available. Also, data-driven approaches do not incorporate a semantic structure of the recognised activities, which, if present, would have allowed reasoning about the activities being executed and user goals, situations, and causes of behaviour. To address these limitations, knowledge-driven approaches rely on symbolic models describing the possible behaviours to reason about the user's actions and situation [136].

Knowledge-driven activity modelling is motivated by the diversity of activities of daily living and real-world applications [24]. An activity can be performed in different ways. For example, a person can enjoy walking on a treadmill while others enjoy walking outdoors. Such domain-dependent activity-specific prior knowledge provides valuable insights into how individuals in specific situations can perform activities. Knowledge-driven activity approaches use domain

knowledge to perform activity modelling and pattern recognition [24].

The knowledge approach can be modelled in different forms, such as schemas, rules or networks. An example is mining-based. This approach creates activity models by mining existing activity knowledge from publicly available sources. More specifically, a set of objects used for each activity are mined to extract information about their usage from text corpora. Then, the models use co-occurrences and associations to estimate the objects used during the performance of the activity. Perkowitz et al. [122] tagged each word in a sentence with its part of speech and customised a regular expression to extract objects used in an activity. They used the Google conditional probabilities applications programming interfaces to determine the probability of object usage.

Other approaches combine symbolic models with probabilistic reasoning [191, 87, 192]. These approaches, known as computational state space models (CSSMS) [136], use concise rule-based representations of the possible actions and the relevant context and probabilistic inference engines to reason about the observed actions and context in a probabilistic manner. The rules are used to generate probabilistic models with which the system can infer the user actions and goals. CSSMS rely on prior knowledge to obtain the context information needed to build user actions and the problem domain. The prior knowledge is provided in the form of precondition-effect rule by a domain expert.

Yordanova et al. [87] presented a tool support for human activity recognition using computational casual behaviour models to describe activities and probabilistic inference machines. Symbolic human behaviour models allow the representation of user actions and the reasoning over them to infer not only current user actions but also to what more complex activity it belongs. Computational Casual Behaviour Models (CCBMs) consist of a symbolic casual human behaviour model and an observation model that are translated into a probabilistic inference system. The symbolic model consists of two parts. The first one is the domain description that contains the available user actions represented as precondition-effect operators, the object types used and the domain constants. The second part is the problem description that contains the problem constants, the initial state and the goal state [87].

The logic-based approaches use various logical formalisms [24]. For example, Wobke [175] used situation theory to address the different probabilities of inferred plans by defining a partial

order relation between plans in terms of levels of plausibility. Some other works used logical theory of actions, such as the event calculus for activity recognition [142, 25].

More recent works introduced activity ontologies to analyse social interaction in nursing homes, car park monitoring scenarios, classify meeting videos, and analyse activities with surveillance cameras [23, 64, 54, 4]. The ontology-based approach arises from the need to have a commonly agreed explicit representation of activity definitions or an ontology [24]. Activity recognition is performed using rule-based algorithms and finite-state machines [64, 4]. Chen et al. [26] constructed context and activity ontologies for explicit domain modelling. They mapped sensor activations over a period of time to individual contextual information. Ye et al. [4] developed a top-level ontology to model and reason on domain knowledge precisely and traceable, serving as a conceptual backbone for developing domain and application ontologies for smart environments.

Given the diversity of techniques, we divide the literature review for human activity recognition into two subsections: classical machine learning approaches, and deep learning approaches.

2.4.1 Classical Machine Learning Approaches

Machine learning methods are driven by data; that is, activity models learn from large-scale datasets of users' behaviours [24]. These methods involve the creation of probabilistic or statistical models, followed by a training and inference process based on statistical classification [24]. The most straightforward approach used for activity recognition is the Naïve Bayes classifier (NBC). The dependence of observations on activity labels is modelled as a probabilistic function that can be used to identify the most likely class given a set of observations [24]. Several works used NBC [10, 19, 32, 155, 147], which achieved good performance when large amounts of sample data are provided.

Ye et al. [187] have applied ontologies to support automatic sensor data segmentation for multi-user concurrent activities. They have employed the Pyramid Match Kernel to separate the activities with similar patterns to a certain degree. This is achieved by calculating the difference of sensor feature distributions in a hierarchical manner. However, they still cannot distinguish the users for the same activities, for example, identifying which user is cooking.

van Kasteren et al. [162] have applied Hidden Markov Models (HMM) to model sequential

relationships of sensor data and activities. The HMM is trained to obtain three probability parameters, where the prior probability of an activity represents the likelihood of the user starting from this activity; the state transition probabilities represent the likelihood of the user changing from one activity to another, and the observation emission probabilities represent the likelihood of the occurrence of a sensor observation when the user is conducting a certain activity. Even though the HMM has built the temporal probabilistic model between activities and sensor observations and thus successfully recognised activities, it has achieved low accuracy on minority activities [160]. Nguyen et al. [112] applied hierarchical Hidden Markov Models (HHMM) to recognise primitive and complex behaviours of multiple people. They construct a unified graphical model composed of a set of HHMMs with data association.

Some machine learning methods suffer from scalability and reusability problems. They require large datasets for training and learning. Also, it is difficult to apply learnt activity models from one person to another [24]. Deep learning models overcome some of these limitations [167] and will be explained in the next section.

2.4.2 Deep Learning Approaches

Deep learning techniques offer an advantage over data-driven and knowledge-driven techniques. They do not rely on heuristic or hand-crafted methods to extract features, nor do they rely on human experience or domain knowledge [167].

First works using Deep Neural Network (DNN) show that with more layers, DNN is more capable of learning from large data [163, 165]. Oniga et. al [117] presented a recognition system from arm posture, body postures and simple activities like standing, sitting, walking, running, etc. using neural networks. Their approach consists of a two-layer feed-forward network, with sigmoid activation function on both the hidden and output layers. They also presented the data acquisition prototype which gathers data of the patient and recognises the abnormal status of the patient's health. These works indicated that, when the HAR data is multi-dimensional and activities are more complex, more hidden layers can help achieve better training performance [14].

More recently, Convolutional Neural Networks (CNNs) have become a popular approach to extract features from low-level sensor data automatically. Bevilacqua et al. [15] proposed to

use of CNNs to classify human activities. They collected 16 activities from the Otago exercise program. They trained several CNNs with signal coming from different sensors. Two sensors were placed on the distal third of each shank, superior to the lateral malleolus, two sensors centred on both feet, in line with the fifth metatarsal head, and one sensor placed on the lumbar region at the fourth lumbar vertebrae.

Autoencoders were used to learn more advanced feature representations. Stacked autoencoders (SAE) were first used by [7, 166] for human activity recognition. The advantage of SAE is that it can perform unsupervised feature learning for HAR, which can be used as a feature extraction tool. Later on, Li et al. [94] investigated the sparse autoencoder by adding Kullback–Leibler divergence and noise to the cost function.

Some other works introduced Restricted Boltzmann machine (RBM) [124, 65, 90] for activity recognition. RBM is a bipartite, fully connected, undirected graph consisting of visible and hidden layers. RBM can also perform unsupervised feature representation [167].

Recurrent Neuronal Networks (RNN) is widely used in speech recognition. However, some approaches implemented long-short term memory (LSTM) models combined with RNN for HAR tasks to deal with resource-constrained environments while still achieving good performance [167]. Singh et al. [145] introduced a recurrent neural network to classify human activities without using any prior knowledge. They used Long Short Term Memory (LSTM) to model temporal sequences and learn long-term dependency problems. LSTM has shown promising results in pattern recognition that are defined by temporal distance.

2.5 Challenges

The recent advance in data mining, machine learning, and deep learning has made it possible to learn complex correlations between low-level sensor data and high-level activities. However, due to the complexity of human actions, activity recognition still faces some challenges. Some of these challenges are shared with the general field of pattern recognition. However, it also faces a number of unique challenges [167]. First, we describe the common challenges with pattern recognition, and secondly, we describe the specific challenges related to HAR.

2.5.1 Common Challenges

Intraclass Variability. There are many ways to perform a simple activity, for example, people may walk at different paces. Recognition systems should be able to recognise the same activity performed differently by different individuals. Intraclass variability can also occur when an activity is performed differently by the same individual [167]. This can be caused by stress, fatigue, or the emotional state in which the activity is performed. For example, walking on a treadmill can be different from walking outdoors.

Interclass Similarity. Interclass similarity occurs when activities have similar sensor data characteristics [197]. To deal with this problem, accurate and distinctive features need to be designed and extracted from sensor readings. A human activity recognition system must be general enough to model all possible changes of a particular activity and distinguish between them.

The NULL Class Problem. Only a few parts of a continuous data stream are relevant for HAR systems. Therefore, activities of interest can easily be confused with activities with similar patterns but are irrelevant to the application in question [167].

Multisubject Interactions. Most researches focus on identifying low-level human activities such as jumping, running, sleeping, etc. Usually, the recognition is done with a single subject without any human-human or human-object interactions [197]. However, in real-world applications, activities are performed with the interaction of other persons and objects. Therefore, it is challenging to track multiple subjects or to recognise group activities. To recognise group-based human activities, a higher level representation must be introduced, which can model the activity as a composition of simpler activities.

2.5.2 Challenges Specific to HAR

Activity Definition. Human activity is complex, diverse and can be performed differently. Therefore, the first challenge is to define the activities under investigation and their specific characteristics. Katz et al. [107] develop the Activities of Daily Living (ADLs) index that includes basic actions that involve caring for one's self and body, including personal care, mobility, and eating.

Class Imbalance. One of the challenges is distinguishing activities with subtle differences and imbalanced distributions, which can have a significant implication in health-related applications. Only a few activities often occur in long-term behavioural monitoring, while most activities occur infrequently. For example, life-threatening situations like falls or heart attacks are often not frequent and may have subtle differences from other daily activities. Recognising them effectively will enhance the robustness of an activity recognition system.

To illustrate this challenge, we use the following example. Figure 2.2 presents the distribution of a set of concurrent activities from two users recorded in a smart home setting [33] and Figure 2.3 presents a confusion matrix of recognising these activities from a *K*-Nearest Neighbour (KNN) technique. As we can see, KNN can fairly well recognise the majority activities like “Sleep" and “Work" and the activities with distinct patterns like “Bed_Toilet_Transition". However, it performs poorly on (1) distinguishing the activities from the same user occurring in the same area; for example, is a user wandering or working in the bedroom?, and (2) differentiating the users for the same type of activities performed in a public area; for example, is the user *R1* or *R2* preparing the meal?. First, some activities do not often occur, especially the wandering in the room being the least reported activity, which results in too few samples to train a reliable classifier. Secondly, these activities can have fewer discriminative patterns than their majority counterpart; they might activate the same set of sensors but with little difference in distributions. For example, the “Wander" activity fires a collection of sensors that significantly overlap with the sensors activated on “Work" and “Sleep".

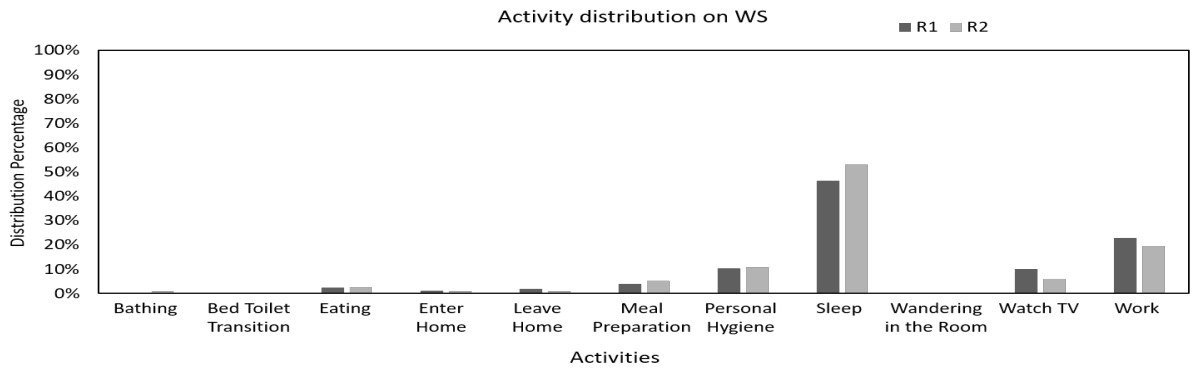


Figure 2.2: Distribution of concurrent activities of two different user

Ground Truth Annotation. A major challenge in current activity recognition research is to collect sufficient labelled data in the environment to train classification models. This task can

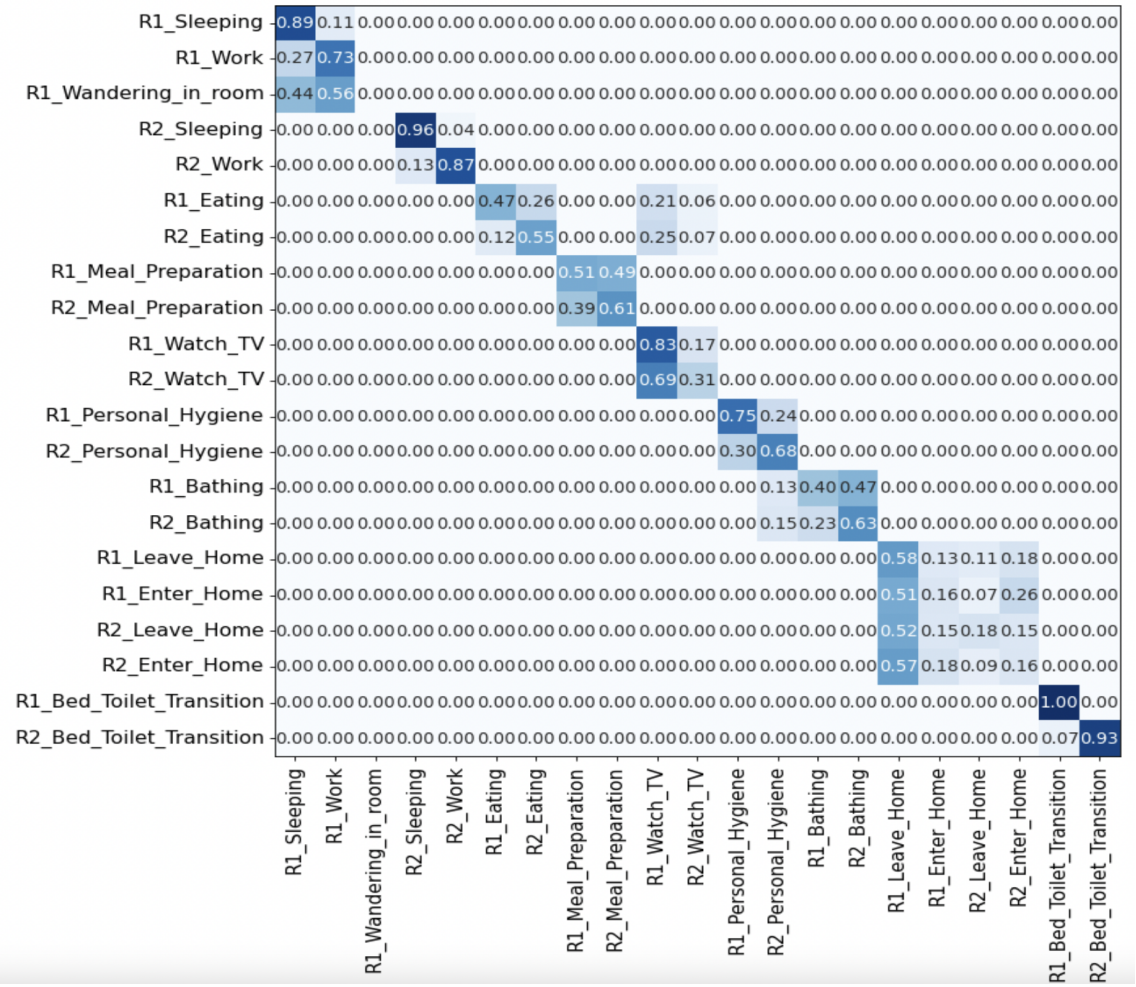


Figure 2.3: Activity recognition on a set of two-user concurrent activities

be expensive and the lack of training labelled samples can compromise the performance of the classifier. Also, motion data gathered from accelerometer or gyroscope sensors are often more difficult to interpret than data from other sensors, such as cameras. In a laboratory setting, annotation can be done based on video footage. However, in a daily life setting, annotation is more difficult.

Data Collection. The challenge is to collect datasets on which HAR systems can be evaluated. The research community has not yet started a joint effort to collect datasets of human physical activity. Using standard datasets is crucial for reproducible research [167].

Data Representation. Finally, there are also challenges in sensor-based HAR associated with information representation [15]. State-of-the-art approaches are based on engineered features. However, these features are mainly based on heuristic methods and frequently, the feature

extraction process requires a deep knowledge of the application domain. Moreover, traditional methods often do not perform well on dynamic data or scale for complex motion patterns.

2.6 Applications

Despite the challenges highlighted above, successfully recognising human activities leads to many useful applications. In this section, we provide an overview of relevant applications.

- **Elder Health Care**. With the advancement in medical science and technology, life expectancy is increasing. With the ageing population comes the problem of medical costs and caring for old people. In recent years, many different technologies have been developed to assist humans in their daily lives. These technologies are called Ambient Assisted Living and are helping people with remote monitoring, medication management, medication reminder, exercise management, and independent living. Human activity recognition can help elderly people to live independently. By monitoring human activities, HAR is helping to reduce medical expenses, reduce the demand of health givers, and improving quality of life.

- **Intelligent Environment**. Building smart environments have become very popular, such as smart homes, offices, and smart health centres. In smart environments, activities performed by the residents are learnt and the system adapts itself. For example, in a smart home electric systems can turn on/off depending if the residents are in the house, in a smart health care centre patients are monitored remotely, or a smart refrigerator can monitor food usage.

- **Security and Surveillance**. Although surveillance cameras can watch 24/7, there is still a need for human effort to monitor and detect any suspicious activity. Therefore, many video-based solutions have been proposed to analyse videos or images from the cameras to recognise the activities, and report any suspicious incident.

- **Human-Computer Interaction**. Nowadays, people can interact with machines by making a gesture or performing a specific activity for giving a command to the machine. For example, games consoles can recognise activities while the user interact with the game.

HAR is also helping robots to interact with humans by recognising their activities or even help them with daily activities. One example is the robotic vacuum cleaner, an autonomous robotic hoover that has intelligent programming.

- **Shopping Experience**. Analysing and understanding shopping behaviour of customers has become very popular. Customers can either be tracked online or indoors. With online stores, customers' behaviour is tracked by placing tags on the website and analysing the users' clicks, products review, and shopping carts. This information along with sales history can provide useful information about interests, products bought together, products compared together, items that users ignore, etc. This information can be used to optimise websites and improve users' shopping experience. Inside stores, surveillance cameras and other sensors such as RFIDs can help recognise shoppers' behaviours. For example, shopping patterns can be detected by analysing which aisle the customer visited, which products they select and which ones they actually place in their baskets. This information can be used to improve shopping experience or even to decide how the products should be placed on the shelves.

2.7 Discussion

We have extensively presented basic concepts on human activity recognition, different sensor technologies, literature review, challenges and potential applications on activity recognition. Over the past decades, many solutions have been proposed to recognise human daily activities. Some of these techniques use surveillance cameras but, as mentioned in earlier sections, vision-based techniques have many limitations. Some other techniques use dense sensing and deploy different sensors when a user performs an activity. Some methods use a hybrid approach and combine wearable and object-tagged sensors. Finally, some techniques require that the user wears a device or sensors attached to daily use objects.

Given the complexity and variety in HAR, in this thesis, we focus on sensor-based human activity recognition and our main interest is ambient assisted living in smart home. Ambient sensors, including positioning or pressure sensors and RFID sensors, are deployed to detect the whereabouts of older adults and their interaction with everyday objects. Wearable sensors, such

as accelerometers and gryoscopes sensors, are commonly used to recognise activities of daily living.

Machine learning and deep learning techniques can help us infer their current activities (e.g., making a meal or performing personal hygiene) and further detect changes in their health conditions over time. Information to train learning models can be collected through various approaches and technologies but, as mentioned in Section 2.5, a major challenge is to collect sufficient labelled data in the environment to train a classification model. This task can be expensive and the lack of training labelled samples can compromise the performance of the classifier. To deal with the expensive challenge of collecting sufficient labelled data, transfer learning can be used to apply knowledge learned from the source domain to the target domain. However, the research question that comes to light is: *Is it possible to relieve the annotation burden on individual users but still be able to build a robust activity recognition model by sharing and transferring activity models across users, even though the sensor deployments and operating environments are different*?

Also, in real-world applications, sensors can produce imperfect data. For example, they are susceptible to breakdowns or may suffer interference. These issues will generate noisy sensor data, or the data distribution can change over time. Most of the existing approaches [91, 168] assume that the sensor data distribution is the same as that used in the model training process. So, *can we build a system that performs robustly in the presence of noise*? The following chapters will address how these challenges are solved in our work.

Chapter 3

Transfer Learning and Domain Adaptation

Transfer Learning is a sub-area of machine learning that focuses on re-utilising knowledge between tasks [168]. Domain adaptation is a branch of transfer learning where a distribution mismatch between two domains is assumed. It has been extensively studied in many areas, including speech and language processing, and more recently, computer vision [168]. In this chapter, we provide an overview of this branch, and we explain basic terminologies necessary to understand transfer learning. Afterwards, we provide a formal definition of domain and domain adaptation, and we focus on the methodologies developed in human activity recognition. Among them, of particular relevance to our work is unsupervised domain adaptation.

3.1 Introduction

Traditional approaches to machine learning assume that the training data and test data are drawn from identical distributions [30]. However, this assumption is not always possible in many real-world applications. To tackle this challenge, transfer learning techniques have been proposed to transfer the knowledge learnt from one domain (known as the source domain) to another domain (known as the target domain), assuming that there is some relationship between the source and target domains.

Research on transfer learning dates back from the 1980s [193, 105]. These first works focus on human learning mechanisms, highlighting the bias in machine learning as a fundamental part of the learning process. For example, we may find that learning to recognise apples might help

to recognise pears. Transfer learning is motivated by the fact that people can apply knowledge learned previously to solve new problems [120]. Since 2005, it has been widely discussed in the Advances in Neural Information Processing Systems (NIPS) Workshop until now [42]. Current transfer learning approaches focus on learning standard or latent statistical features from both source and target tasks in multitasking [152]. Before mathematically defining transfer learning, we firstly introduce the concepts of domain, task, and dataset defined by Pan et al. [120].

A **domain** consists of two main components: a feature space of inputs $\mathcal{X}$ and a marginal probability distribution of inputs $P(X)$, where $X = x_1, ..., x_n \in \mathcal{X}$ is a set of learning samples. For example, if our learning task is a document classification, and each term is taken as a binary feature, then $\mathcal{X}$ is the space of all document vectors. The marginal distribution is about the value of each feature and describes the probability distribution of the variables in the dataset.

Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a **task** consists of a label space and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is not observed but can be learned from the training data, which consist of pairs (x_i, y_i) where $x_i \in X$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x.

Finally, a **dataset** is a collection of data characterised by a specific domain and a specific task. In transfer learning, the source and target dataset can be different either in the source and target domains or in both tasks.

In this thesis, we denote the source domain as $\mathcal{D}_s = \{(x_1^s, y_1^s), ..., (x_{n_s}^s, y_{n_s}^s)\}$, where $x_i^s \in \mathcal{X}^s$ is the data instance and $y_i^s \in \mathcal{Y}^s$ is the corresponding class label. Similarly, we denote the target domain as $\mathcal{D}_t = \{(x_1^t, y_1^t), ..., (x_{n_t}^t, y_{n_t}^s)\}$ where $x_i^t \in \mathcal{X}^t$ is the data instance and $y_i^t \in \mathcal{Y}^t$ is the corresponding output. Now that we have defined important concepts we can provide a formal definition of transfer learning [30].

Definition 1. Given a source domain $\mathcal{D}_s$ and a learning task $\mathcal{T}_s$, a target domain $\mathcal{D}_t$ and a learning task $\mathcal{T}_t$, transfer learning aims to help improve the learning of the target predictive function $f_t(\cdot) \in \mathcal{D}_t$, using the knowledge in $\mathcal{D}_s$ and $\mathcal{T}_s$, where $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{T}_s \neq \mathcal{T}_t$.

The source domain can differ from the target domain by having a different feature space, different probability distribution, different label space or label distribution. However, all transfer learning problems assume that there exists some relationship between the source and target

domains which allows for the successful transfer of knowledge between them [131]. The question that arises is *how much two domains differ from each other*. Wouter et al. [85] introduced a *symmetric difference hypothesis* divergence ($\mathcal{H}\Delta\mathcal{H}$-divergence). This measure takes two classifiers with VC dimension d and looks at to what extent they disagree with each other:

$$d_{\mathcal{H}\Delta\mathcal{H}}(P_s, P_t) = 2\text{sup}_{h,h' \in \mathcal{H}} |\Pr_s[h \neq h'] - \Pr_t[h \neq h']| \tag{3.1}$$

where h refers to the decision made by the classifier, the probability Pr is calculated as the following integral: $\Pr_s[h \neq h'] = \int_{\mathcal{X}} [h(x) \neq h'(x)] P_s(x) \mathrm{d}x$, P_s and P_t are the marginal distributions of the source and target domain respectively. The sup stands for the *supremum*, which in this context finds the pair of classifiers h, h' for which the probability is largest and returns the value of that difference [85].

Given u_s and u_t samples of size m from P_s and P_t, respectively, and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(u_s, u_t)$ the empirical $\mathcal{H}\Delta\mathcal{H}$-divergence between samples, the following equation is true

$$d_{\mathcal{H}\Delta\mathcal{H}}(P_s, P_t) \leq \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(u_s, u_t) + 4\sqrt{\frac{dlog(2m) + log(\frac{2}{\delta})}{m}} \tag{3.2}$$

for any $\delta \in (0,1)$, with probability at least $1-\delta$. Equation 3.1 shows that the empirical $\mathcal{H}\Delta\mathcal{H}$-divergence between two samples from distributions P_s and P_t converges uniformly to the true $\mathcal{H}\Delta\mathcal{H}$-divergence for hypothesis classes of finite VC dimension d [12].

Given the error of the join hypothesis, $e^*_{s,t} = \min_{h \in \mathcal{H}} [e_s(h) + e_t(h)]$, and the $\mathcal{H}\Delta\mathcal{H}$-divergence, a bound can be found on the difference between the true target error, e_t of a trained source classifier, $\hat{h}_s = \arg\min_h \hat{R}_s(h)$, and that of the optimal target classifier, $h^*_t = \arg\min_h R_t(h)$. $R(h)$ refers to the expected loss, also called the risk, of a particular classifier and $\hat{R}(h)$ is the *empirical risk* of the target classifier. This bound has the following form [12]:

$$e_t(\hat{h}_s) - e_t(h^*_t) \leq e^*_{s,t} + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(P_s, P_t) + \mathcal{C}(\mathcal{H}) \tag{3.3}$$

where $\mathcal{C}(\mathcal{H})$ describes the complexity of the classification task. The bound states that, the larger $e^*_{s,t}$ and $d_{\mathcal{H}\Delta\mathcal{H}}$ are, the less a source classifier will generalise in the target domain.

The challenge that remains is to design an adaptation strategy to improve adaptation performance that narrows the generalisation error bounds. Therefore, the following section focuses specifically on the topic of dataset shift problems. First, we describe the causes and afterwards, we describe the three main variations to which dataset shift can diverge to.

3.2 Dataset Shift

The term dataset shift was first introduced by Quiñonero-Candela et al. [128]. The problem of dataset shift is closely related to transfer learning. The former deals with relating information in two closely related environments to help with the prediction in one given the data in the other. While the latter deals with the general problem of how to transfer information from one environment to help with learning, inference, and prediction in a new environment [128].

3.2.1 Types of Dataset Shift

This subsection explains different kinds of shift that can appear in a classification problem.

- **Prior shift** refers to changes in the distribution of the class; that is, the prior probabilities of the classes are different $P_{\mathcal{Y}}^{tr} \neq P_{\mathcal{Y}}^{tst}$, where $P_{\mathcal{Y}}^{tr}$ and $P_{\mathcal{Y}}^{tst}$ are the prior class probability for the training and testing set respectively, but the conditional distributions are equivalent, $P_{\mathcal{X}|\mathcal{Y}}^{tr} = P_{\mathcal{X}|\mathcal{Y}}^{tst}$, where $P_{\mathcal{X}|\mathcal{Y}}^{tr}$ and $P_{\mathcal{X}|\mathcal{Y}}^{tst}$ are the conditional distribution for the training and testing set respectively [85]. For example, if the training set has equal prior probabilities on the number of spam emails received, we expect 50% of the training set to contain spam emails and 50% to contain non-spam. If, in reality, only 90% of our emails are spam, then our prior probability of the class variables has changed.

- **Concept shift** is not related to the data distribution or the class distribution but instead is related to the relationship between the two variables. For example, consider a medical setting where the aim is to make a prognosis for a patient based on their age, severity of their flu, general health, and socio-economic status. Originally, the classes are defined as “remission" and “complications". But, other aspects are counted as a form of ”complication”

during the test time and are so labeled. In this case, the data distributions remain constant while the posteriors probabilities change [85].

- **Covariate shift** is the most common data shift. It is given when the distributions on the training and test set do not match, the default assumptions of independent and identical distributed datasets are not valid [148]. From a domain adaptation perspective, the biased sampling corresponds to the source domain and the target domain to the unbiased sample. For example, a face recognition algorithm is trained in the source domain with a dataset that has a much more significant proportion of older faces in it.

3.2.2 Causes of Dataset Shift

In some machine learning models, there is an assumption that the distributions of the datasets do not change over time. If this is not true, and the distributions change, we need to model for that change. To do so, we first need to understand why such a shift may occur. Of the various causes of dataset shift, some can take place in the design process of training data sampling. Others can be categorised as environmental causes since the cause of shift is due to the inevitable changes over time of the environment's characteristics. Moreno-Torres et al. [108] presented the two most important causes:

1. Sample selection bias is when the discrepancy in distribution is due to the fact that the training samples have been obtained through a biased method. This causes the training samples to be selected non-uniformly from the population to be modeled.

2. Non-stationary environments occur when the training environment is different from the testing environment. Depending on the problem, this change can be different:

    - In $X \rightarrow Y$ problems, a non-stationary environment could generate covariate shift or concept shift. That is, changes in $P_{\mathcal{X}}$ or $P_{\mathcal{Y}|\mathcal{X}}$, respectively.
    - In $Y \rightarrow X$ problems, it could generate prior probability shift with a change in $P_{\mathcal{Y}}$ or concept shift with a change in $P_{\mathcal{X}|\mathcal{Y}}$

Non-stationary problems commonly appear in remote sensing applications, where a dataset collected in a given period of time for a specific environment is employed to train a classifier but,

when that classifier is deployed, mismatches may appear due to seasonal changes or because of a new distribution of the environment [5].

Generally, according to the different situation between source and target domains and tasks, different transfer learning approaches can be applied. The most typical approaches can be classified in the following categories [195]:

- **Statistical Approach** is employed at instance, feature and classifier level by measuring and minimising the divergence of statistical distributions between the source and target datasets. Typical methods are instance re-weighting [71], feature space mapping [118], and classifier parameter mapping [127].

- **Geometric Approach** assumes that the domain shift can be reduced by using the relationship of geometric structures between the source and target datasets. Subspace alignment [48], intermediate subspace [55], and manifold alignment [36] are commonly used methods.

- **Higher-level Representation Approach** finds a high-level representation that is representative, compact, and invariant between datasets. It does not require any labelled data but assumes that there are domain invariant higher-level representations between datasets. Examples of this approach are sparse coding [130], low-rank representation [143], deep neural networks [43], stacked denoising auto-encoders [27], and attribute space [2].

- **Correspondence Approach** constructs a relationship between domains by finding correspondence pair samples from different domains. The typical methods are sparse coding with correspondence [200], and manifold alignment [194].

- **Class-based Approach** assumes that labelled data is available from both domains and connects the source and target dataset by label information. Examples of methods used in this approach are feature augmentation [73], metric learning [137], linear discriminative model [181], and bayesian model [47].

- **Self labelling** trains a model using a labelled source dataset and creates pseudo labels for the target domain. Then the model is retrained with the target data and the pseudo labels. An example of this approach is self-training [38].

- **Hybrid Approach** combines two or more approaches to improve knowledge transfer between domains. Some examples are correspondence and class-based [41], statistic and class-based [44], higher-level representation and statistics [202].

In Table 3.1, we present an overview of transfer learning techniques. We classify the approaches based on two important concepts: (1) label availability; that is, if the source and target datasets are labelled, semi-labelled, unlabelled or any label is available, and (2) if the source and target domain have the same or different feature and label space. The following section presents a comprehensive review of domain adaptation in more detail. We mainly focus on unsupervised domain adaptation approaches.

Feature space	Label space	Source Data	Target Data	Approach Name
Same	Same	Labelled	Labelled	Supervised Domain Adaptation
			Semi-labelled	Semi-supervised Domain Adaptation
			Unlabelled	Unsupervised Domain Adaptation
			Not available	Domain Generalization
Different	Same	Labelled	Labelled	Supervised Heterogeneous Domain Adaptation
			Semi-labelled	Semi-supervised Heterogeneous Domain Adaptation
			Unlabelled	Unsupervised Heterogeneous Domain Adaptation
Same	Different	Labelled	Labelled	Sequential/ Online Transfer Learning, Few-shot Learning
			Unlabelled	Unsupervised Transfer Learning
			Not available	Zero-shot Learning
		Unlabelled	Labelled	Self-taught Learning
Different	Different	Labelled	Labelled	Heterogeneous Transfer Learning

Table 3.1: Transfer learning techniques classification.

3.3 Domain Adaptation

Given the situation when the distribution of training and test data do not match, we face the problem known as *domain adaptation*, a particular case of transfer learning. The challenge in domain adaptation is to overcome the differences between domains so that the classifier trained on the training dataset (source domain) generalises well on the test data (target domain) [85].

Domain adaptation can be categorised into *supervised* domain adaptation where labels on the target domain are available, *unsupervised* domain adaptation where labels on the target domains

are not available, and *domain generalisation* where a domain agnostic model is generalised by learning from multiple domains. In the following, we will briefly introduce representative approaches in the other categories and then focus on unsupervised domain adaption for accelerometer and sensor-based data.

3.3.1 Supervised Domain Adaptation

As labels are available in supervised domain adaption, it is possible to perform within-class adaptation. For example, Xu et al. [180] propose d-SNE where samples from both source and target domains are transformed to common latent space; i.e., stochastic neighborhood embedding (SNE) space, and then a modified Hausdoff distance is employed to minimise the distance between samples from the same classes but maximise the distance between samples from different classes. Morsing et al. [109] propose to deal with covariate shift by connecting samples in a penalty graph structure.

Conditional Generative Adversarial Networks (CGANs) proposed by Mirza el at. [104] extend the original model by introducing extra information to both the generator and discriminator. This additional information can be any kind of information such as class labels or data from other modalities. Both, the generator and the discriminator are multilayer perceptrons with Rectified Linear Units (ReLU) as the activation for hidden layers and sigmoid for the output layer. Their model can be used to learn a multi-modal model and in image tagging.

Odena et al. [115] have introduced a new approach called auxiliary classifier (AC-GAN) to improve image samples quality by adding more structure to the GAN latent space with a specialised cost function. They modified the standard GAN formulation to include a corresponding class label to every generated sample. The discriminator gives both a probability distribution over sources and a probability distribution over the class labels.

Mao et al. [100] propose the least square generative adversarial networks (LSGANs) that use the least square loss function for the discriminator to improve the learning process. The least square loss function moves the generated samples towards the decision boundary even though they are correctly classified. LSGANs are thus able to generate samples that are closer to real data.

3.3.2 Domain Generalisation

A classic approach in domain generalisation is to combine training samples from different source domains to train a classifier and regulate the weights of the classifier for an unseen target domain. CCSA (Classification and Contrastive Semantic Alignment) is one of the first deep learning techniques that tackle both domain adaptation and generalisation. It uses contrastive loss to encourage samples with the same class labels from different domains to be close in the embedding space [110]. Li et al. [93] employ an adversarial autoencoder to align distributions from different domains where Maximum Mean Discrepancy (MMD) is used to minimise the difference in distributions.

3.3.3 Unsupervised Domain Adaptation

In this section, we discuss existing methods for unsupervised domain adaptation. We divide the literature review into two sections: non-deep learning-based and deep learning models. In particular, we provide more details on the methods we will use for comparison.

3.3.3.1 Non-deep Learning Models

Feature transformation is a classic type in unsupervised domain adaptation, which maps the features of the source and target domain into a high-dimensional space. Previous work [119] has demonstrated that finding *good* feature representations can help reduce the difference in distributions between domains.

Hotelling et al. proposed [69] Canonical Correlation Analysis (CCA). The goal of CCA is to find a linear transformation of the source and target domains so that they are maximally correlated [69]. CCA is a representation learning technique that preserves the main characteristic of the relationship between the two domains. This method shares many mathematical similarities with dimensionality reduction techniques such as principal components analysis (PCA) and with regression methods such as partial least squares regression (PLS) [17].

Given two feature spaces, $\mathcal{X}_s$ and $\mathcal{X}_t$, CCA finds a canonical coordinate space that maximises correlations between the projections of the feature spaces onto that space. Assume that we

represent the linear combinations of these feature spaces as $\hat{\mathcal{X}}_s = \xi_s^T \mathcal{X}_s$ and $\hat{\mathcal{X}}_t = \xi_t^T \mathcal{X}_t$ such that $var(\hat{\mathcal{X}}_s) = \xi_s^T \Sigma_{\mathcal{X}_s} \xi_s$, $var(\hat{\mathcal{X}}_t) = \xi_t^T \Sigma_{\mathcal{X}_t} \xi_t$, and $cov(\hat{\mathcal{X}}_s, \hat{\mathcal{X}}_t) = \xi_s^T \Sigma_{\mathcal{X}_s, \mathcal{X}_t} \xi_t$.

The first pair of canonical variates $\hat{\mathcal{X}}_s^1$ and $\hat{\mathcal{X}}_t^1$ can be defined as linear combination vectors $\{\xi_s{}^1, \xi_t{}^1\}$ that maximises the correlation of $\hat{\mathcal{X}}_s$ and $\hat{\mathcal{X}}_t$:

$$\rho(\mathcal{X}_s, \mathcal{X}_t) = \max \text{corr}(\xi_s^T \mathcal{X}_s, \xi_t^T \mathcal{X}_t) = \max_{\xi_s, \xi_t} \frac{cov(\hat{\mathcal{X}}_s, \hat{\mathcal{X}}_t)}{\sqrt{var(\hat{\mathcal{X}}_s)}\sqrt{var(\hat{\mathcal{X}}_t)}}.$$

That is, the maximum canonical correlation is the maximum of ρ with respect to ξ_s and ξ_t. The solution to this problem can be done by computing the $\mathcal{QR}$ decomposition of the transposed feature space matrices, $\mathcal{X}_s^T$ and $\mathcal{X}_t^T$ [46].

Long et al. proposed [97] a Joint Distribution Adaptation (JDA) approach, which aims to jointly adapt both the marginal distribution and conditional distribution from the source and target domain to generate a new feature transformation T.

JDA learns a feature representation that reduces the difference between $P(y_s|x_s)$ and $P(y_t|x_t)$, and $P(\mathcal{X}_s)$ and $P(\mathcal{X}_t)$. This can be done by minimising the following equation:

$$\min_T \|\mathbb{E}_{P(x_s|y_s)}[T(x_s)|y_s] - \mathbb{E}_{P(x_t|y_t)}[T(x_t)|y_t]\| \tag{3.4}$$

where T is the feature transformation and $\mathbb{E}_{P(x_s|y_s)}[T(x_s)|y_s]$ and $\mathbb{E}_{P(x_t|y_t)}[T(x_t)|y_t]$ are the joint expectations of the features x and labels y of the source and target domain respectively. The problem with equation 3.4 is that we do not use the labels of the target domain, thus $P(x_t|y_t)$ cannot be estimated. Instead, JDA adopts the Maximum Mean Discrepancy (MMD) distance measure to reduce the difference between the marginal distributions $P(\mathcal{X}_s)$ and $P(\mathcal{X}_t)$.

Pan et al. [119] have proposed to find such representation through *transfer component analysis* (TCA). TCA finds a representation across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD). MMD measures the similarity between the source and the target domain by computing the distances as follows.

$$D(\mathcal{X}_s, \mathcal{X}_t) = \|\frac{1}{n_s} \sum_{x_i \in \mathcal{X}_s} \phi(x_i) - \frac{1}{n_t} \sum_{x_j \in \mathcal{X}_t} \phi(x_j)\|_{\mathcal{H}}^2, \tag{3.5}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the RKHS and ϕ is a feature map to map the original data points to RKHS.

TCA learns a set of common *transfer components* such that the difference in distributions of data in the source and target domains can be reduced [119]. TCA can be viewed as a special case of JDA. The main difference between the two methods is that JDA adapts the marginal distributions and conditional distributions simultaneously [97].

Boqing et al. [58] have proposed to minimise the distance between the source and target domains with a kernel-based method called *geodesic flow kernel* (GFK) that integrates an infinite number of subspaces to represent the geometric changes and statistical properties from the source to the target domain. GFK constructs an infinite-dimensional feature space $\mathcal{H}^{\infty}$ that contains information about the source domain $\mathcal{D}_S$, the target domain $\mathcal{D}_T$, and the *phantom* domains interpolation between the two domains. That is, given $\mathcal{D}_S$ and $\mathcal{D}_T \in \mathbb{R}^{(D-d)}$, the source and target domains respectively, and $\mathcal{R}_S \in \mathbb{R}^{D\times(D-d)}$ the orthogonal complement to $\mathcal{D}_S$, i.e, $\mathcal{R}_S^T \mathcal{D}_S = 0$. Using the canonical Euclidean metric for the Riemannian manifold, the geodesic flow is parametrized as $\Phi : t \in [0,1] \rightarrow \Phi(t) \in \mathcal{G}(d,D)$ with constraints $\Phi(0) = \mathcal{D}_S$ and $\Phi(1) = \mathcal{D}_T$. For $t \neq 0,1$, $\Phi(t) = \mathcal{D}_S \mathcal{U}_1 \Gamma(t) - \mathcal{R}_S \mathcal{U}_2 \Sigma(t)$, where $\mathcal{U}_1 \in \mathbb{R}^{d\times d}$ and $\mathcal{U}_2 \in \mathbb{R}^{(D-d)\times d}$ are orthonormal matrices:

$$\mathcal{D}_S^T \mathcal{D}_T = \mathcal{U}_1 \Gamma \mathcal{V}^T, \quad \mathcal{R}_S^T \mathcal{D}_T = -\mathcal{U}_2 \Sigma \mathcal{V}^T \tag{3.6}$$

where Γ and Σ are diagonal matrices of size $d \times d$ whose diagonal elements are $\cos\theta_i$, and $\sin\theta_i$. The *overlap* degree between $\mathcal{D}_S$ and $\mathcal{D}_T$ is measured by θ_i [56].

Feature-Level Domain Adaptation (FLDA) [86] fits a probabilistic sample transformation function that models the transfer between the source and target domain. The transfer model is a data-dependent distribution that models the likelihood of the target data conditioned on observed source data. The parameters of the model are estimated by maximising the likelihood of the target data under the transfer distribution conditioned on the source data. The transfer distribution $p_{Z|X}$ describes the relation between the source and the target domain. Given $p_{Z|X}$ and p_X, the marginal distribution over the target domain is

$$q_Z(z|\theta,\eta) = \int_{\mathcal{X}} p_{Z|X}(z|x,\theta) p_X(x|\eta) \mathrm{d}x \tag{3.7}$$

where θ are the parameters of the transfer model, and η the parameters of the source model.

First, the parameters η are learnt by maximising the likelihood of the source domain data under the model $p_{\mathcal{X}}(x|\eta)$. Then, the parameters θ are estimated by maximising the likelihood of the target domain data under the model $q_{\mathcal{Z}}(z|\theta,\eta)$.

Bickel et al. [16] proposed Importance-weighting with logistic discrimination (IW) . Given a labeled training sample $L = \langle (x_1, y_1), ..., (x_m, y_m) \rangle$ governed by an unknown distribution $p(x|\lambda)$. Labels are drawn according to an unknown target concept $p(y|x)$. Let $T = \langle x_{m+1}, ..., x_{m+n} \rangle$ be an unlabeled test set. The test set is governed by a different unknown distribution $p(x|\theta)$. The goal is to find a discriminative model for learning under two different distributions. In other words, the goal is to find a classifier $f : x \mapsto y$ and to predict the missing labels $y_{m+1}, ..., y_{m+n}$ for the test instances. The model should minimise the loss function $\mathbb{E}_{(x,y)\sim\theta}[l(f(x), y)]$ that is defined with respect to the unknown test distribution $p(x|\theta)$. The discriminative model estimates weights for the training instances instead of modelling the distribution over the instances. That is, the contribution of each training instance to the optimisation problem is weighted with a density ratio: for each element x of the training set, selector $\sigma = 1$ indicates that $x \in L$. For each x in the test data, $\sigma = 0$ indicates that $x \in T$. The conditional probability $p(\sigma = 1|x, \theta, \lambda)$ discriminates training $(\sigma = 1)$ against test instances $(\sigma = 0)$. The density ratio can be expressed as follows:

$$\frac{p(x|\theta)}{p(x|\lambda)} = \frac{p(\sigma = 1|\theta,\lambda)}{p(\sigma = 0|\theta,\lambda)} \frac{p(\sigma = 0|\theta,\lambda)}{p(\sigma = 1|\theta,\lambda)} \frac{p(x|\theta)}{p(x|\lambda)} \tag{3.8}$$

The model parameters are calculated with a joint maximum a posteriori (MAP) hypothesis of both the parameters of the density ratio and the final classifier.

3.3.3.2 Deep Learning Models

Deep Adaptation Network (DAN) [96] embeds the hidden representations of the task-specific layers of a CNN in RKHS and explicitly matches the mean embeddings of source and target domain distributions. As mean embedding matching is sensitive to the kernel choices, an optimal multi-kernel selection procedure is performed to reduce the domain discrepancy.

DAN focuses on the multiple kernel variant of MMD (MK-MMD) proposed by Gretton et al. [60]. MK-MMD jointly maximise the two-sample test power and minimise the Type II error, that is, the failure of rejecting a false null hypothesis. The MK-MDD $d_k(p, q)$ between

probability distributions p and q is defined as the RKHS distance between the mean embeddings of p and q.

$$d_k^2(p,q) \triangleq \left|\left|\mathbb{E}_p[\phi(x_s)] - \mathbb{E}_q[\phi(x_t)]\right|\right|^2_{\mathcal{H}_k} \tag{3.9}$$

where $\mathcal{H}_k$ is the reproducing kernel Hilbert space endowed with a characteristic kernel k. An important property is that $p = q$ if $d_k^2(p,q) = 0$ [60]. The characteristic kernel associated with the feature map ϕ, $k(x_s,x_t) = \langle\phi(x_s),\phi(x_t)\rangle$, is defined as the convex combination of m positive-defined kernels $\{k_u\}$,

$$\mathcal{K} \triangleq \left\{k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u\right\} \tag{3.10}$$

where the constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that the derived multi-kernel k is characteristic.

DAN fine-tunes a CNN model on the source labelled samples and introduces MK-MMD-based multi-layer adaptation regulariser to perform layerwise matching so that the source and target domain are as similar as possible under the hidden representations of fully connected layers.

$$\min_{\Theta} \frac{1}{n_a} \sum_{i=1}^{n_a} J(\theta(x_i^a), y_i^a) + \lambda \sum_{l=l_1}^{l_2} d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l) \tag{3.11}$$

where $\lambda > 0$ is a penalty parameter, l_1 and l_2 are layers indices between which the regulariser is effective. $\mathcal{D}_*^l$ is the lth layer hidden representation for the source and target samples and $d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l)$ is the MK-MMD between the source and target evaluated on the lth layer representation.

Joint Adaptation Networks (JAN) [98] extends DAN by aligning the joint distributions of the multiple domain-specific layers based on joint maximum mean discrepancy (JMMD). JMMD measures the Hilbert-Schmidt norm between kernel mean embedding of empirical joint distributions of source and target data.

Denote by $\mathcal{L}$ the domain-specific layers where the activations are not safely transferable. The discrepancy in the joint distributions of the activations in layers $\mathcal{L}$ can be reduced by integrating the JMMD over the domain-specific layers $\mathcal{L}$ into the CNN error

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i^s), y_i^s) + \lambda \hat{\mathcal{D}}_{\mathcal{L}}(P, Q) \tag{3.12}$$

where $\lambda > 0$ is a tradeoff parameter of the JMMD penalty, $J(\cdot, \cdot)$ is the cross-entropy loss function, $\hat{\mathcal{D}}_{\mathcal{L}}$ is the empirical estimate of JMMD and $P(X^s)$ and $Q(X^t)$ are the marginal distributions of the source and target domains, respectively.

The universal RKHS kernel-based MMD may suffer from vanishing gradients for low-bandwidth kernels. To overcome this issue, JAN includes multiple fully-connected layers parametrised by θ to JMMS to make the function class of JMMS richer. In this way, JAN maximises JMMD with respect to θ to approach the virtud of the original MMD, maximising the test power of JMMD such that distributions of the source and target domains are more distinguishable. This leads to

$$\min_f \max_\theta \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i^s), y_i^s) + \lambda \hat{\mathcal{D}}_{\mathcal{L}}(P, Q; \theta) \tag{3.13}$$

The goal of JAN is to reduce the shift in the joint distributions across domains and to learn transferable features such that the target risk can be minimised by jointly minimising the source risk and domain discrepancy.

Domain-Adversarial Neural Network (DANN) [53] is proposed to learn domain-invariant features by combining domain adaptation with feature learning. The distribution alignment between two domains is achieved through standard backpropagation training. The model focuses on learning features that are discriminative for the main learning task on the source domain and domain-invariant with respect to the shift between the domains.

To tackle the challenging domain adaptation tasks, DANN focuses on the $\mathcal{H}$-divergence that relies on the capacity of the hypothesis class $\mathcal{H}$ to distinguish between samples generated by the source domain $\mathcal{D}_s$ from samples generated by the target domain $\mathcal{D}_t$. Ben-David et al. [13] proved that, for a symmetric hypothesis class $\mathcal{H}$, the empirical $\mathcal{H}$-divergence between two samples can be computed as follows

$$\hat{d}_{\mathcal{H}}(S, T) = 2\left(1 - \min_{\eta \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} I[\eta(x_i) = 0] + \frac{1}{m} \sum_{i=n+1}^{N} I[\eta(x_i) = 1]\right]\right) \tag{3.14}$$

where $S \sim (\mathcal{D}_s^x)^n$, $T \sim (\mathcal{D}_t^x)^m$ and $I[a]$ is the indicator function which is 1 if predicate is true, and 0 otherwise.

Ben-David et al. [13] showed that the $\mathcal{H}$-divergence $\hat{d}_{\mathcal{H}}(\mathcal{D}_s^x, \mathcal{D}_t^x)$ is upper bounded by its empirical estimate $\hat{d}_{\mathcal{H}}(S,T)$ plus a constant complexity term that depends on the Vapnik–Chervonenkis (VC) dimension of $\mathcal{H}$ and the size of samples $\mathcal{S}$ and $\mathcal{T}$. Ben-David et al. [13] proof that in order to control $\mathcal{H}$-divergence the feature representation of both source and target domains should be as indistinguishable as possible. Under such representation, a hypothesis with a low source risk will perform well on the target data.

DANN implements this idea to learn a model that can generalise well from one domain to another. It ensures that the internal representation of the neural network contains no discriminative information about the origin of the input (source or target data), while preserving a low risk on the labeled source samples.

Tzeng et al. [157] 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. The goal is to regularise the learning of the source and target mappings to minimise the distance between source and target mapping distributions. If this is the case then the source classifier C_s can be directly applied to the target representation.

$$\min_{M_s,C} \mathcal{L}_{cls}(X_s, Y_s) = -\mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} \sum_{k=1}^{K} I_{[k=y_s]} \log C(M_s(x_s)) \tag{3.15}$$

First, the domain discriminator D classifies whether a data point is drawn from the source or the target domain. This discriminator is optimised using a supervised loss

$$\mathcal{L}_{adv_D}(X_s, X_t, M_s, M_t) = -\mathbb{E}_{x_s\sim X_s}[\log D(M_s(x_s))] - \mathbb{E}_{x_t\sim X_t}[\log(1 - D(M_t(x_t)))] \tag{3.16}$$

where M_s and M_t are the mapping distributions of the source and target domains, respectively.

Second, the source and target mappings are optimised according to a constrained adversarial objective.

$$\begin{aligned} &\min_{D} \mathcal{L}_{adv_D}(X_s, X_t, M_s, M_t), \\ &\min_{M_s, M_t} \mathcal{L}_{adv_M}(X_s, X_t, D) \\ &\text{s.t.} \varphi(M_s, M_t) \end{aligned} \tag{3.17}$$

where $\varphi(M_s, M_t)$ is the mapping optimisation constraints and $\mathcal{L}_{adv_M}$ is the adversarial mapping loss

$$\mathcal{L}_{adv_M}(X_s, X_t, D) = -\mathbb{E}_{x_t \sim X_t}[\log D(M_t(x_t))] \tag{3.18}$$

The model effectively learns an asymmetric mapping by using a pre-trained source model as initialisation for the target representation space and fixes the source model during adversarial training. In this way, ADDA is optimised in stages. First $\mathcal{L}_{cls}$ is optimise over M_s and C by training the classifier using a labelled source data. The source mapping distribution M_s is fixed while learning the target mapping distribution M_t, $\mathcal{L}_{adv_D}$ and $\mathcal{L}_{adv_M}$ can be optimised without revisiting 3.15.

Tang et al. [154] proposed Discriminative Adversarial Domain Adaptation (DADA) which reduces domain discrepancy by generating a mutually inhibitory relation between its domain prediction and category prediction for any input instance. The adversarial training conducts competition between the domain neuron and the true category neuron. DADA enables explicit alignment between the joint distributions, thus improving target data classification.

Given $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ of labeled instances from the source domain $\mathcal{D}_s$ and $\{x_j^t\}_{j=1}^{n_t}$ of unlabeled samples from the target domain $\mathcal{D}_t$, the objective of unsupervised domain adaptation is to learn a feature extractor $G(\cdot)$ and a task classifier $C(\cdot)$ such that the expected target risk $\mathbb{E}_{(x^t, y^t) \sim \mathcal{D}_t}[\mathcal{L}_{cls}(C(G(x^t)), y^t)]$ is low for a certain classification loss function $\mathcal{L}_{cls}(\cdot)$.

The Correlation Alignment (CORAL) proposed by Sun et al. [149] uses asymmetric transformations to match the mean and covariance of the two distributions. CORAL is an unsupervised domain adaptation method that aligns the second-order statistics of the source and target distributions with a linear transformation. The original CORAL model relies on transforming the extracted features and then training an SVM classifier in the next step.

The CORAL model minimises the distance between the covariance of the source and target domains by applying a linear transformation LT to the original source features and using the Frobenius norm as the matrix distance metric:

$$\min_{LT} = ||C_{\hat{S}} - C_T||_F^2 = \min_{LT} ||LT^T C_S LT - C_T||_F^2 \tag{3.19}$$

where $C_{\hat{S}}$ is the covariance of the transformed source features, and C_S and C_F are the covariance matrices of the source and target domain, respectively.

Sun et al. [150] extended the model to incorporate it into deep networks by constructing a differentiable loss function that minimises the source and target correlation difference. This new loss function is the CORAL loss. In this sense, the CORAL loss is defined as the distance between the second-order statistics or covariances of the source and target features:

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2} ||C_s - C_T||_F^2 \tag{3.20}$$

where $||\cdot||_F^2$ denotes the squared matrix Frobenius norm. The covariance matrices of the source and target data are given by:

$$C_S = \frac{1}{n_S - 1}(D_S^T D_S - \frac{1}{n_S}(1^T D_S)^T (1^T D_S)) \tag{3.21}$$

$$C_T = \frac{1}{n_T - 1}(D_T^T D_T - \frac{1}{n_T}(1^T D_T)^T (1^T D_T)) \tag{3.22}$$

where D_S and D_T are the training samples of the source and target domains respectively, and n_S and n_T are the sample size of the source and target data respectively. The gradient with respect to the input features is calculated using the chain rule:

$$\frac{\partial \mathcal{L}_{CORAL}}{\partial D_S^{ij}} = \frac{1}{d^2(n_S - 1)}((D_S^T - \frac{1}{n_S}(1^T D_S)^T 1^T)^T (C_S - C_T))^{ij} \tag{3.23}$$

CORAL later is extended in a deep neural network, called *DeepCORAL*, to learn a non-linear transformation that aligns correlations of layer activation between the source and target networks [151].

Saito et al. [138] employ a task-specific classifier as a discriminator to consider the relationship between target samples and class decision boundaries when aligning distributions. Zhao

et al. [198] have proposed multi-source distillation domain adaptation that first adversarially maps the target domain into each source domain and selects the source training samples that are close to the target domain to fine-tune the source classifier. Then the improved source classifiers will classify the mapped target samples, and the prediction results will be aggregated for a final prediction.

Chen et al. [28] have designed a Re-weighted Adversarial Adaptation Network (RAAN) for unsupervised domain adaptation to reduce disparate domain discrepancies and adapt the classifier. First, they train a domain discriminator network together with a deep convolutional neural network in an adversarial manner to minimise the optimal transformation based on EM distance. The label distribution is matched by estimating a re-weighted source domain label distribution to adapt the classifier.

Adversarial Domain Adaptation with Domain Mixup (ADADM) [179] advances adversarial learning by mixing transformed source and real target domain samples to train a more robust generator. This is done by using a variant of VAE-GAN propused by Larsen et al. [92]. In the same way as convention variational autoencoder, an encoder N_e maps inputs from source and target domains to the standard Gaussian distribution $\mathcal{N}(0,I)$. For every sample, a mean vector μ and a standard deviation vector σ are used as the feature embedding. At feature level, the feature embeddings of source and target domains are linearly mixed to produce mixup features

$$\begin{aligned} x^m &= \lambda x_s + (1-\lambda)x_t, \\ l^m &= \lambda l_s + (1-\lambda)l_t = \lambda \end{aligned} \tag{3.24}$$

where x^m are mixup samples with corresponding soft domain labels l^m, $\lambda \in [0,1]$ is the mixup ratio, and λ follows a Beta distribution.

The source and target samples are embedded to (μ_s, σ_s) and (μ_t, σ_t) in the latent space by a shared encoder N_e. The two domains' embeddings are then linearly mixed to produce mixup feature embedding (μ_m, σ_m)

$$\begin{aligned} \mu^m &= \lambda\mu_s + (1-\lambda)\mu_t, \\ \sigma^m &= \lambda\sigma_s + (1-\lambda)\sigma_t \end{aligned} \tag{3.25}$$

The embedding of the source domain is used to do K-way object classification by the classifier C and the source and target domain are aligned on category level through enforcing the decoded samples to be as similar as possible to source samples and preserve class information inputs.

In recent years, the research in GAN has well advanced and several coupled GAN architectures have been proposed in domain adaptation and image-to-image translation [189, 204]. For example, DupGAN [70] learns domain-invariant representation via an encoder, a generator, and two discriminators. The encoder aims to encode samples from both domains into a latent space, a conditional generator decodes latent representations back into source and target domains conditioned on the domain code, and discriminators on each domain to tell whether a sample is from the specific domain or generated. However this approach assumes both source and target domain shares the same feature space, due to the design on the encoder. Bi-directional GAN [189, 204], originated in image-to-image translation, unpairs two GANs to enforce cycle (or bi-directional) consistency between source and target domains, making sure each image can be recovered through two generators' operation. This approach has achieved promising performance and does not assume the same feature space between source and target domains. Therefore, we will base our approach on this architecture.

The above techniques are used for domain adaptation in general. However, our interest relies on accelerometer and binary sensor data. Therefore, in the following section, we present a specific literature review related to these topics.

3.3.4 Domain Adaptation on Accelerometer Data

There have been quite a few attempts of transfer learning on accelerometer data; e.g., from one user to another [199], from one body position (e.g., chest) to another (e.g., hips) [169], and from one device to another [81]. However, as accelerometer data share the same dimensions, i.e., timestamp and x-, y-, and z-dimension, generated feature spaces can be uniform as long as they

use the same feature extraction technique. Thus, the focus is to align the distributions rather than transfer feature spaces.

Qin et al. [126] propose Adaptive Spatial-Temporal Transfer Learning (ASTTL) to allow more accurate source selection to perform domain adaption. Chang et al. [22] have looked into feature matching and adversarial learning in adapting the activity model from one sensor position to another. These recent techniques are built on a similar assumption that both source and target domains share the same feature space; therefore, they can share the same activity model [169] or feature extractor [126, 22].

Zhao et al. [199] propose a TransEMDT system to transfer accelerometer-based activity recognition models between different users. The idea is to train a decision tree on one user and then predict activity labels on another user's accelerometer data. A k-means clustering algorithm is applied to the classification results. Then the original decision tree model will be updated by iteratively resampling the most confident data on the new user. Similarly, Khan and Roy propose an instance-based transfer boost algorithm with k-means clustering to transfer activity models between smart phones and smart watch [81].

Maekawa et al. [99] have proposed an unsupervised approach to recognise physical activities from accelerometer data. They utilise information about users' characteristics such as height and gender to compute the similarity between users, and find and adapt the models for the new users from the similar users.

Wang et al. [169] have proposed a Stratified Transfer Learning (STL) model to recognise physical activities from different users. They first train classifiers on the annotated source domain dataset and use the classifiers to generate pseudo activity labels on the target domain dataset. Then they perform intra-class knowledge transfer; that is, map the sensor data of both source and target domain on the same activity label and use various types of transfer kernels to project both domains' feature spaces to a common subspace. Then they will re-train classifiers on the common subspaces to re-label the target domain dataset. This approach has produced promising results when there is no labelled data in the target domain.

Generative adversarial models have been employed in the image-to-image translation task [190, 203]; for example, generating a sketch from a real image. The main idea is to use two GANs, where one is to generate target images on the input of the source images and the other is to

generate source images on the input of the target images. The loss function is a combination of both GANs. For example, CycleGAN [203] learns a generator which produces images in one domain given images from the other domain and a model is trained with a cycle-consistency constraint which enforces a strong connection across domains by mapping an image from the source domain to the target domain and then back to the source domain which will result in the same starting image. However, one limitation of CycleGAN is that it only learns one-to-one mapping. Almahairi et al. [6], extended the CycleGAN model and introduced a model called Agumented CycleGAN which learns many-to-many mappings between domains. The key idea is to use auxiliary variables separately from the input image to capture the variations independent from the content to be translated. Liou et al. [95] introduced a tuple of GANs (CoGAN) to learn a joint distribution of multi-domain images. It consists of a pair of GANs; each is responsible for synthesizing images in one domain. During the training process, both GANs share a subset of parameters. The generators share their high-layer weights and the discriminators share their low-layer weights. In this way, the GANs learn to synthesize pairs of corresponding images without correspondence supervision.

Karras et al. [78] proposed StyleGAN which re-designs the generator architecture of style transformer network to control the image synthesis process. The generator starts from constant learned input and adjusts the style of the image at each convolution layer based on the latent code. The training efficiency and outcomes achieved are better than pairwise transformations thanks to the power of joint leaning.

Choi et al. [29] proposed a GAN-based method called StarGAN to perform image-to-image translations for multiple domains. Their method allows simultaneous training of multiple datasets with different domains using only a single generator and a discriminator. StarGAN incorporates multiple datasets containing different types of labels and uses a mask vector that allows to ignore unspecified labels and focus on known labels provided by particular datasets. Besides its good performance, StarGAN generates images of higher visual quality compared to existing methods.

Suzuki et al. [153] extended StarGAN method to use multi-channel sensory data and introduced a generative adversarial network based style transformer to produce a user's gesture data. First gesture data is transformed into another gesture data (intra-user transformation) or one user's gesture data is transformed to another user's data (inter-user transformation), and then the

output is used to train a personal classifier. Their method enables users to reduce the effort in collecting personal training data.

3.3.5 Domain Adaptation on Binary Event Sensor Data

Transfer learning on binary event sensor data is different from accelerometer data. Features generated on accelerometer data are in the same feature space and transfer learning focuses on transferring the distributions of features between different subjects. However, in binary sensor data, sensor features can be drastically different. Furthermore, each environment can have a different sensor deployment in terms of the number and the locations of sensors being placed. This heterogeneity in feature spaces brings an extra challenge on transfer learning of activity models. It often requires an intermediate mechanism to bridge the feature spaces in the source and target domain.

Rosales et al. [135] proposed a 2-staged domain adaptation where semantics similarity is employed to perform linear transformation of sensor features from one domain to another domain and then a variational autoencoder (VAE) is used for fine alignment between transferred features and source features. Other than semantics, Feuz et al. [49] map feature spaces via meta-features on each sensor; that is, the time a sensor is activated, and intervals and sequence between sensor activation. These approaches have achieved promising results in resolving heterogeneity between feature spaces but they require extra effort to craft the knowledge [161, 184, 132, 135] and learn meta-mapping [49]. Also the effectiveness of these approaches is significantly subject to the reliability of such knowledge [135].

Zheng et al. [201] propose an algorithm for cross-domain activity recognition that transfers the labelled data from a source domain to a target domain. The activity model in the source domain can help complete the similar activity model in the target domain. The similarity is measured not only on the objects involved in the activities but also on their underlying physical actions. One example in [201] is that the activity ‘Washing-laundry’ is similar to ‘Hand-washing dishes’ on the action of ‘Hand washing’. They use the web search and apply the information retrieval techniques to build the similarity function that produces different probabilistic weights of actions and objects on activities of interest. These weights will be further used to train a multi-class weighted support vector machine to support activity recognition.

van Kasteren et al. [161] propose a manual mapping between sensors in different households and learn the parameters of a target model using the EM algorithm to transit probabilities of HMM models from source to target. Similarly, Rashidi et al. [132] learn sensor mappings based on their location and activity models' roles. The role is characterised by mutual information, measuring the dependence between an activity and a sensor, and suggests the sensor's relevance in predicting the corresponding activity. Feuz et al. [49] propose a data-driven approach to automatically map sensors based on their meta-features, which are mainly about when a sensor reports, and time intervals between events reported by this sensor and other sensors.

Ye et al. [184] propose *shared learning* on scarcely and partially annotated data from multiple users to achieve satisfactory activity recognition accuracies. The hypothesis is that as long as each user contributes a very small number of labelled examples (even though these examples might not cover a complete set of activity types), a shared learning approach will learn annotated examples across all the users and complement each other to build an activity recognition model to cover all the activities. This approach has the potential of reducing the annotation burden on each user and has demonstrated its effectiveness when each user contributes to a very small number of annotated activities. However, the performance of this approach still needs a significant improvement.

Adaptive Spatial-Temporal Transfer Learning (ASTTL) [126] is proposed to transfer activity knowledge and select appropriate source domain for cross-dataset HAR problem. It extends GFK with the Markov property to learn temporally adaptive features in the manifold space. Convolutional deep Domain Adaptation model for Time Series data (CoDATS) [174] also tackles the source domain selection problem and it is built on the domain adversarial neural network (DANN) [53]. Chang et al. [22] have developed unsupervised domain adaptation algorithms on feature matching and confusion maximisation and performed in-depth analysis of these algorithms in wearing diversity.

3.4 Challenges of Domain Adaptation in HAR

Although transfer learning techniques have progressed in the last few years, there are still many challenges. First, researchers have not applied transfer learning yet when the source data is not

labelled. Current approaches use labelled source data to improve transfer performance in the target domain. However, transfer-based activity recognition when the source data is not labelled has received little attention. Similarly, transferring across different label spaces is a much less studied problem in transfer-based activity recognition [30].

Work to date in transfer learning falls mainly in transferring knowledge learned in the source domain to the target domain over the same variables. What remains is to generalise between domains where the type of objects and variables are different [39]. This is called *relational-knowledge*, which requires a particular relationship in the data that can be learned and transferred across populations [30]. That is, knowledge from one domain is applied in another by establishing a correspondence between the objects and relations in them.

More work needs to be done to improve transfer across sensor modalities and knowledge across multiple environments. For example, instead of transferring knowledge from one environment to another, can we transfer from one environment to a completely different one? For example, can we train a model in a smart home and transfer the knowledge learned to to another smart home with a different setup?

Finally, a major challenge in current activity recognition research is to collect sufficient labelled data in the environment to train classification models. Transfer learning has been proposed to deal with this problem, however, the challenge still remains.

Chapter 4

Knowledge-driven Unsupervised Domain Adaptation

4.1 Overview

Sensor-based human activity recognition recognises daily human activities through a collection of ambient and wearable sensors. It is the key enabler for many healthcare applications, especially in ambient assisted living. In addition, the advance of sensing and communication technologies has driven the deployment of sensors in many residential and care home settings. However, the challenge still resides in the lack of sufficient, high-quality activity annotations on sensor data, which most existing activity recognition algorithms rely on.

In this chapter, we present our first contribution in the thesis; we propose an unsupervised domain adaptation technique for activity recognition, called *UDAR*, which supports sharing and transferring activity models from one dataset to another heterogeneous dataset without the need for activity labels on the latter. This approach has combined knowledge- and data-driven techniques to achieve coarse- and fine-grained feature alignment. We have evaluated UDAR on third-party, real-world datasets and it has demonstrated high recognition accuracy and robustness against sensor noise, compared to the state-of-the-art domain adaptation techniques. Most of the material is extracted from our publication [140].

4.2 Introduction

Recognising everyday routine activities can be challenging, as it involves understanding human behaviour from a series of observations derived from motion, location, physiological signals and environmental information. Most of the existing approaches [91, 168] assume that the sensor data distribution is the same as that used in the model training process. However, this assumption is not always valid. A major challenge in current activity recognition research is to collect sufficient labelled data in the environment to train classification models. This task can be expensive, and the lack of training labelled samples can compromise the classifier's performance. Transfer learning is proposed to apply knowledge learned from the source domain to the target domain to deal with this problem.

We hypothesise that we can accurately recognise one user's (referred to as the *target user*) activities by performing unsupervised adaptation of activity models from another user (referred to as the *source user*). Instead of collecting activity labels on the target user, we can transfer the knowledge on the source user and automatically predict activity labels on the target users. The main challenge resides in the *mapping between heterogeneous feature spaces*. Both users can live in different environments with different spatial layouts and are deployed with different numbers of sensors or different types of sensing technologies. In transfer learning, this problem is regarded as *unsupervised domain adaptation* [52].

In this chapter, we explore a research question: *is it possible to relieve the annotation burden on individual users but still be able to build a robust activity recognition model by sharing and transferring activity models across users, even though the sensor deployments and operating environments are different*? We hypothesis that our method UDAR, which supports sharing and transferring activity models from one dataset to another heterogeneous dataset without the need for activity labels on the latter dataset, can address this question.

The main contributions and novelty of our method are listed as follows.

- We have designed a workflow that combines knowledge- and data-driven techniques in performing domain adaptation at different stages. We build on a general ontology for smart home datasets and achieve coarse-grained feature space remapping to link heterogeneous datasets without the need for labelled data in the target domain. We apply Variational

Autoencoder (VAE) to perform fine-grained feature space alignment. VAE has achieved promising results in learning effective latent feature representations in computer vision [84] and also in minimising the distance of the source and target feature spaces based on their latent feature representations [3].

- We have performed an extensive empirical evaluation on third-party, real-world datasets that have different spatial layouts and sensor deployments. We have designed different experiments on assessing the effectiveness and robustness of domain adaptation with different training data percentages and sensor noise settings. The results have demonstrated the robustness of UDAR as it has consistently outperformed the state-of-the-art domain adaptation techniques. These results are presented in chapter 7.

Section 4.3, describes the problem – *unsupervised domain adaptation* and presents our approach in a workflow. Section 4.4, introduceS the pre-annotation process with coarse-grained knowledge-driven feature space remapping and section 4.5, describes fine-grained VAE based feature alignment.

4.3 Problem Statement and Overview

In this section, we define the problem of unsupervised domain adaptation, illustrate it in a concrete example, and present the workflow of our approach UDAR.

Definition 2. Assume that a source and target domain dataset is defined as follows.

- A *source* domain dataset consists of a collection of labelled instances, $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where an instance x_i^s ($\in \mathcal{X}^s$) is labelled with a class label y_i^s ($\in \{1, 2, ..., C\}$). Here $\mathcal{X}^s$ is a M_s-dimensional feature space.

- A *target* domain dataset consists of a collection of unlabelled instances $\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}$, where $x_j^t \in \mathcal{X}^t$. Here $\mathcal{X}^t$ is a M_t-dimensional feature space.

Both source and target domains have different feature spaces but share the same label space; *i.e.*, $\mathcal{X}^s \neq \mathcal{X}^t$ such that they have different dimensions $M_s \neq M_t$, and their marginal distributions and conditional distributions are different; *i.e.*, $P(\mathcal{X}^s) \neq P(\mathcal{X}^t)$ and $P(y^s|x^s) \neq P(y^t|x^t)$. *Unsupervised*

domain adaptation is to predict a label y_j^t for each instance x_j^t in the target domain dataset and $y_j^t \in \{1, 2, ..., C\}$.

Figure 4.1 illustrates the above problem. Two houses A and B are presented, each of which is deployed with a number of binary event-driven sensors [162]. For example, House A is configured with infrared passive motion sensors, which report 1 when the presence of an object or a user is detected. House B is configured with RFID to monitor the presence of an object and switch sensors to monitor the 'open' and 'close' states of a cupboard or a door. Our task is to transfer an activity model from one house (*e.g.*, A) to predict labels in the other house (*e.g.*, B), without using any activity labels on house B.

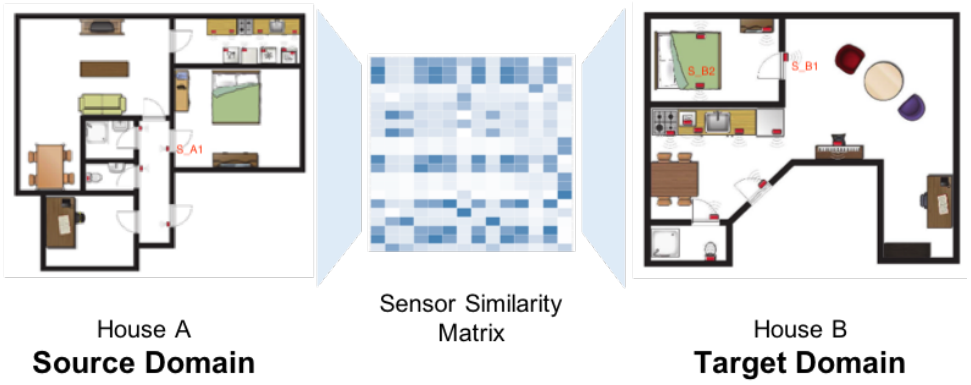


Figure 4.1: The representation of sensor deployments in two different smart homes: House A and House B [162]. A sensor similarity matrix is used to initialise the similarity of sensor features between both houses.

4.3.1 Overview

A general strategy to address the unsupervised domain adaptation problem is to align feature spaces from both domains and then find a common subspace where both feature spaces can be projected onto and minimise their distance [120]. Aligning feature spaces can be done through matching distributions [120], but this is infeasible as the dimensionality of feature spaces is completely different here. Another way for alignment is based on class labels; that is, align instances from both domains when they share the same class label [169]. Since we do not use the target domain's labels, we need to find a way to generate pseudo labels on the instances in the target domain. In the following, we list the main steps in UDAR.

Step 1 - Knowledge-driven feature remapping between source and target domains, where we use simple semantics to transfer feature space from the target domain to the source domain;

Step 2 - Pre-annotating on the target domain, where we train a classifier on the source domain dataset and generate pseudo labels on the semantics-transferred target domain dataset;

Step 3 - Performing domain adaptation, where we align feature spaces in both source and target domains based on the generated pseudo labels;

Step 4 - Re-annotating on the target domain, where we train a classifier on the transferred target feature space along with their pseudo labels and predict labels on the target dataset.

4.4 Knowledge-driven Feature Remapping and Pre-annotating

This section will describe how we generate pseudo labels on the target dataset to prepare for domain adaptation. Knowledge-driven feature remapping is to map sensor features based on the sensor semantics, where they are deployed and which objects they are attached to. This feature remapping has demonstrated promising results in transferring learning between heterogeneous smart home environments [184], but semantics can be coarse-grained as they ignore any feature distribution on activities. Therefore, they often cannot lead to accurate and fine-grained feature space mapping. This work will only use knowledge-driven feature remapping to generate pseudo labels and then perform more sophisticated domain adaptation later.

4.4.1 Feature Remapping

Ye et al. [186] have presented a general ontology to project sensors in different smart home environments onto the same location and object ontologies. The location ontologies represent the spatial containment relationship between location concepts; *e.g.*, `Bedroom` $\sqsubseteq$ `SleepingArea`. The object ontologies are extracted from WordNet [103] and represent the semantic relations between lexical concepts; *e.g.*, `Door` $\sqsubseteq$ `MovableBarrier`. Through the conceptual hierarchy

of the ontologies, we can calculate semantic similarity between a pair of sensors based on the similarities between their location and object concepts; that is,

$$sim(s_{s,i}, s_{t,j}) = \omega_L \times sim(l_i, l_j) + \omega_O \times sim(o_i, o_j), \tag{4.1}$$

$$\omega_O + \omega_L = 1 \tag{4.2}$$

where ω_L and ω_O are the weights on location and object concepts contributing to the similarity of sensors, $s_{s,i}$ and $s_{t,j}$ are ith and jth sensor in a dataset s and t respectively, and l_i, l_j, o_i, o_j are the location and object concepts in the general ontologies that the sensors i and j are mapped to. The similarity measure between domain concepts is based on their hierarchy [178], which has been detailed in [186]. The weights are set as 0.5 for both object and location because we consider both of their contributions to activity recognition are equally important.

For example, in Figure 4.1, consider the sensor node, marked as *S_A*1, is attached to the bedroom door in House A, and *S_B*1 and *S_B*2 to the bedroom door and bed in House B. When projecting all these sensors onto the same location and object ontologies: `Bedroom` – location concepts for these three sensors, `Door` – object concepts for *S_A*1 and *S_B*1, and `Bed` – object concepts for *S_B*2. Using the above formula 4.1, we can calculate the similarities between *S_A*1 and both *S_B*1 and *S_B*2, which are 1.0 and 0.8 respectively. In this way, we can produce a similarity matrix between each pair of sensors from the source and target domain. A more detailed description can be found in [183]. There might exist some sensors in the target domain that cannot find strong matches in the source domain; *i.e.*, a sensor's similarity scores with all the sensors in the source domain are low. We will leave the feature alignment and the re-annotation process to learn the correlation of these sensors and activity labels.

4.4.2 Pre-annotation

The pre-annotation step is to generate pseudo activity labels on the unlabelled target dataset, using the classifier trained on the source domain dataset. We aim to predict the labels as accurately as possible, as we will use the labels to align feature spaces in the source and target domain datasets. To enhance the accuracies of label generation, we design a stacked ensemble on the source domain dataset, which is presented in Figures 4.2 and 4.3 .

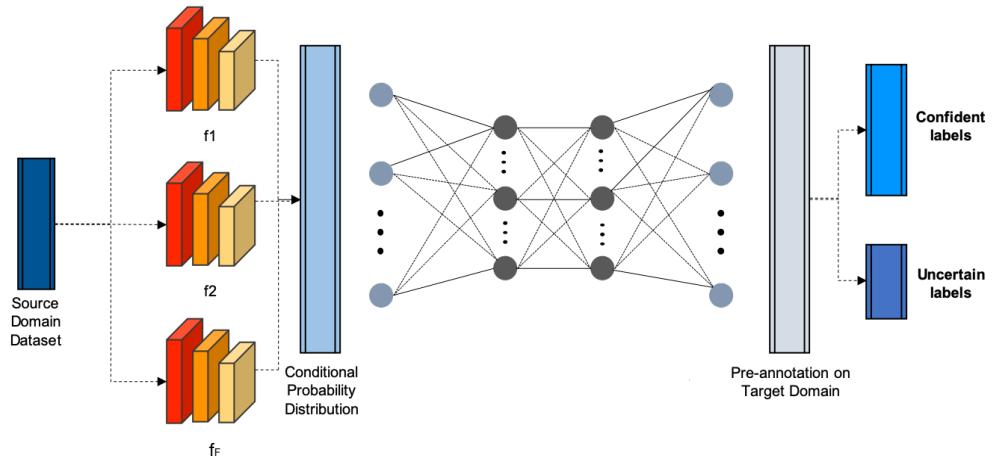


Figure 4.2: The stacked ensemble to predict activity labels on the unlabelled target domain dataset

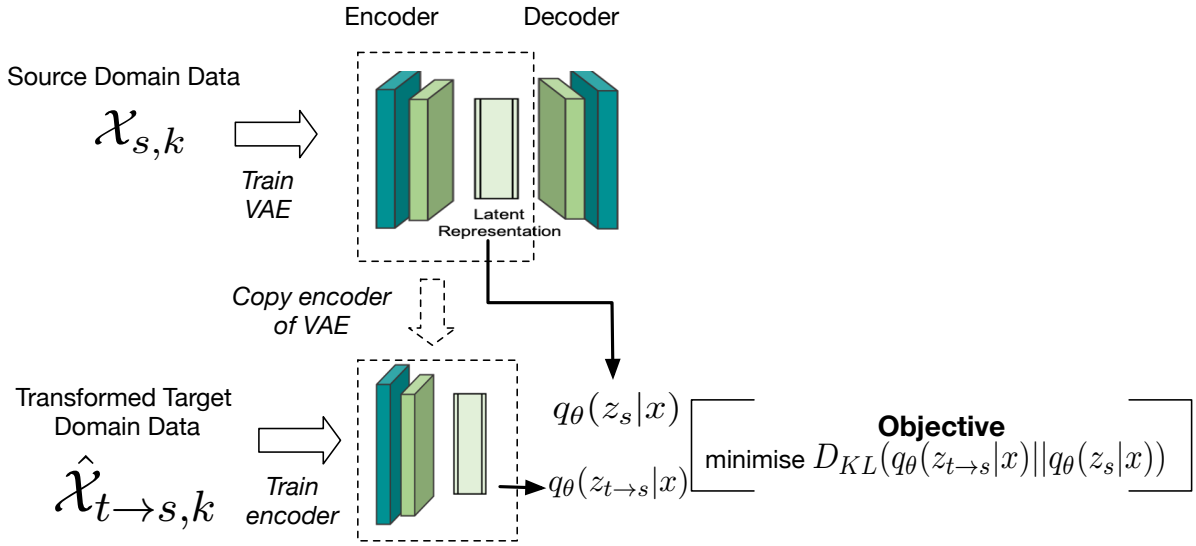


Figure 4.3: Fine-grained feature alignment with VAE

First we train a number of independent classifiers on the source domain dataset, and use them to produce probability distributions on each source instance; that is, $P_{f_i} = [p_{f,1}, p_{f,2}, ..., p_{f,N}]$ represents the probability distribution from a classifier f_i on each class, given that there exists a set of classes $\{1,2,...,N\}$ and a collection of classifiers $\{f_1, f_2, ..., f_F\}$. Then we concatenate these probability distributions together $[P_{f_1}, P_{f_2}, ..., P_{f_F}]$ and build a neural network on top of them to learn the correlations of each base classifier's probability distributions and activity labels.

We map the target domain dataset onto the source domain; *i.e.*, $\hat{\mathcal{X}}^{t\to s} = \mathcal{X}^t S_{t\times s}$, where $S_{t\times s}$ is the sensor similarity matrix from the target to source domain. Then the trained stacked ensemble f is applied to predict labels on $\hat{\mathcal{X}}^{t\to s}$; *i.e.*, $(y_j, p_j) = f(\hat{x}_j^{t\to s})$, indicating that the ensemble f predict a class label y_j on a jth instance in $\hat{\mathcal{X}}^{t\to s}$ with a posterior probability p_j. We collect a

collection of confident instances in the target domain whose posterior probability is higher than a pre-defined threshold τ; *i.e.*, $\{(\hat{x}_j^{t\to s}, y_j) | (y_j, p_j) = f(\hat{x}_j^{t\to s}), p_j \geq \tau\}$. We assume that a high confidence score indicates that most classifiers 'agree' on the same result. A low confidence value is an indication of uncertainty. We leave all the uncertain instances unlabelled for now.

4.5 Domain Adaptation and Re-annotating

Once we generate pseudo activity labels on the target dataset, we align feature spaces from both source and target domain based on their activity labels and perform in-class transfer. We align the instances in the source and target datasets if they share the same label, and learn the affinity between feature spaces for each label. For example, the activity 'eating' in House A and House B is the same for both domains even though it has different distributions and we assume it should lay on the same intrinsic subspace. Here we introduce how to use a Variational AutoEncoder (VAE) for in-class transfer to learn the latent representations that reveal meaningful relationships between the source and target domain.

4.5.1 Domain Adaptation

Domain adaptation is used to match the feature distributions of the source and target domains. This can be achieved by projecting the feature spaces in the source and target domain onto the same subspace so as to minimise their distances. Here we perform in-class domain adaptation. For each class label k ($\in \{1, 2, ..., N\}$), we collect its instances in the source domain; *i.e.*, $\{(x_i^s, y_i^s) | y_i^s = k\}$, and its confident instances in the transformed target domain from the pre-annotation process; *i.e.*, $\{(x_j^{t\to s}, y_j^t) | y_j^t = k\}$ and y_j^t is a label predicted on the trained stack ensemble f. We denote the above instances from the source and target domain on the same class label k as $\mathcal{X}^{s,k}$ and $\hat{\mathcal{X}}^{t\to s,k}$ respectively. The task of domain adaption is to align these two feature spaces.

4.5.1.1 Variational AutoEncoders

Variational AutoEncoders (VAEs) are a variational inference approach for an autoencoder based latent factor model [84]. A VAE is a generative model that draws sample x using latent variable z;

$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$, where $p_\theta(z)$ is the prior distribution on latent variable z, $p_\theta(x|z)$ is the conditional distribution of generating x given z, and θ is the model parameter. $p_\theta(x)$ is intractable because the likelihood function $p_\theta(x|z)$ is complex, which often is modelled as a neural network with a nonlinear hidden layer [84]. To tackle this problem, VAE introduces an encoder network $q_\phi(z|x)$ to approximate the intractable true posterior $p_\theta(z|x)$. That is, a VAE consists of two networks: an *encoder* $q_\phi(z|x)$ that produces the distribution over the latent representation of the variable z given the input data x and a *decoder* $p_\theta(x|z)$ that produces the distribution over x given a latent representation of the variable z.

The marginal likelihood of individual data points $x^{(i)}$ then can be rewritten as

$$\log(p_\theta(x^{(i)})) = D_{\mathrm{KL}}[q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})] + \mathcal{L}(\theta,\phi;x^{(i)}). \tag{4.3}$$

The second term $\mathcal{L}(\theta,\phi;x^{(i)})$ is called *evidence lower bound* (ELBO) on the marginal likelihood of the data point $x^{(i)}$, which can be written as

$$\mathcal{L}(\theta,\phi;x^{(i)}) = \mathbb{E}_{q_\phi(z|x)}[\log(p_\theta(x^{(i)}|z))] - D_{\mathrm{KL}}[q_\phi(z|x^{(i)})||p_\theta(z)], \tag{4.4}$$

where the decoder $p_\theta(x|z)$ and the encoder $q_\theta(z|x)$ are parameterised as the neural networks. The choice of $q_\theta(z|x)$ is often a factorised Gaussian distribution. The first term of the right hand side of the equation is the expected value of the data likelihood, while the KL divergence is a regulariser for the encoder to align the approximate posterior with the prior distribution of the latent variables.

The overall model is trained by stochastically optimising the ELBO using the reparameterisation trick to make the network differentiable [84]. The reparameterisation trick works as follow. If $x \sim N(\mu,\Sigma)$, we can standardise it as x_{std}; i.e $\mu = 0$ and $\Sigma = 1$, and revert it to the original distribution by reverting the standardisation process using $x = \mu + \Sigma^{1/2}x_{std}$.

Having that in mind, we can convert a standard normal distribution into a Gaussian; that is,

$$z = \mu(\mathcal{X}) + \Sigma^{1/2}(\mathcal{X})\varepsilon, \tag{4.5}$$

where $\varepsilon \sim N(0,1)$. In this way, the backpropagation does not depend on z. Finally, the weights

and parameters are updated according to the loss function optimisation.

$$\mathbb{E}[g^2]_t = \beta\mathbb{E}[g^2]_{t-1} + (1-\beta)\frac{\partial C}{\partial \mathcal{W}}^2, \tag{4.6}$$

$$\mathcal{W}_t = \mathcal{W}_{t-1} - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t}}\frac{\partial C}{\partial \mathcal{W}}, \tag{4.7}$$

where $\mathbb{E}[g]$ is the moving average of square gradients, $\frac{\partial C}{\partial \mathcal{W}}$ is the gradient of the cost function with respect to the weight, η is the learning rate, and β the moving average parameter.

4.5.1.2 VAE-based Domain Adaptation

We use VAE to align semantics-based remapped feature spaces in the target domain with the feature space in the source domain to adjust data distributions in order to achieve fine-grained feature alignment. The proposed training framework is presented in Figure 4.3.

We first train the VAE on the source data to obtain the source domain latent representations; that is, given the training data $\mathcal{X}^{s,k}$ in the source domain on an activity class k, and $z_s \sim q_\theta(z_s)$ the latent representation, the posterior distribution $q_\theta(z_s|x)$ is modelled as a multivariate Gaussian distribution with the estimated mean $\mu(\mathcal{X}^{s,k})$ and covariance $\Sigma(\mathcal{X}^{s,k})$; i.e, $q_\theta(z_s) = \mathcal{N}(z_s; \mu(\hat{\mathcal{X}}^{s,k}), \Sigma(\hat{\mathcal{X}}^{s,k}))$.

Second, we transform the target data on the same class k using the sensor similarity matrix; that is, $\hat{\mathcal{X}}^{t\to s,k} = \mathcal{X}^{t,k}\mathcal{S}_{t\times s}$. Then we obtain the posterior distribution $q_\theta(z_{t\to s}|x)$ of the latent representations $z_{t\to s}$ on the transformed target data. We use a neural network that has the same architecture and weights of the encoder in the previous VAE so that we can learn the domain adaptive features by mapping the target domain data into the feature distribution of the source domain. We will then retrain the network with the transformed target data $\hat{\mathcal{X}}^{t\to s,k}$. The training objective is to minimise the KL divergence between the posterior distribution of the latent representations $q_\theta(z_s|x)$ and $q_\theta(z_{t\to s}|x)$:

$$D_{KL}(q_\theta(z_{t\to s}|x)||q_\theta(z_s|x)) = \frac{1}{2}(tr(\Sigma_s^{-1}\Sigma_{t\to s}) + (\mu_s - \mu_{t\to s})^T\Sigma_s^{-1}(\mu_s - \mu_{t\to s}) - l + ln\frac{|\Sigma_s|}{|\Sigma_{t\to s}|}), \tag{4.8}$$

where $tr(\Sigma_s^{-1}\Sigma_{t\to s})$ is the trace function to compute the sum of diagonal of $\Sigma_s^{-1}\Sigma_{t\to s}$, and l is the dimension of the latent representation. This process aligns the latent probability distribution function of the transformed target data to that of the source data by matching their means and the eigenvalues of their covariance.

4.5.2 Re-annotation

Once we have aligned the source and target feature spaces, we will go back to re-annotate uncertain instances remaining in the target dataset. To achieve this, we train a classifier $f_{s\to l}$ on the encoded source domain $\hat{\mathcal{X}}^{s\to l}$; *i.e.*, $\hat{\mathcal{X}}^{s\to l} = vae.encode(\mathcal{X}^s)$, where l is the latent space learnt by a VAE. We use this classifier to predict labels on the encoded target domain $\hat{\mathcal{X}}^{t\to l}$; *i.e.*, $\hat{\mathcal{X}}^{t\to s\to l} = vae.encode(\hat{\mathcal{X}}^{t\to s})$. We assume that the newly predicted labels on the target domain are more reliable than the labels predicted at the pre-annotation step as now the source and target domains are mapped to the same latent subspace. Then we train a new classifier f_t with confident instances from the target domain; *i.e.*, $\{(x_j^t, y_j^t)|(y_j^t, p_j) = f_{s\to l}(x_j^{t\to s\to l}), p_j \geq \tau\}$, where τ is the confidence threshold. Then we predict labels for all the remaining unlabelled instances in the target domain. This process is illustrated in Algorithm 1.

Algorithm 1 Re-annotation

Require: a trained VAE *vae*, labelled source domain data $\mathcal{X}^s$, and unlabelled target domain data $\mathcal{X}^t$
1: map $\mathcal{X}^s$ onto the latent space l of *vae*: $\hat{\mathcal{X}}^{s\to l} = vae.encode(\mathcal{X}^s)$
2: train a classifier $f_{s\to l}$ on $\hat{\mathcal{X}}^{s\to l}$
3: map $\mathcal{X}^t$ onto the latent space l of *vae*: $\hat{\mathcal{X}}^{t\to s\to l} = vae.encode(\mathcal{X}^t S_{t\times s})$
4: use $f_{s\to l}$ to predict labels on $\hat{\mathcal{X}}^{t\to s\to l}$
5: collect instances in the target domain that are predicted with high confidence: $\{(x_j^t, y_j^t)|(y_j^t, p_j) = f_{s\to l}(x_j^{t\to s\to l}), p_j \geq \tau\}$
6: train a classifier f_t on the above target instances $\{(x_j^t, y_j)\}$
7: predict the remaining unlabelled instances in $\mathcal{X}^t$

4.6 Conclusions

This chapter proposes UDAR as a knowledge-driven unsupervised domain adaptation algorithm to enable transferring activity recognition systems across heterogeneous domains. However, the limitations of UDAR are that knowledge-driven annotation can be not accurate and UDAR requires extra knowledge engineering effort. Furthermore, the sensor similarity matrix might not

be available for all transfer learning tasks, limiting this method's scope. In chapter 5, we present two deep learning models that do not rely on any predefined knowledge to tackle this problem.

Chapter 5

GAN-based Unsupervised Domain Adaptation Techniques

5.1 Overview

In this chapter, we present two data-driven GAN-based unsupervised domain adaptation techniques. The first method, called *shift*-GAN, is presented in section 5.4. It is proposed for resolving heterogeneous feature space between source and target domain using a Bidirectional Generative Adversarial Networks (Bi-GAN) and Kernel Mean Matching (KMM) in an innovative way to learn intrinsic, robust feature transfer between two heterogeneous domains. Most of the material of this section is extracted from our publication [140].

Although *shift*-GAN works well it does not separate classes that have similar patterns. To solve this limitation, in section 5.5 we introduce a second model called ContrasGAN that uses contrastive learning to minimise the intra-class discrepancy and maximise the inter-class margin. With contrastive learning, we hypothesise that we can better discriminate samples from different class labels and lead to more class-discriminative adaptation. This contribution has been accepted for publication in Pervasive and Mobile Computing Journal.

In section 5.2, we briefly describe unsupervised domain adaptation. Finally, in section 5.3, we introduce Generative Adversarial Networks (GAN), explain Bi-GAN architecture, and describe the process of performing domain adaptation using Bi-GAN.

5.2 Introduction

Let $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ be the labelled source domain and $\mathcal{D}_t = \{x_t^i\}_{i=1}^{N_t}$ be the unlabelled target domain, where $x^s (\in \mathbb{R}^{M_s})$ and $x^t (\in \mathbb{R}^{M_t})$ is a M_s and M_t-dimensional feature vector, and M_s can be different from M_t. Both domains share the same label space $\mathcal{Y}$. We aim to perform adaptation between $\mathcal{D}_s$ and $\mathcal{D}_t$ with the objective to predict labels for all the instances in $\mathcal{D}_t$.

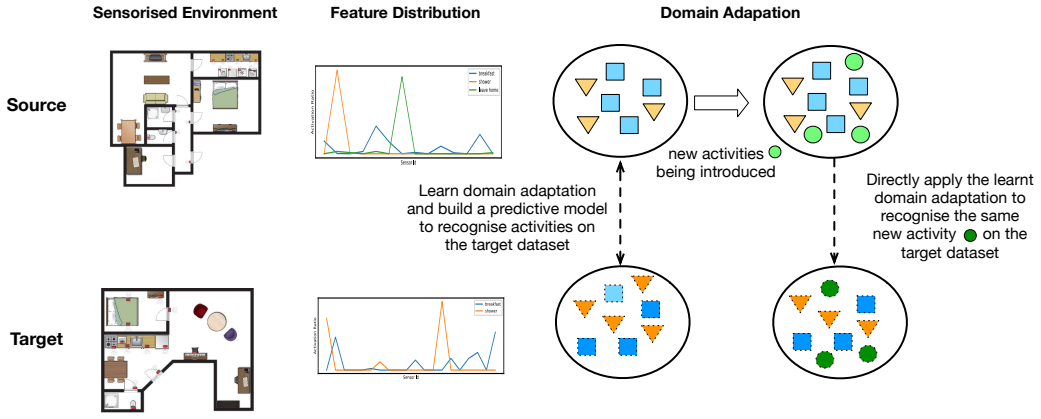


Figure 5.1: A use case [79] of generalised unsupervised domain adaptation.

We will illustrate the above definition through an example in Figure 5.1. Assume that there are two sensorised house settings (i.e., source and target) having different spatial layouts and installed with different sensors (as marked in red dots) [79]. The sensor data collected on these two houses are different and so are the sensor features extracted. The activity set $\mathcal{Y}$ to predict can be the same; i.e., a common set of daily activities such as preparing breakfast and performing personal hygiene.

Our aim is to learn a feature space transformation function $g_{s\to t}$ that maps the source domain features into the target domain features; i.e., $g_{s\to t}(x^s) = \tilde{x}^t$. Then, we can build a classifier with transformed data $\{(\tilde{x}_i^t, y_i^s)\}_{i=1}^{N_s}$, with which we can predict labels on real target data D_t.

We consider the transformation function $g_{s\to t}$ is *generalised* or activity-invariant, if it can be applied to sensor data on emerging, new activities that have not been observed in $\mathcal{D}_s$. Let $D_s' = \{(x_i'^s, y_i')\}_{i=1}^{N_s'}$ be a new collection of labelled source domain's data, which has the same feature space as $\mathcal{D}_s$ but has a different label space; that is, $y_i'^s \in \mathcal{Y}'^s$ and $\mathcal{Y}'^s \cap \mathcal{Y}^s = \emptyset$. The transformation function is regarded *intrinsic* to features, independent of specific activity classes

if $g_{s\to t}(x'^s)$ still holds on the new data $\mathcal{D}'_s$ without the need of retraining; i.e., $g_{s\to t}(x'^s) = \tilde{x}'^t$. In Figure 5.1, if the source domain's data are annotated with a new activity 'leaving home', we can use the function to transform the source domain data on this new activity to the target domain, without the need of retraining the function.

To achieve this feature transformation, we proposed to use Generative Adversarial Networks (GAN). GAN-based approach has been widely applied in domain adaptation [157]. In the following section, we describe GAN's architecture and explain how it can be used to perform domain adaptation.

5.3 Feature Transformation via GAN

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.

5.3.1 Generative Adversarial Network

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.

The idea behind GAN is to train two models – a generator and a discriminator – in an adversarial process. The generator G takes as input a random noise vector z and uses a multilayer perceptron with $\theta^{(G)}$ as parameters such as weights and biases. The discriminator D estimates the probability of a given sample coming from a real dataset. It takes as an input x and uses another multilayer perceptron with $\theta^{(D)}$ parameters. The models are represented by two functions, each of which is differentiable both with respect to its inputs and parameters.

The two models compete against each other during the training process: the generator G is trained to generate samples that could easily be mistaken for real data. While, the discriminator D is trained to maximise the probability of assigning the correct label to both training examples and generated samples from G. In other words, D and G are playing a **minimax game**. The discriminator wishes to minimise $J^D(\theta^{(D)}, \theta^{(G)})$ while controlling only $\theta^{(D)}$. The generator wishes

to minimise $J^G(\theta^{(D)},\theta^{(G)})$ while controlling only $\theta^{(G)}$. Their interaction can be summarised in the following loss function. Let P_d be the original data's distribution, P_g be the generator's distribution, and P_z be the noise variable z's distribution.

$$\begin{aligned}\min_G \max_D \mathrm{L(D,G)} =& \mathbb{E}_{x\sim P_d}[\log D(x)] + \mathbb{E}_{z\sim P_z}[\log(1-D(G(z)))] \\ =& \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1-D(x))],\end{aligned} \tag{5.1}$$

where $\mathbb{E}_{x\sim P_d}[\log D(x)]$ corresponds to the log-likelihood of maximising the probability of assigning the correct label, and $\mathbb{E}_{x\sim P_g}[\log(1-D(x))]$ represents the log-likelihood of generating samples as real as possible.

5.3.2 Bi-directional GAN (Bi-GAN)

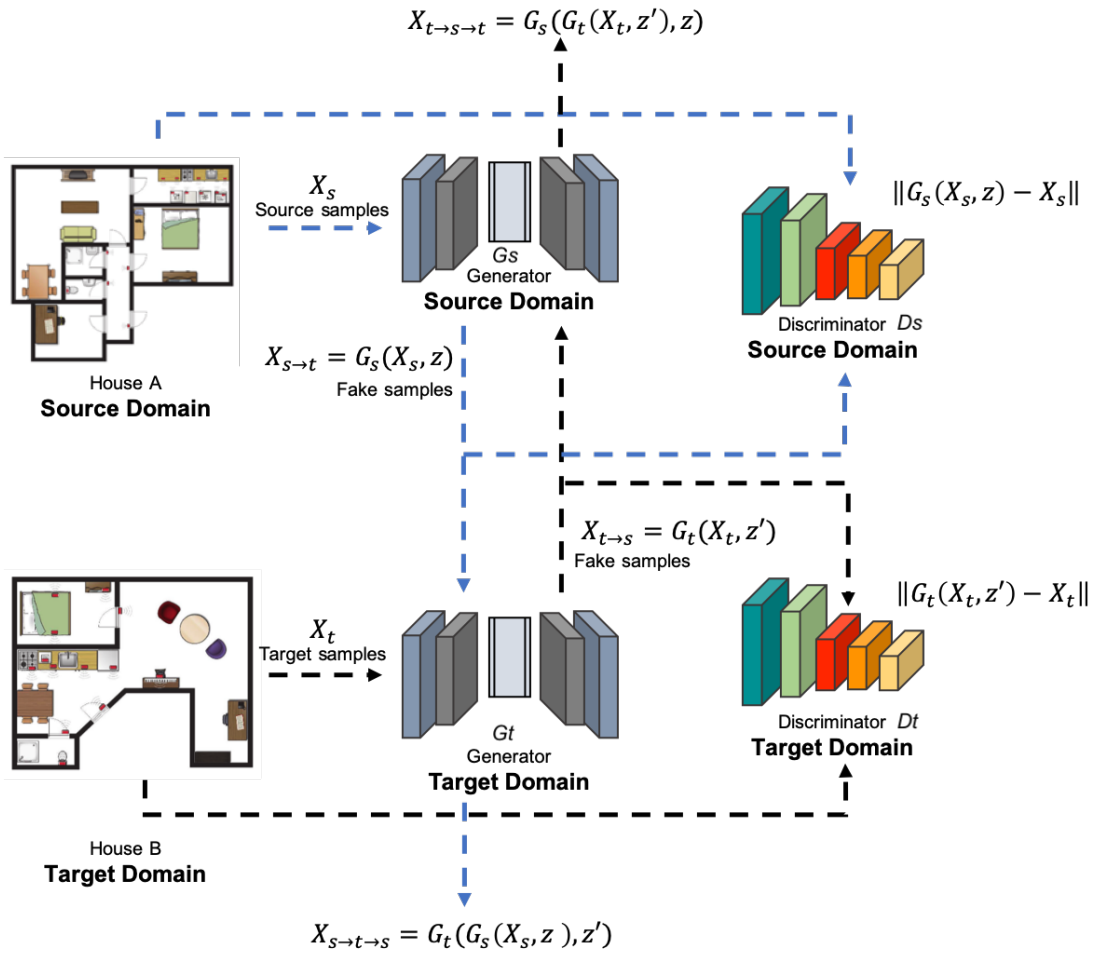


Figure 5.2: The architecture of Bi-GAN

Figure 5.2 describes the architecture of Bi-GAN. The Bi-GAN model consists of two GANs: $\{G_s, D_s\}$ and $\{G_t, D_t\}$, each composed of a generator and a discriminator on the source and target domain respectively. $G_s(x^s) = \tilde{x}^t$ takes a source instance x^s and generates a corresponding instance $\tilde{x}^t$ in the target domain. $G_t(x^t) = \tilde{x}^s$ takes a target instance x^t and generates a corresponding instance $\tilde{x}^s$ in the source domain. Both generators are trained to generate fake samples as close as to the real samples in the other domains and their objective function is to minimise the reconstruction losses:

$$\mathcal{L}_s^g = ||G_t(G_s(x^s, z), z') - x^s||, \tag{5.2}$$

$$\mathcal{L}_t^g = ||G_s(G_t(x^t, z'), z) - x^t||, \tag{5.3}$$

where z and z' are random noise introduced in G_s and G_t.

The discriminator D_s is a binary classifier to detect whether an input is generated by G_s or a real sample from the target domain, and D_t is to detect whether an input is generated by G_t or a real sample from the source domain. Their loss functions are defined as:

$$\mathcal{L}_s^d = D_s(G_s(x^s, z)) - D_s(x^t), \tag{5.4}$$

$$\mathcal{L}_t^d = D_t(G_t(x^t, z')) - D_t(x^s) \tag{5.5}$$

The combined loss function on both generators and discriminators is:

$$\begin{aligned} \mathcal{L}^g(x^s, x^t) = & \lambda_s ||G_t(G_s(x^s, z), z') - x^s|| + \\ & \lambda_t ||G_s(G_t(x^t, z'), z) - x^t|| - \\ & D_t(G_t(x^t, z')) - D_s(G_s(x^s, z)) \end{aligned} \tag{5.6}$$

5.4 *shift*-GAN

This section presents the four main steps of *shift*-GAN and explains how we integrate Bi-GAN and KMM to learn intrinsic, robust transfer between two domains, which are activity-invariant. *shift*-GAN works as follow:

1. *Feature space transformation* – perform unsupervised feature space transformation between source and target datasets with GAN; that is, we learn the mapping function $g_{s\to t}$ and obtain $\tilde{X}^t = g_{s\to t}(X^s)$.

2. *Feature distribution alignment* – shift the transformed features $\tilde{X}^t$ towards the real target data X^t; that is, $\bar{X}^t = \beta\tilde{X}^t$, where $\beta = [\beta_1, \beta_2, ..., \beta_N]$, N is the size of transformed samples $\tilde{X}^t$, and β_i is a weighting factor on each transformed sample.

3. *Classifier training* – train a classifier on the aligned, transformed features $\bar{X}^t$ and their corresponding labels inherited from the source domain.

4. *Prediction* – use the trained classifier to predict labels on the data in the target domain.

5.4.1 Feature Space Transformation

Algorithm 2 Bi-GAN training [189]

Data: Unlabelled source domain $\mathcal{D}_s = \{x_i^s\}_{i=1}^{N_s}$ and unlabelled target domain $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$
Build two generators G_A and G_B and two discriminators D_A and D_B

repeat
  foreach *iteration* **do**
    sample L-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{x_j^s\}_{j=1}^L \subseteq \mathcal{D}_s$ and $\{x_j^t\}_{j=1}^L \subseteq \mathcal{D}_t$
    update the parameters on D_s to minimise $\frac{1}{L}\sum_{j=1}^L \mathcal{L}_s^d(x^s, x^t)$
    update the parameters on D_t to minimise $\frac{1}{L}\sum_{j=1}^L \mathcal{L}_t^d(x^t, x^s)$
  sample L-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{x_j^s\}_{j=1}^L \subseteq \mathcal{D}_s$ and $\{x_j^t\}_{j=1}^L \subseteq \mathcal{D}_t$
  update the weights on both generators to minimise $\frac{1}{L}\sum_{j=1}^L \mathcal{L}^g(x^s, x^t)$
until *converge*;

To perform the feature space transformation, we adopt the same training process of Bi-GAN [189], which does not need labels on neither source nor target domains. The training process is shown in Algorithm 2. At the end of training, both generators act as the mapping functions. Bi-GAN enables to transform examples from the source domain to the target domain. In principle, the generator can generate many instances on each source example. The quality of each instance can vary due to the random variable z. In image-to-image application, human experts can visually inspect the images and perform the selection process. However, this practice

is infeasible for sensor data generation, so we extend one-to-many instance generation and selection process.

For a given source example x^s, we use G_s to generate N number of target samples, calculate their reconstruction loss using Eq (2) (3), and order them in an ascending order. Then we select the top-k ($1 \leq k \leq N$) samples that have the smallest reconstruction loss. The rationale is to choose the best transformed samples for the target domain while covering the diverse feature space by using k samples. In the end, we will have $\tilde{X}^t = \{\tilde{x}_j^t\}_{j=1}^{N_s'}$ ($N'^s = k * N^s$), where $\tilde{x}_j^t = G_s(x_i^s)$ ($1 \leq i \leq N_s$ and $1 \leq j \leq N_s'$).

5.4.2 Covariate Shift Correction via Kernel Mean Matching

The transformed examples $\tilde{X}_t$ might still not reflect the true target data distribution. To better align the distribution, we are looking into Kernel Mean Matching (KMM), which is designed as a non-parametric distribution matching method between training and testing samples. KMM reweights the training examples such that the means of the training and testing examples when projected in a Reproducing Kernel Hilbert Space (RKHS) are close. In this way, the training data will be better aligned with testing data, leading to improved classification accuracy [111]. KMM has been successfully applied with GAN to control the image generation process [75]. Inspired by the promising results, we will apply KMM to shift feature distributions to improve classification accuracy. In the following, we will briefly introduce the theoretical background of KMM and illustrate how it is integrated in *shift*-GAN.

The idea of KMM is to assign each instance in generated target domain data $\{\tilde{x}_i^t\}_{i=1}^{N_s'}$ with an importance weight β_i, which will be factored in a weighted loss function on a classifier f:

$$L_w(f) = \sum_{i=1}^{N_s'} \beta_i l(f(\tilde{x}_i^t), y_i). \tag{5.7}$$

The purpose of the importance weight is to shift the source domain data closer to the target domain data such that $P(X^t) = \beta P(\tilde{X}^t)$, where $\beta = [\beta_1, \beta_2, ..., \beta_{N_s'}]$. In order to find suitable

values of $\beta \in \mathbb{R}^{N'_s}$, we need to minimise the discrepancy between means of $\tilde{X}^t$ and X^t subject to

$$\beta_i \in [0,1] \text{ and } |\frac{1}{N'_s}\sum_{i}^{N'_s}\beta_i(x_i^s) - 1| \leq \varepsilon, \tag{5.8}$$

where ε is set as 0.01. The first part limits the scope of discrepancy between $P(\tilde{X}^t)$ and $P(X^t)$ and ensures the robustness by limiting the influence of individual instances. The second part ensures that $\beta P(\tilde{X}^t)$ is close to a probability distribution.

To find β, a feature space $\mathcal{F}$ is used, which is a RKHS with a universal kernel $k(x,x') = \langle \Phi(x), \Phi(x') \rangle$. With the feature map Φ: $X_t \to \mathcal{F}$, we define

$$K_{ij} := k(\tilde{x}_j^t, x_j^t) \tag{5.9}$$

$$\kappa_i := \frac{N'_s}{N_t}\sum_{j=1}^{N_t} k(\tilde{x}_i^t, x_j^t) \tag{5.10}$$

Then the discrepancy equation is defined as:

$$\begin{aligned} &||\frac{1}{N'_s}\sum_{i=1}^{N'_s}\beta_i\Phi(\tilde{x}_i^t) - \frac{1}{N_t}\sum_{j=1}^{N_t}\Phi(x_j^t)||^2 \\ &= \frac{1}{N'^2_s}\beta^T K\beta - \frac{2}{N_t^2}\kappa^T\beta + C \end{aligned} \tag{5.11}$$

where C is a constant. Thus, finding suitable β can be formulated as a quadratic problem [59], such that

$$\begin{aligned} &\min_{\beta} \frac{1}{2}\beta^T K\beta - \kappa^T\beta \\ &\text{subject to } \beta_i \in [0,B] \text{ and } |\sum_{i=1}^{N'_s}\beta_i - N'_s| \leq N'_s\varepsilon. \end{aligned} \tag{5.12}$$

5.4.3 Prediction on Target Dataset

Algorithm 3 presents an overall algorithm of *shift*-GAN and Figure 5.3 shows the workflow. It starts with training two generators G_s and G_t and then with G_s we can transform source dataset into target dataset. Then we align the transformed data with unlabelled target dataset to learn weighting factor β. After alignment, we build a SVM classifier with the transformed source dataset, with which we can predict labels on the transformed target dataset. SVM classifier

demonstrated superior performance than other classifiers. These results will be discussed in chapter 7.

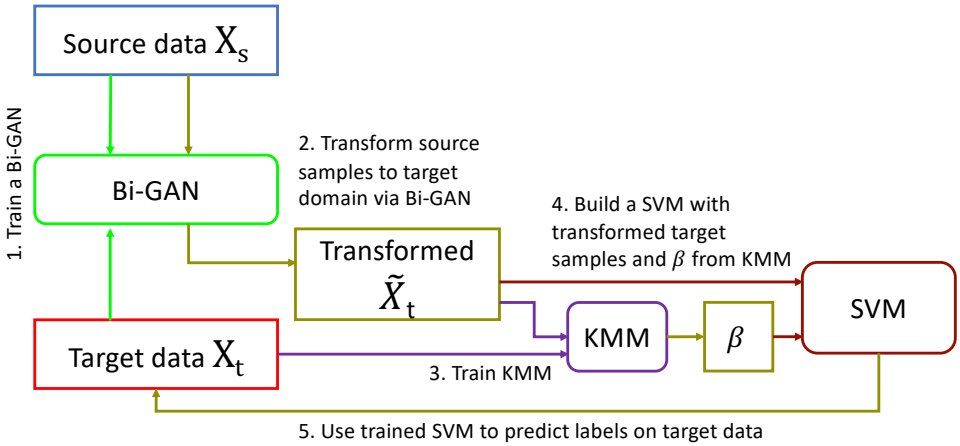


Figure 5.3: The overall workflow of *shift*-GAN

Algorithm 3 *shift*-GAN Training

Data: Labelled source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and unlabelled target domain $\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}$
Learn a generator G_s by training a Bi-GAN with $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_j^t\}_{j=1}^{N_t}$ in Algorithm 1
Generate top-k samples in the target domain on each source sample x_i^s: $\{\tilde{x}_l^t\}_{l=1}^{N_s'}$, $N_s' = k * N_s$
Learn β with $\{\tilde{x}_l^t\}_{l=1}^{N_s'}$ and $\mathcal{D}_t$ using Eq (12)
Build a SVM classifier f with β and $\{\tilde{x}_i^t\}_{i=1}^{N_s'}$
Predict labels for instances in $\mathcal{D}_t$

shift-GAN is a powerful technique that can learn invariant transformation functions between source and target domain. However, *shift*-GAN focuses on the translation between individual samples and does not consider the classes. To tackle this problem in the following section 5.5, we present a contrastive approach to better discriminate activities with similar patterns and improve the prediction of classes with few training samples.

5.5 Unsupervised Domain Adaptation via Contrastive Learning

This section presents the third contribution of this thesis called ContrasGAN. ContrasGAN is an unsupervised domain adaptation technique that introduces contrastive learning to improve

class alignment. It is composed of the following components and its workflow is presented in Figure 5.4:

1. *Feature space transformation* - perform unsupervised feature space transformation between source and target domains via Bi-GAN;
2. *Class-level alignment* - perform class-level alignment via contrastive learning;
3. *Target label prediction* - predict labels on the data in the target domain.

5.5.1 Feature Space Transformation

To perform feature space transformation, we modify the architecture of Bi-GAN presented in section 5.3.2 to introduce another term – *expectation loss*, which constrain the global mapping between two feature spaces [11] to better globally align the generated space with the original space:

$$\begin{aligned}\mathcal{L}_{exp} &= \mathbb{E}[(G_t(X^t) - X^s)^2] + \mathbb{E}[(G_s(X^s) - X^t)^2] \\ &= \frac{1}{n_t}\sum_{i=1}^{n_t}(G_t(x_i^t) - x_i^s)^2 + \frac{1}{n_s}\sum_{i=1}^{n_s}(G_s(x_i^s) - x_i^t)^2\end{aligned} \tag{5.13}$$

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

$$\mathcal{L} = \mathcal{L}_{G_s} + \mathcal{L}_{G_t} + \mathcal{L}_{exp} - \mathcal{L}_{D_s} - \mathcal{L}_{D_t} \tag{5.14}$$

The training of Bi-GAN goes as follows. We first train two discriminators D_s and D_t as binary classifiers on $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_i^t\}_{i=1}^{N_s}$ using the loss functions in Equations 5.3.2. 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 5.5.1. Then we iterate the above two steps for several iterations until both generators and discriminators converge.

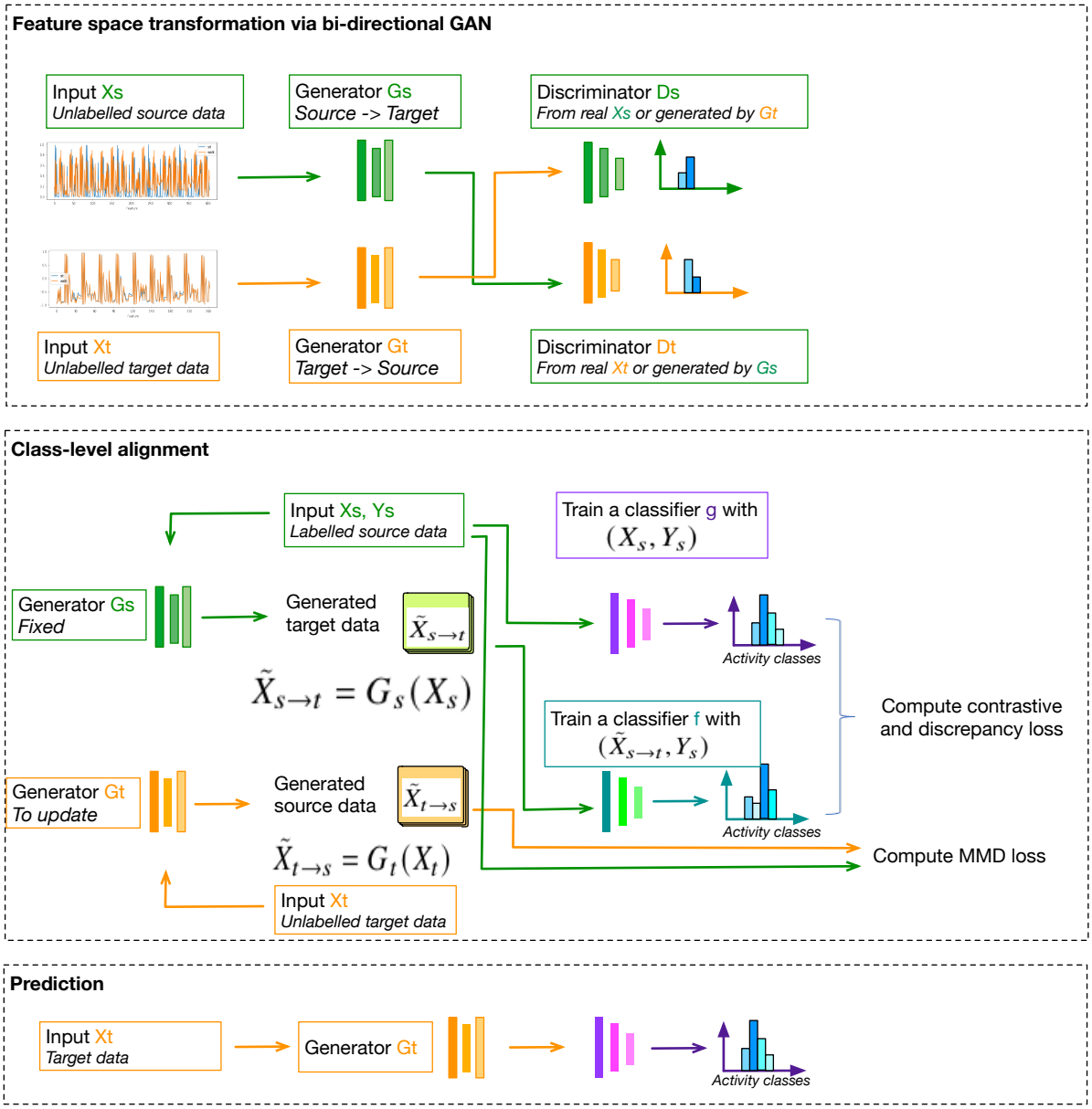


Figure 5.4: Workflow of ContrasGAN

5.5.2 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 into 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 the samples that have 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; that is, $\tilde{\mathcal{D}}_{s\to t} = \{(\tilde{x}_i^{s\to t}, y_i^s) \mid \tilde{x}_i^{s\to t} = G_t(x_i^s)\}$. 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 the classifiers 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 [37]. To further improve the performance, we add maxmimum 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.

5.5.2.1 Contrastive Loss

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

Once we have labelled the target data, we will use contrastive loss to minimise the intra-class discrepancy and maximise the inter-class margin. The intra-class domain discrepancy is minimised to compact the feature representations of samples within a class, whereas the inter-class domain discrepancy is maximised to push the representations of each other 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 it runs over pairs of samples to ensure that semantically similar samples are embedded close together. Here semantically similar samples means 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.

$$con\_dist((\tilde{x}_i^{t\to s}, \tilde{y}_i^{t\to s}), (x_j^s, y_j^s)) = \begin{cases} ||\tilde{x}_i^{t\to s} - x_j^s||^2 & \tilde{y}_i^{t\to s} = y_i^s \\ max(0, m - ||\tilde{x}_i^{t\to s}, x_j^s||^2) & \tilde{y}_i^{t\to s} \neq y_j^s \end{cases} \tag{5.15}$$

where $y^{t\to s}$ is the predicted label on the sample $\tilde{x}^{t\to s}$ and y^s is the label on the real source sample x^s. It measures the distance of a pair of similar samples which belong to the same class and constrains the distance of a pair of dissimilar samples which belong to different classes. m is pre-defined margin. The margin specifies 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 [62], the contrastive loss function is defined on the distance function as follows.

$$\mathcal{L}_{con} = \frac{1}{\varepsilon} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} con\_dist((\tilde{x}_i^{t\to s}, \tilde{y}_i^{t\to s}), (x_j^s, y_j^s)). \tag{5.16}$$

where ε is a parameter used to normalise the contrastive loss. The normalisation prevents the overall loss is dominated by the individual loss with higher magnitude. We employ a batch normalisation; that is, the loss is normalised by the number samples in a batch, which helps balance the magnitude of different loss components.

5.5.2.2 Discrepancy Loss

We also want to enforce the 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 x_i^t and $x_i^{t\to s}$ be a target sample and its corresponding transformed sample in the source domain. $p_i^t = [p_1^t, ..., p_N^t]$ be the probability output vector on x_i^t from the classifier f, indicating the confidence of inferring each class ($\in [1, ..., N]$). Similarly, $p_i^{t\to s}$ be the

probability output vector on $x_i^{t\to s}$ from the classifier g. The discrepancy loss between $p_i^{t\to s}$ and p_i^t is defined as:

$$\mathcal{L}_{disc} = \mathbb{E}[\sum_{i=1}^{n_t}(\frac{1}{N}\sum_{n=1}^{N}|p_i^{t,n} - p_i^{t\to s,n}|)] \tag{5.17}$$

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

5.5.2.3 MMD Loss

MMD is to estimate the distance of two distributions with their mean of projected embeddings in the reproducing kernel Hilbert space (RKHS) [18]. MMD is motivated by the fact that if two distributions are identical, all of their statistics should be the same.

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

$$mmd\_dist(X,X') = \left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i') \right\|_{\mathcal{H}} \tag{5.18}$$

where ϕ 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:

$$L_{MMD} = mmd\_dist(G_{t\to s}(X^t), X^s) \tag{5.19}$$

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:

$$\mathcal{L}_{mmd}(X^s, X^{t\to s}) = \sum_{i=1}^{N_s}\sum_{j=1}^{N_{t\to s}} k\left(\phi\left(\frac{\tilde{x}_j^{t\to s}}{\|\tilde{x}_j^{t\to s}\|}\right), \phi\left(\frac{x_i^s}{\|x_i^s\|}\right)\right) \tag{5.20}$$

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; $k(x,x') = \langle\phi(x), \phi(x')\rangle_{\mathcal{H}}$ [18]. The MMD loss forces

the normalised features in the two domains to be identically distributed improving the global domain alignment.

5.5.2.4 Algorithm and Training Regime

Algorithm 4 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 4 ContrasGAN Training

Data: Labelled source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and unlabelled target domain $\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}$

1. Train generators G_s and G_t by training a Bi-GAN with $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_j^t\}_{j=1}^{N_t}$
   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 $\mathcal{D}_t$; $\{x_j^s\}_{j=1}^{L}$ and $\{x_j^t\}_{j=1}^{L}$
    update the parameters on D_s to minimise $\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_s^d(x^s, x^t)$
    update the parameters on D_t to minimise $\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_t^d(x^t, x^s)$
  sample L-sized instances from both $\mathcal{D}_s$ and $\mathcal{D}_t$; $\{x_j^s\}_{j=1}^{L} \subseteq \mathcal{D}_s$ and $\{x_j^t\}_{j=1}^{L} \subseteq \mathcal{D}_t$
  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 $\mathcal{D}_t$; $\{(x_j^s, y_j^s)\}_{j=1}^{L}$ and $\{x_j^t\}_{j=1}^{L}$
      generate transformed target samples via G_t: $\{\tilde{x}_j^t | \tilde{x}_j^t = G_t(x_j^t)\}_{j=1}^{L}$
      infer class labels and posterior probabilities on original target data and transformed target data
      update the parameters of g to minimise $\mathcal{L}_{con}$
    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*;

5.6 Conclusions

This chapter presented two GAN-based unsupervised domain adaptation techniques for heterogeneous feature spaces. It only requires 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. 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. Below we summarise the main contributions of *shift*-GAN and ContrasGAN:

- We proposed *shift*-GAN as a general unsupervised domain adaptation technique to enable activity transfer across heterogeneous datasets, including accelerometer and binary sensors.
- We have extended Bi-GAN by not just performing one-to-one instance translation but one-to-many instance translation along with instance selection process to allow more robust domain adaptation.
- We have validated *shift*-GAN extensively on different transfer tasks across multiple datasets. All these datasets feature different sensor deployments, spatial layouts of environments, and different end users. Our results have demonstrated that *shift*-GAN has outperformed classic and deep domain adaptation techniques. The results are discuss in chapter 7.
- ContrasGAN makes it possible to adapt an activity model with a well-annotated dataset to a large number of real-world settings without the need of collecting any additional labels.
- Contrastive learning improves classification accuracy by learning a feature space where similar samples are put close to each other while dissimilar ones are pushed apart.

In the following chapter, we present the experimental setup and evaluation methodologies and in chapter 7, we discuss the evaluation results.

Chapter 6

Experimental Setup and Evaluation Methodologies

6.1 Introduction

We evaluate our methods in the context of cross-body, cross-user and cross-sensor human activity recognition. We compare them to baseline models and other competing domain adaptation techniques. Section 6.3 introduces the benchmark datasets used in the experiments. The general experimental setup is presented in section 6.6.3, including an introduction to the implementation framework, configuration and hyperparameter selection and an explanation of comparison techniques.

6.2 Evaluation Objectives

The main goal of the evaluation is to assess the effectiveness of UDAR, *shift*-GAN and ContrasDGAN in *generalised unsupervised* domain adaptation; that is, how accurately UDAR, *shift*-GAN and ContrasDGAN can recognise activities in the target domain without using any labelled data in the target domain, and to what extent the domain adaptation being learnt can be generalised. In the following, we will introduce the experiment setup and the implementation details.

6.3 Datasets

To assess the generality and feasibility of UDAR, *shift*-GAN and ContrasDGAN, we consider the two most common types of datasets: ambient binary sensors and accelerometers. All these 6 datasets are collected by third parties and are publicly available. These 6 datasets exhibit a wide range of domain adaptation challenges, especially the varying feature complexity (from 14 to 405) and a large number of activities (from 6 to 19). Also, there exists a high similarity between activity classes and high diversity of patterns in one activity class. All these challenges have added extra complexity to unsupervised domain adaptation, which will be discussed later. The detailed setting for each dataset is as follow.

6.3.1 Binary Sensor Datasets

For ambient binary sensor datasets, we use three datasets collected and curated by the University of Amsterdam (named A, B, and C respectively in the remainder of this chapter) [162]. They are collected on three different users in three different residential settings, each being deployed with binary sensors, including infra-red position sensors, switch sensors, and water flow sensors. In House A, the sensor network is composed of 14 state-change sensors on household objects like doors, cupboards, or toilet flush. In House B and C, each network node was equipped with heterogeneous sensors: passive infrared to detect motion in a specific area; pressure mats to measure whether someone is sitting on a couch or lying in bed; switches to monitor whether doors and cupboards are open or closed; mercury contacts to detect the movement of objects (e.g., drawers); and water flow sensors to detect the flush of the toilet. All these sensors output binary readings (0 or 1), indicating whether or not a sensor fires. In these three datasets, the activities “Toilet", “Leave House", and “Sleep" dominate the datasets, while the activity “drink” is the least frequently recorded activity.

For binary sensor data, we employed state-of-the-art techniques to extract features [45]; that is, the activation ratio within a fixed interval (i.e., 60 seconds) as sensor features. Figure 6.1 presents the activity distribution of these three datasets.

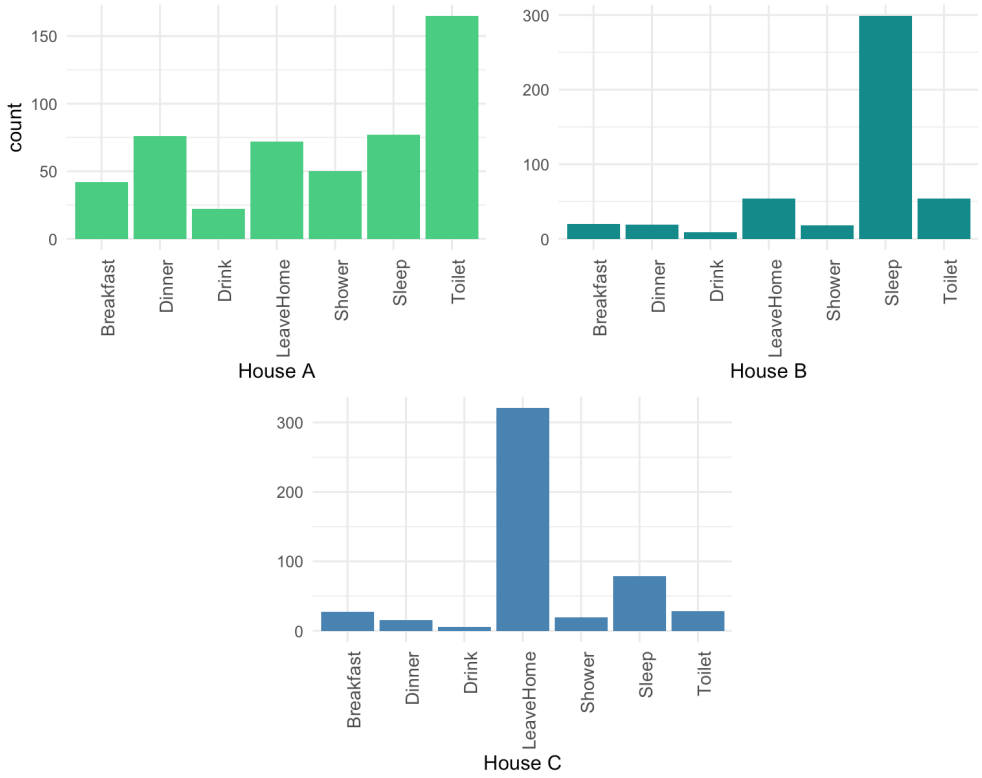


Figure 6.1: Activity distribution of the 3 binary sensor datasets used in evaluation

6.3.2 Accelerometer Sensor Datasets

For accelerometer data, we select 3 commonly-used HAR datasets: WISDM [172], UCI daily and sports (DSADS) [9, 8], and PAMAP2 (PAMAP) [133].

The WISDM dataset was collected from 51 subjects when they were performing 18 daily activities including walking, jogging, brushing teeth, eating and drinking. During the data collection, each subject either wore a smartwatch (i.e., LG G Watch) on their dominant wrist or had a smartphone (i.e., Samsung Galaxy or Google Nexus 5/5X) in their pocket. The accelerometer and gyroscope's data from both watch and phone were collected at a rate of 20 Hz. We use the handcrafted features from WISDM datasets, including the mean, standard deviation, mel-frequency cepstrum coefficients (MFCC) of each dimension, and correlations between dimensions [172]. The PAMAP dataset records 12 activities performed by 9 subjects, including sitting, lying, house cleaning, and ironing. Each subject wears 3 accelerometer units on their dominant arm, chest, and dominant side ankle. The DSADS consists of 19 daily activities performed by 8 subjects, including exercising on a stepper, rowing, and running on a treadmill. Each subject wears 5 accelerometer units on their torso, right arm, left arm, right leg, and left

leg. We use the feature dataset [170] generated from these two datasets. That is, 27 features are extracted per sensor on each body part, including mean, standard deviation, and spectrum peak position. Table 6.1 summarises the characteristics of these datasets.

Table 6.1: Descriptions of Datasets

Dataset		No. of Features	No. of Samples	No. of Users	No. of Activities	Activities
PAMAP		243	7352	9	12	ascending stairs, cycling, descending stairs, ironing, lying, nordic walking, rope jumping, running, sitting, standing, vacuum cleaning, walking
DSADS		405	9120	8	19	sitting, standing, lyiing on back and on right side, ascending and descending stairs, standing in an elevator still, moving around in an elevator, walking in a parking lot, walking on a treadmill with a speed of 4km/h, running on a treadmill with a speed of 8 km/h, exercising on a cross trainer, cycling on an exercise bike in horizontal and vertical positions, rowing, jumping, and playing basketball
WISDM	PHONE	90	40553	51	18	walking, jogging, stairs, sitting, standing, typing, brushing teeth, eating soup, eating chips, eating pasta, drinking from cup, eating sandwich, kicking, playing catch w/ Tennis Ball, dribllinlg, writing, clapping, folding clothes
	WATCH	90	34942			

For the experiments, we use similar and different body parts as described in Table 6.2. Beyond the tasks within each dataset, we also consider the tasks between PAMAP and DSADS on their four common activities, including walking, standing, sitting, and lying.

Table 6.2: Descriptions of transfer learning tasks on body parts

Dataset	Task	Body Parts
DSADS	RA-LA	right arm (RA) to left arm (LA)
DSADS	RL-LL	right leg (RL) to left leg (LL)
DSADS	RA-T	right arm (RA) to torso (T)
DSADS	LA-RA	left arm (LA) to right arm (RA)
DSADS	LL-RL	left leg (LL) to right leg (RL)
DSADS	T-RA	torso (T) to right arm (RA)
DSADS	RA-RL	rigth arm (RA) to right leg (RL)
DSADS	LA-LL	left arm (LA) to left leg (LL)
DSADS	LA-T	left arm (LA) to torso (T)
PAMAP	H-C	hand (H) to chest (C)
PAMAP	C-H	chest (C) to hand (H)
PAMAP	P.Ankle-P.Chest	ankle to chest
PAMAP	P.Ankle-P.hand	ankle to hand
PAMAP	P.Chest-P.Ankle	chest to ankle
PAMAP	P.hand-P.Ankle	hand to ankle

6.4 Implementation Frameworks and Libraries

The code[1] for the experiments is in Python, using the PyTorch, Numpy and Pandas libraries. The experiments have been run on a modest computer configured with an Intel Core i7-9700K CPU 3.60GHz and 32GB memory.

[1]The source code can be accessed here: `https://github.com/An5r3a`.

6.5 Configuration and Hyperparameter Selection

UDAR configuration consists of two main components: (1) for *the pre-annotation step* we used three base classifiers: the random forest classifier with 50 trees, SVM with RBF kernel and the grid parameter searching to find the optimal values for C and γ, and k-Nearest Neighbour (kNN) with $k = 5$. All the base classifiers are from the Scikit-learn library[2]. These classifiers are vanilla classifiers and their confidence probabilities are calculated as follows. SVM estimates the multi-class probability via Pairwise Coupling [177], RF computes the probability as the mean of the predicted class probabilities of the trees in the forest, and kNN computes the probability as the fraction of classes among the selected neighbours. On top of these three base classifiers, we build a stacked ensemble, which is implemented as a neural network consisting of 2 hidden layers and the sparse categorical cross-entropy loss function. (2) In *the domain adaptation step* all models are implemented with PyTorch, the loss function of the VAE is minimised using the RMSProp optimisation. The optimizer is parametrised with a learning rate of 10^{-2}. We use `tanh` as the activation function except for the output layer. The mini-batch size is set to 100 instances. In order to choose the best setting for VAE, we have done the grid search on the number of layers from 1 to 3 and the number of neurons from $S - S/2$ to $S + S/2$, where S is the number of sensor features and choose the setting that leads to the highest accuracy for each dataset.

***shift*-GAN** consists of two main components: Bi-GAN and SVM with KMM. For the Bi-GAN, both generators G_s and G_t have identical network architecture. The leaky ReLU activation function is used in both generators with the exception of the output layer which uses `tanh` function. We observed that using the leaky ReLU activation function allows the model to learn more quickly as it allows gradients to flow backwards through the layer unimpeded. A leaky ReLU is like a normal ReLU, except that there is a small non-zero output for negative input values. Mathematically, ReLU is defined as $y = \max\{0, x\}$. The downside of ReLU activation function is that it is zero for all negative values. This problem is called *dying* ReLU, which states that we can have *dead* neurons during the learning process if the derivative slope is zero. Once a neuron gets negative, it is unlikely for it to recover. Then the accumulated gradient for the weight update will be multiplied by zero. To fix this issue, Leaky ReLU assigns a small slop for negative values, which improves the learning process as it does not have zero-slope parts.

[2]Scikit-learn library can be accessed at: `https://scikit-learn.org/`.

To help the discriminators generalise better, we make use of the parameter smooth. This process is known as **label smoothing**. Label smoothing regularises a model based on a softmax with k output values by replacing the hard 0 and 1 classification values with values of $\frac{\varepsilon}{k}$ and $1 - \frac{k-1}{k}\varepsilon$, respectively. We update the cross-entropy loss with these soft values.

The loss function of GAN is minimised using the Adam optimisation. The optimiser is parameterised with a learning rate of 10^{-2} and the mini-batch size is set to 100. In order to choose the best setting, we have done the grid search on the number of layers from 1 to 3 and the number of neurons from $S - S/2$ to $S + S/2$, where S is the number of sensor features and choose the setting that leads to the highest accuracy for each dataset. Similarly, we have run grid search for configuring the weights of source and target generators λ_s and λ_t in Eq (6) in the range of [100, 1000].

ContrasDGAN consists of two main components: Bi-GAN and class-level alignment. 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, 20% dropout rate is used. The discriminator has a similar architecture as the generator and the only difference is that the last layer of the discriminator is for classification, which has sigmoid as activation function and has binary output. 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 set the optimiser as ADAM, the learning rate is set to $1e^4$, and decaying to $1e^3$. We train the model for 500 epochs with a batch size of 100.

We have done the grid-search with the hyperparameters including the number of layers for generators, discriminators, and classifiers, batch size, learning rate, and number of epochs and dropout rates. We choose the configuration that leads to the highest accuracy.

6.6 Evaluation Methodologies

This section describes the evaluation setup, including metrics and a general description of comparison techniques used to assess the effectiveness of our methods.

6.6.1 Evaluation Metrics

The accuracy of recognising activities is evaluated using two parameters: precision and recall. Precision is the ratio of the times that an activity is correctly recognised to the times that it is inferred. Recall is the ratio of the times that an activity is correctly inferred to the times that it actually occurs. The F1-score is the harmonic mean of precision and recall:

$$\text{F1-score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{6.1}$$

As our datasets are imbalanced, we use both micro-F1 and macro-F1 scores. The micro-F1 is the F1-scores averaged across all instances, and the macro-F1 is the F-scores averaged on all activity classes.

6.6.2 Comparison Techniques

We compare our methods with the state-of-the-art domain adaptation models and other competing domain adaptation techniques. First, we select 5 classic techniques that have achieved best performance in transferring heterogeneous feature spaces [135] including Geodesic Flow Kernel (GFK) [56], Transfer Component Analysis (TCA) [119], Feature-Level Domain Adaptation (FLDA) [86], Joint Distribution Adaptation (JDA) [97], and Importance-weighting with logistic discrimination (IW) [61], along with a linear baseline technique Canonical Correlation Analysis (CCA). Then we choose recent unsupervised domain adaptation techniques. We mainly focus on the recent adversarial techniques that have achieved high adaptation accuracy and demonstrated robustness in transferring heterogeneous feature spaces. Therefore, we select Adversarial Discriminative Domain Adaptation (ADDA) [157], Domain-Adversarial Neural Network (DANN) [53], Deep Adaptation Networks (DAN) [96], and Adversarial Domain Adaptation with Domain Mixup (ADADM) [179], Joint Adaptation Networks (JAN) [98], Deep Correlation Alignment (DeepCORAL) [151].

Beyond these techniques, we also add the upper bound and lower bound baseline: (1) for the upper bound baseline we train 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; and (2) for the lower bound baseline we train 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.

6.6.3 Experiments

We conducted experiments on several real-world datasets to assess the performance of UDAR, *shift*-GAN and ContrasDGAN in tackling the main types of domain adaptation tasks in human activity recognition:

Cross-body Transferring an activity model from one wearing body position to another; *e.g.*, from left leg to right leg. It is motivated by *wearing diversity* of wearable sensors [22] 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, which can reduce annotation cost and improve recognition accuracy and robustness to the variability of wearing positions.

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 [135]. This can be the most complicated domain adaptation task in HAR where the 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.

In the following, we provide a description of each experiment.

6.6.3.1 Performance of Unsupervised Domain Adaptation

Our first experiment is to assess the effectiveness of domain adaptation. To do so, we compare state-of-the-art domain adaptation techniques and deep learning-based domain adaptation techniques. For each technique, we use all the source domain data and randomly split the target domain data into 80% for training and 20% for testing. The labels on the target dataset are not used during training.

6.6.3.2 Impact of Training Data

We also assess the impact of training data on the effectiveness of domain adaptation. It is desirable to use less training data while achieving comparable accuracy. Therefore, in this experiment, we vary the percentage of training data in the target domain from 20% to 80% and assess the impact of the training data on the accuracy of domain adaptation.

6.6.3.3 Robustness to Sensor Noise

The performance of the sensors can vary over time affecting drastically the sensor features. For example, a sensor could break or a wrong calibration can cause signal interference resulting in deterioration in the measurement. The sensor configuration can be cost-inefficient for large-scale deployment and require a lot of maintenance to calibrate the sensors. Here we aim to assess the impact of sensor noise on the performance of domain adaptation and thus to shed light on sensor maintenance management. To do so, we systematically inject noise into sensor features and compare the accuracy of the recognition accuracy with the state-of-the-art domain adaptation techniques.

We inject random Gaussian noise into the target domain data to simulate the real-world situation where the environment to be adapted to is compromised with unexpected sensor noise. On the test data of the target domain, we randomly select a number of sensors, and for each randomly selected sensor, we inject it with Gaussian noise. The percentage of sensors is chosen from 25% to 100% with a step size of 25%. The mean and variance of Gaussian noise are randomly sampled between 0 and 1.

6.6.4 Summary

In this chapter, we have explained our experimental strategy including the configuration of the models, hyperparameter tuning and datasets used. These datasets are well-known public datasets for activity recognition for experimentation and evaluation. We have also provided a comprehensive explanation of each of the methods used for comparison. These approaches have achieved very good results on different transfer learning tasks. In the next chapter, we will proceed with the comparison results and discussion.

Chapter 7

Results and Discussion

7.1 Introduction

We conduct various experiments on three real-world datasets commonly used in human activity recognition to evaluate the domain adaptation performance of our methods and baseline techniques. The general experimental setup is presented in chapter 6, which includes a description of the learning tasks, benchmark datasets, hyperparameter selection, comparison techniques and evaluation metrics.

In this chapter, we report first the overall performance of UDAR, *shift*-GAN and ContrasGAN against several baseline domain adaptation techniques on binary and accelerometer data. In all accelerometer experiments, we do not compare our first proposed method UDAR. The reason is that UDAR is a knowledge-driven feature remapping technique that maps sensor features based on the sensor semantics which is not available for accelerometer data.

Accelerometer results are divided into two section. Section 7.2.2.1 presents the results for cross-body experiments that evaluate transferring an activity model from one wearing body position to another. Section 7.2.2.2 discusses the results for cross-sensor experiments which transfer an activity model across different sensing technologies, for example from phone to watch. In section 7.3, we discuss the stability and convergence of UDAR, *shift*-GAN and ContrasGAN.

We also investigate issues that could happen in real-world problems. In section 7.4 we analyse the impact of varying the percentage of training data on the effectiveness of domain adaptation and in section 7.5 we study the effects of sensor noise in domain adaptation. With these

experiments, we aim to understand which conditions can affect domain adaptation performance and to validate the robustness of our models. We will compare our methods with the best performing deep and non-deep learning-based techniques from the previous experiments.

7.2 Performance of Unsupervised Domain Adaptation

In this section, we analyse and compare the performance of UDAR, *shift*-GAN and ContrasGAN against baseline techniques. Beyond these techniques, we add the upper- and lower-bound baselines. We report results first for binary sensor data followed by accelerometer results.

7.2.1 Binary Sensor Data

The binary sensor data experiments measures the accuracy of transferring an activity model learnt on one house (*e.g.*, House A) to another house (*e.g.*, House B). We used three datasets curated by the University of Amsterdam (named *A*, *B*, and *C* respectively in the remainder of this chapter) [162]. On these three datasets, we define six adaptation tasks: A-B, B-A, A-C, C-A, B-C, and C-B. Here the task A-B means that A acts as the source domain and B as the target domain.

Tables 7.1 and 7.2 report the micro-F1 and macro-F1 scores of the proposed methods and the baseline techniques on *binary sensor data* experiments. ContrasGAN, *shift*-GAN and UDAR have achieved a performance improvement in micro-F1 score of 41%, 35% and 41% and in macro-F1 score of 46%, 32% and 43% over the lower bound accuracy, respectively.

Compared to deep learning-based domain adaptation techniques, UDAR outperforms ADADM, DADA, JAN, DANN, DeepCORAL and DAN in 14%, 7%, 42%, 4%, 34% and 5% in micro-F1 score, respectively. DANN outperforms UDAR on tasks A-B, B-A, A-C and C-A. However, for C-B and B-C it struggles in finding meaningful latent representations when the dataset has more noise. DADA also fails to minimise the domain discrepancy between the source and target domain with noisy datasets. DAN deals with large divergence better, however, UDAR outperforms DAN in 12% and 14% on B-C and C-B tasks, respectively. Although JAN is an extension of DAN, it has the worst performance between the adversarial techniques. The joint

maximum mean discrepancy does not improve minimising the shift between the distributions of the different houses.

It is evident that UDAR outperforms non-deep learning-based methods. More specifically, the performance improvement in micro-F1 score of UDAR over each technique is: 23% (GFK) , 35% (TCA), 51% (FLDA), 32% (JDA), 69% (IW) and 79% (CCA). On C-A, UDAR performs worse than GFK by 11.4%, which is because House C is very noisy, finding a joint subspace that is still discriminative is hard.

Table 7.1: Comparison of micro-F1 scores between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary datasets.

Task	Lower bound	ContrasDGAN	shift-GAN (2021)	UDAR (2020)	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	FLDA (2016)	JDA (2014)	GFK (2012)	IW (2012)	TCA (2011)	CCA (2001)	Upper bound
A-B	0.54	**0.91**	0.80	0.85	0.60	0.86	0.24	0.91	0.31	0.78	0.48	0.26	0.74	0.14	0.36	0.22	0.87
B-A	0.43	0.81	0.79	0.78	0.82	0.83	0.70	0.86	0.70	**0.88**	0.09	0.68	0.27	0.65	0.76	0.12	0.88
A-C	0.48	0.83	0.81	0.82	0.51	**0.93**	0.23	0.86	0.33	0.77	0.38	0.46	0.75	0.14	0.56	0.25	0.88
C-A	0.40	0.84	0.77	0.80	0.87	0.78	0.58	**0.87**	0.64	0.82	0.21	0.73	0.65	0.12	0.62	0.11	0.83
B-C	0.42	0.88	0.79	**0.92**	0.78	0.53	0.68	0.70	0.64	0.81	0.70	0.64	0.83	0.12	0.61	0.15	0.87
C-B	0.38	0.84	0.80	**0.92**	0.78	0.78	0.51	0.67	0.72	0.79	0.62	0.68	0.69	0.43	0.39	0.23	0.86
Avg.	0.44	0.85	0.79	0.85	0.73	0.79	0.49	0.81	0.56	0.81	0.41	0.57	0.65	0.27	0.55	0.18	0.87

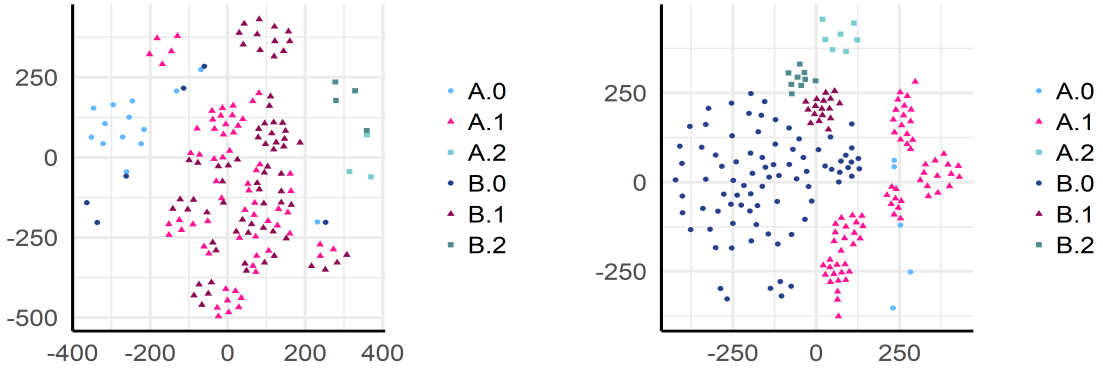
Table 7.2: Comparison of macro-F1 scores between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary datasets.

Task	Lower bound	ContrasDGAN	shift-GAN (2021)	UDAR (2020)	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	FLDA (2016)	JDA (2014)	GFK (2012)	IW (2012)	TCA (2011)	CCA (2001)	Upper bound
A-B	0.50	**0.9**	0.68	0.84	0.51	0.85	0.11	0.88	0.15	0.72	0.40	0.23	0.71	0.07	0.22	0.18	0.86
B-A	0.36	0.81	0.76	0.78	0.71	0.70	0.58	0.78	0.58	**0.86**	0.09	0.62	0.23	0.43	0.69	0.12	0.88
A-C	0.45	0.83	0.74	0.77	0.34	**0.93**	0.10	0.84	0.16	0.72	0.29	0.41	0.51	0.13	0.70	0.20	0.87
C-A	0.29	0.82	0.62	0.75	0.83	0.64	0.44	**0.78**	0.50	0.78	0.07	0.69	0.60	0.11	0.61	0.10	0.83
B-C	0.40	0.87	0.76	**0.87**	0.64	0.36	0.56	0.59	0.50	0.76	0.66	0.59	0.65	0.07	0.58	0.10	0.86
C-B	0.31	0.83	0.68	**0.87**	0.64	0.64	0.38	0.57	0.62	0.73	0.57	0.63	0.64	0.38	0.34	0.18	0.85
Avg.	0.38	0.84	0.70	0.81	0.61	0.69	0.36	0.74	0.42	0.76	0.35	0.53	0.56	0.20	0.52	0.15	0.86

Between the non-deep learning-based techniques, CCA is the worst, which reflects its limitations. CCA requires a one-to-one correspondence between data points in the source and target domains. This correspondence is then used to find the linear transformation to correlate both domains. To align the dimensions of the data points in both domains during the domain adaptation step, first, we select the instances from a class from the source domain and instances from the target domain on the same class using the pseudo labels. If the size of instances is different in each domain, we select a random sample to fit the dimensions. The canonical functions that maximise the correlation between both domains might depend on the random samples in each set. The sample size per class is important when the sample size is small; *i.e.*, if only a few instances in a certain class are selected, the learnt canonical correlation can be meaningless and not effective. On the other hand, a smaller number of samples from one domain

can be dominated by the samples from the other domain, leading to incorrect representation of the instances with small number of samples. For example, in the A-B task, the activity 'drink' represents 4% and 1% of the activity distribution in House A, and House B, respectively. CCA requires more instances for the alignment to be possible. When the sample size contains less than 5 instances, CCA will struggle to find a meaningful correlation between both samples.

JDA struggles to adapt the marginal distributions and conditional distributions when the source and target domains are considerably dissimilar. FLDA constructs a feature-level transfer model that calculates the difference between the target and source domain for each feature individually. However, its working assumption does not suit the problem that we are targeting. FLDA assumes a strong correlation between features on the corresponding activities in the source and target domain. For example, given that a sensor *S* is related to an activity 'shower' in House A, and House A and B have similar sensor features, then FLDA will assume that the mapped sensor *S* is only related to the activity 'shower' in House B, but not to any other activities. However, it is difficult to distinguish activities that activate a common set of sensors.



(a) House A to B using UDAR

(b) House A to B using TCA

Figure 7.1: Activity visualisation in transferring House A to B. t-SNE is applied on the feature representations of (a) the latent feature space on UDAR, and (b) the common subspace learnt on TCA for both the source and target domain. The activity labels for the source domain are A.0 - Leave Home, A.1 - Toilet, A.2 - Shower, and for the target domain are B.0 - Leave Home, B.1 - Toilet, B.2 - Shower.

Figure 7.1 visualises the feature spaces transformed in UDAR and TCA onto a 2D plot using t-Distributed Stochastic Neighbor Embedding (t-SNE) [159], a multidimensional scaling technique. In t-SNE, the pairwise distances $\delta_{ij}^2 = \|x_i - x_j\|^2$ between the high-dimensional data points x_i and x_j are converted into a joint probability distribution P over all pairs of non-identical points. The matrix P has entries:

$$p_{ij} = \frac{exp(-\delta_{ij}^2/\sigma)}{\sum_k \sum_{l \neq k} exp(-\delta_{ij}^2/\sigma)} \tag{7.1}$$

for $\forall i$ and $\forall j$ such as $i \neq j$. The aims of t-SNE is to model each object by a point y_i in a low-dimensional map in such a way that the pairwise similarities p_{ij} are modeled as well as possible in the map.

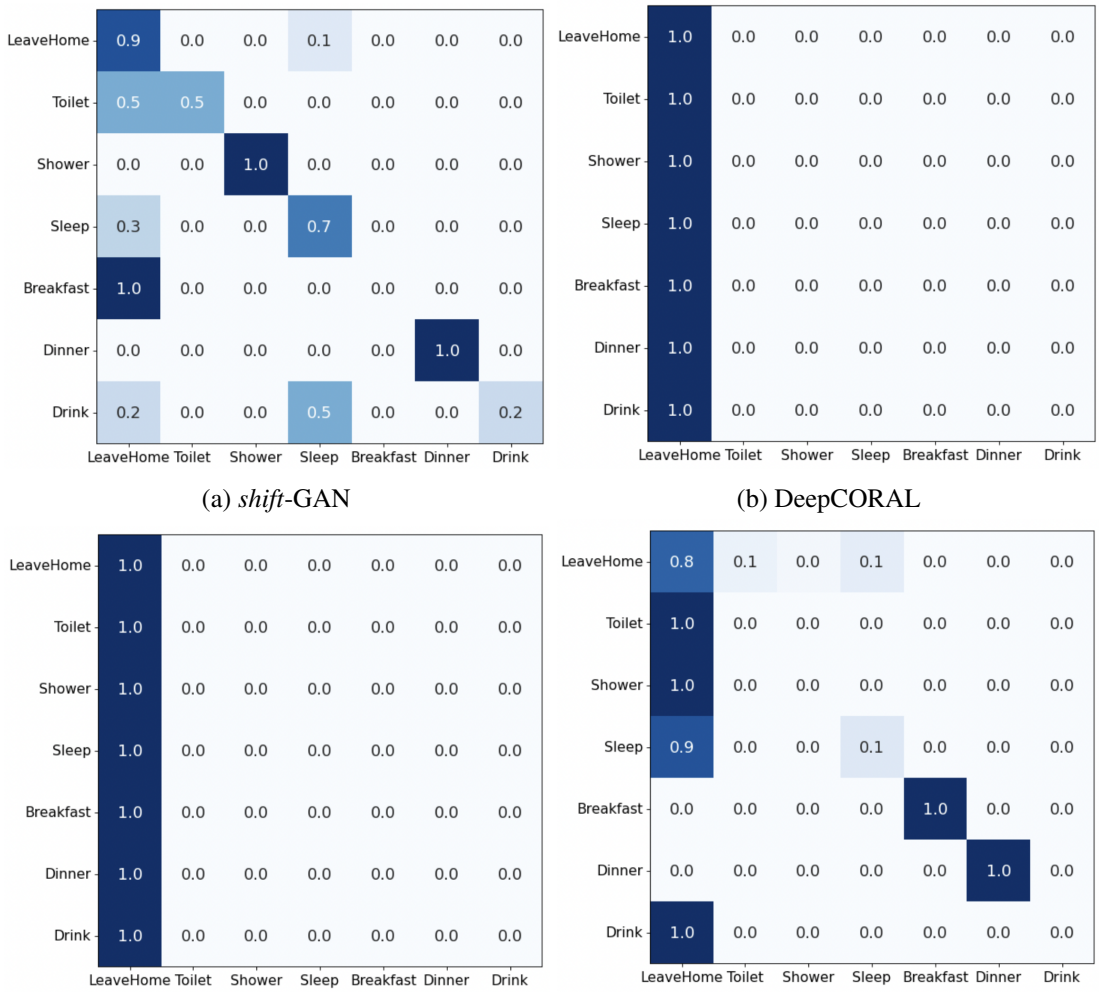
As we can see, in Figure 7.1a, when we encode the feature spaces of the source and target domain in the latent feature space learnt from UDAR, the data points that correspond to the same activity are clustered together, implying that the latent representations from the source and target domains are well aligned. On the contrary, the data points for the same activity are more separated in Figure 7.1b, which visualises the latent representations learnt from TCA on both source and target domains. This means that the latent space of TCA fails to capture inherent common representations of the source and target domain.

UDAR outperforms *shift*-GAN in micro-F1 and macro-F1 scores by 6% and 11%, respectively. This is because UDAR is a knowledge-driven method that aligns sensor features using a sensor similarity matrix. Domain knowledge and the use of VAE to align feature space help improving activity recognition. However, UDAR can not be applied in many situations as it requires a sensor mapping which limits the application of the approach. *shift*-GAN does not require engineering effort to map two domains. *shift*-GAN leverages the generative capability of GAN in mapping features between the source and target domain on high dimensional and high heterogeneous spaces.

DANN and DAN outperform *shift*-GAN with 3% and 2% in micro-F1 scores, respectively. The reason is that they leverage convolutional layers to learn transferred features on the combined source and target features and then minimise the discrepancy on embeddings on task-specific layers. Combining features together allows better feature contrast and alignment than *shift*-GAN where the focus is only to learn the mapping functions between domains.

Looking across the tasks, we can see that *shift*-GAN performs better than ADADM, DADA, JAN, DANN and DeepCORAL on tasks B-C and C-B. This means that *shift*-GAN is more robust to noisy datasets. *shift*-GAN has an improvement over the lower bound of 35% and 32% in micro-F1 and macro-F1 scores respectively. Most deep learning techniques do not perform well especially on B-C task. This is the case of DeepCORAL, DADA and JAN which achieve a

micro-F1 score of 64%, 53% and 68% respectively. Figure 7.2 presents the confusion matrix of *shift*-GAN, DeepCORAL, DADA and JAN on the B-C task. We can see that DeepCORAL and DADA are biased towards the activity with more training instances and are unable to recognise other activities. JAN shows a similar pattern but it can distinguish other activities such as 'Breakfast' and 'Dinner'. Although *shift*-GAN struggles in recognising 'Breakfast', is more effective in finding discriminative features and is better dealing with imbalanced datasets.

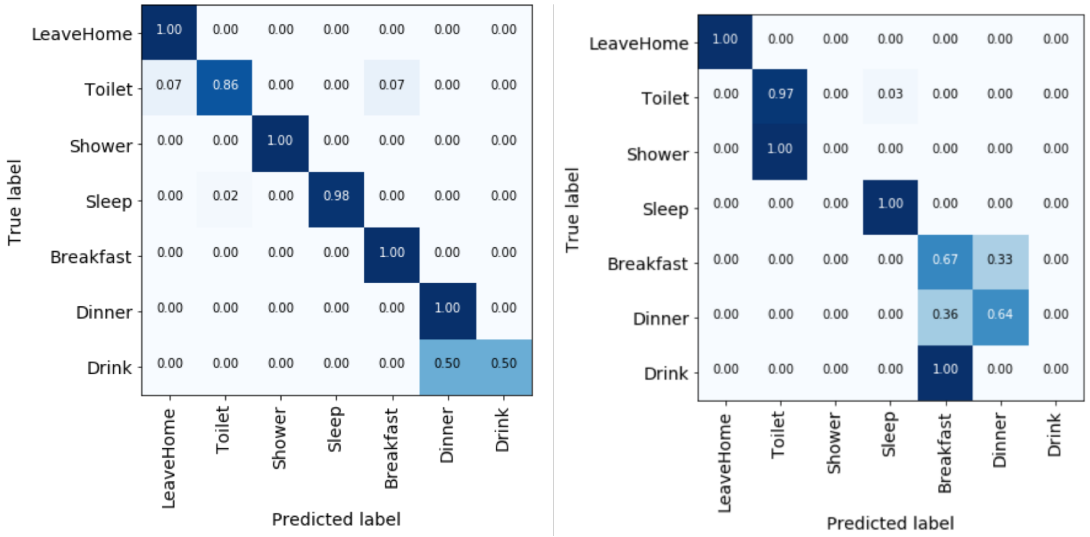


(a) *shift*-GAN (b) DeepCORAL

(c) DADA (d) JAN

Figure 7.2: Confusion matrices on the B-C task

Among the non-deep learning-based techniques, GFK performs better, but still 25% less than *shift*-GAN in both micro-F1 and macro-F1 scores. Figure 7.3 presents the confusion matrix of *shift*-GAN and GFK on the A-B task. We can see that GFK has good discriminative power; however, *shift*-GAN is more capable of recognising activities that have fewer distinctive patterns



(a) *shift*-GAN (b) GFK

Figure 7.3: Comparison of confusion matrices on task A-B between *shift*-GAN and GFK

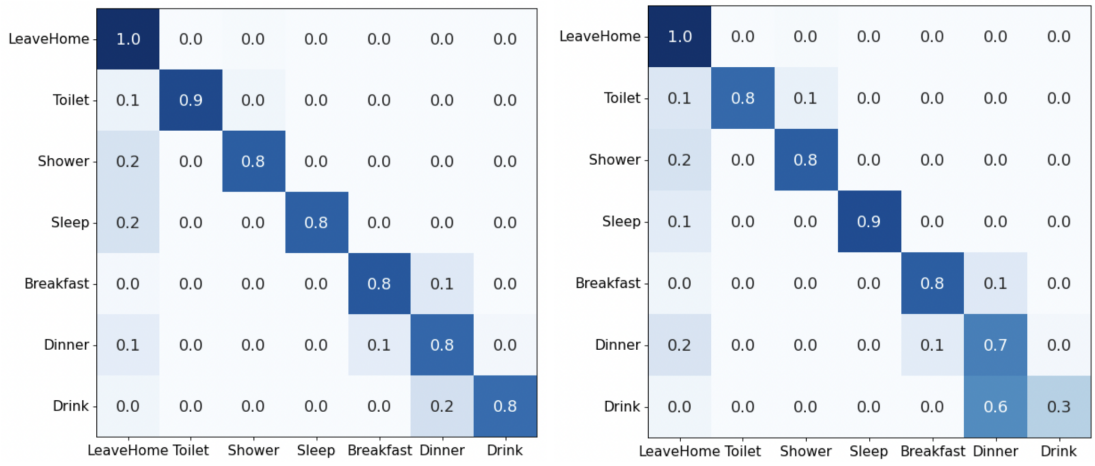
like 'Drink', and is better at finding discriminative features between activities that fire the same set of sensors; for example, 'Toilet' and 'Shower'. Therefore, we conclude that *shift*-GAN is more effective than GFK when dealing with imbalanced datasets and is better at recognising activities that have less frequent patterns than the other techniques.

CCA, IW and FLDA perform the worst. Compared to CCA and IW, *shift*-GAN is less affected by the sample size and it performs well when there is little overlap between the source and target domain [86].

ContrasGAN outperforms *shift*-GAN by 6% and 11% in micro-F1 and macro-F1 scores. Figure 7.4 presents the confusion matrix of ContrasGAN and *shift*-GAN on the C-B task. House C is a very noisy dataset, making it more difficult to find discriminative features. We can see that *shift*-GAN has good discriminative power, however, ContrasGAN is more capable of differentiating between activities that have less distinctive patterns; for example, 'Drink'. This demonstrates the effectiveness of using a contrastive learning to find better discriminative features.

7.2.2 Accelerometer Sensor Data

For the accelerometer sensor, data we define two types of experiments: *Cross-body* to measure the performance of domain adaptation techniques in transferring an activity model from one



(a) ContrasGAN (b) *shift*-GAN

Figure 7.4: Comparison of confusion matrices on task C-B between ContrasGAN and *shift*-GAN

wearing body position to another. It is motivated by *wearing diversity* of wearable sensors [22] in that users tend to change where to put their sensors or wearables depending on their preference and their current activities, and *Cross-sensor* to test the effectiveness of domain adaptation across different sensing technologies. In addition, this experiment tackles a problem where a system tries to deploy a new set of sensors for activity recognition without collecting any activity labels.

7.2.2.1 Cross-Body Experiments

For *cross-body* experiments, we define 13 domain adaptation tasks using DSADS and PAMAP datasets. On the DSADS dataset, we perform 9 transfer learning tasks, (1) between the sides of the same position including 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 parts: hand (H), chest (C), and Ankle (A).

Table 7.3: Comparison of micro-F1 scores between ContrasGAN, *shift*-GAN and baseline techniques on accelerometer datasets.

Task	Lower bound	ContrasDGAN	shift-GAN	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	FLDA (2016)	JDA (2014)	GFK (2012)	IW (2012)	TCA (2011)	CCA (2001)	Upper bound
RA-LA	0.68	0.90	0.91	0.74	0.75	0.70	0.78	0.55	0.72	0.59	0.74	0.74	0.43	0.67	0.18	0.94
RL-LL	0.80	0.90	**0.95**	0.60	0.82	0.63	0.73	0.55	0.78	0.55	0.66	0.69	0.49	0.60	0.16	0.97
RA-T	0.47	0.89	0.90	0.60	0.67	0.56	0.78	0.55	0.72	0.47	0.50	0.49	0.37	0.51	0.12	0.95
H-C	0.41	0.65	0.59	0.58	0.59	0.44	0.64	0.62	0.72	0.58	0.52	0.70	0.48	0.56	0.19	0.88
LA-RA	0.67	0.81	0.71	0.58	0.56	0.69	0.64	0.51	0.83	0.69	0.46	0.58	0.50	0.53	0.16	0.93
LL-RL	0.58	0.87	0.71	0.65	0.65	0.68	0.55	0.51	0.83	0.42	0.72	0.61	0.38	0.50	0.13	0.89
T-RA	0.43	0.86	0.63	0.60	0.61	0.62	0.54	0.49	0.82	0.36	0.79	0.56	0.36	0.54	0.11	0.98
C-H	0.49	0.83	0.52	0.64	0.65	0.65	0.55	0.51	0.83	0.37	0.43	0.62	0.36	0.52	0.14	0.94
RA-RL	0.43	0.80	0.68	0.65	0.63	0.67	0.54	0.51	0.80	0.44	0.36	0.66	0.38	0.52	0.12	0.92
LA-LL	0.57	0.80	0.77	0.68	0.71	0.65	0.65	0.52	0.82	0.69	0.44	0.67	0.48	0.53	0.19	0.97
LA-T	0.54	0.82	0.71	0.67	0.66	0.68	0.67	0.50	0.84	0.55	0.36	0.63	0.50	0.55	0.16	0.99
P.Ankle-P.Chest	0.36	0.82	0.74	0.70	0.68	0.70	0.63	0.53	0.81	0.42	0.72	0.68	0.36	0.57	0.13	0.96
P.Ankle-P.hand	0.34	0.87	0.73	0.80	0.80	0.80	0.48	0.50	0.85	0.36	0.79	0.66	0.38	0.59	0.11	0.95
P.Chest-P.Ankle	0.46	0.84	0.53	0.61	0.63	0.61	0.52	0.51	0.83	0.53	0.43	0.70	0.35	0.56	0.15	0.96
P.hand-P.Ankle	0.56	0.80	0.51	0.62	0.62	0.62	0.42	0.53	0.80	0.54	0.36	0.66	0.37	0.58	0.12	0.96
Avg.	0.52	0.83	0.71	0.65	0.67	0.65	0.61	0.53	0.80	0.50	0.55	0.64	0.41	0.56	0.14	0.95

Table 7.4: Comparison of macro-F1 scores between ContrasGAN, *shift*-GAN and baseline techniques on accelerometer datasets.

Task	Lower bound	ContrasDGAN	shift-GAN	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	FLDA (2016)	JDA (2014)	GFK (2012)	IW (2012)	TCA (2011)	CCA (2001)	Upper bound
RA-LA	0.67	0.90	0.89	0.67	0.73	0.53	0.77	0.50	0.64	0.50	0.53	0.70	0.43	0.66	0.17	0.94
RL-LL	0.79	0.89	**0.94**	0.56	0.82	0.52	0.71	0.50	0.74	0.52	0.52	0.65	0.50	0.59	0.14	0.97
RA-T	0.44	0.89	0.89	0.55	0.66	0.50	0.77	0.50	0.52	0.44	0.50	0.48	0.36	0.50	0.11	0.95
H-C	0.36	0.64	0.55	0.56	0.56	0.32	0.64	0.52	0.49	0.54	0.32	0.65	0.47	0.52	0.18	0.88
LA-RA	0.52	0.80	0.65	0.55	0.55	0.66	0.49	0.48	0.81	0.69	0.43	0.53	0.51	0.46	0.15	0.91
LL-RL	0.52	0.87	0.70	0.62	0.64	0.68	0.51	0.50	0.82	0.41	0.72	0.58	0.37	0.49	0.12	0.89
T-RA	0.26	0.85	0.61	0.57	0.58	0.59	0.52	0.49	0.81	0.34	0.79	0.53	0.36	0.44	0.10	0.97
C-H	0.28	0.82	0.48	0.61	0.63	0.63	0.53	0.49	0.82	0.35	0.41	0.60	0.36	0.47	0.13	0.91
RA-RL	0.25	0.80	0.62	0.62	0.61	0.64	0.52	0.48	0.78	0.41	0.34	0.63	0.37	0.49	0.11	0.88
LA-LL	0.40	0.78	0.73	0.64	0.69	0.62	0.55	0.51	0.81	0.69	0.41	0.64	0.47	0.48	0.18	0.86
LA-T	0.43	0.81	0.67	0.65	0.64	0.65	0.61	0.46	0.80	0.52	0.34	0.61	0.51	0.50	0.15	0.88
P.Ankle-P.Chest	0.30	0.81	0.69	0.66	0.55	0.60	0.48	0.47	0.80	0.41	0.72	0.64	0.36	0.55	0.12	0.72
P.Ankle-P.hand	0.28	0.73	0.61	0.79	0.70	0.63	0.37	0.50	0.85	0.34	0.79	0.66	0.37	0.54	0.10	0.83
P.Chest-P.Ankle	0.41	0.84	0.43	0.59	0.54	0.61	0.38	0.46	0.82	0.50	0.41	0.64	0.34	0.48	0.13	0.75
P.hand-P.Ankle	0.50	0.73	0.41	0.60	0.55	0.62	0.38	0.48	0.80	0.51	0.34	0.64	0.31	0.56	0.11	0.84
Avg.	0.43	0.81	0.66	0.62	0.63	0.59	0.55	0.49	0.75	0.48	0.50	0.61	0.41	0.52	0.13	0.88

Tables 7.3 and 7.4 report the micro-F1 and macro-F1 scores of ContrasGAN and *shift*-GAN and the baseline techniques on *cross-body* experiments. ContrasGAN and *shift*-GAN have achieved a performance improvement in micro-F1 scores of 60% and 36% and in macro-F1 scores of 90% and 54% over the lower bound accuracy, respectively.

On average, the performance improvement of ContrasGAN and *shift*-GAN in terms of micro-F1 scores is 77% and 55% over non-deep learning-based techniques respectively. In terms of macro-F1 scores, the improvement is even much better of 84% and 49% over non-deep learning based techniques, respectively. Compared to adversarial domain adaptation techniques, ContrasGAN shows an improvement of 28% and 34% in micro-F1 and macro-F1 scores respectively. In comparison, *shift*-GAN has an improvement of 9% for both micro-F1 and macro-F1 scores.

As expected, the classification performance is poor for the state-of-the-art methods. It is evident that GFK, TCA and JDA techniques show better performance than CCA, FLDA and IW but they do not outperform the other techniques. The performance of TCA and JDA is more stable and better than the performance in the previous sensor data experiment. This might be because we are trying to align two different domains with different and similar body parts. Thus, this suggests that the *similarity* between the source and target domain is important for a successful cross-domain learning for classic transfer learning methods. However, our results suggest that ContrasGAN and *shift*-GAN can perform a suitable domain adaptation when using different and similar body parts at the same time across different datasets.

ContrasGAN obtains the highest averaged micro-F1 and macro-F1 scores: 83% and 81%, which is 5% and 7% 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 *shift*-GAN with 12% and 15% in micro-F1 and macro-F1 scores, demonstrating the effectiveness of contrastive learning in capturing discriminative transfer features between classes.

Figures 7.5, 7.6, 7.7, 7.8 present confusion matrices on ContrasGAN, *shift*-GAN, DANN, and DAN. *shift*-GAN, DANN, and DAN struggle to distinguish similar activities, for example, sitting, standing, lying back and lying side, and walking on a treadmill at different speeds. ContrasGAN

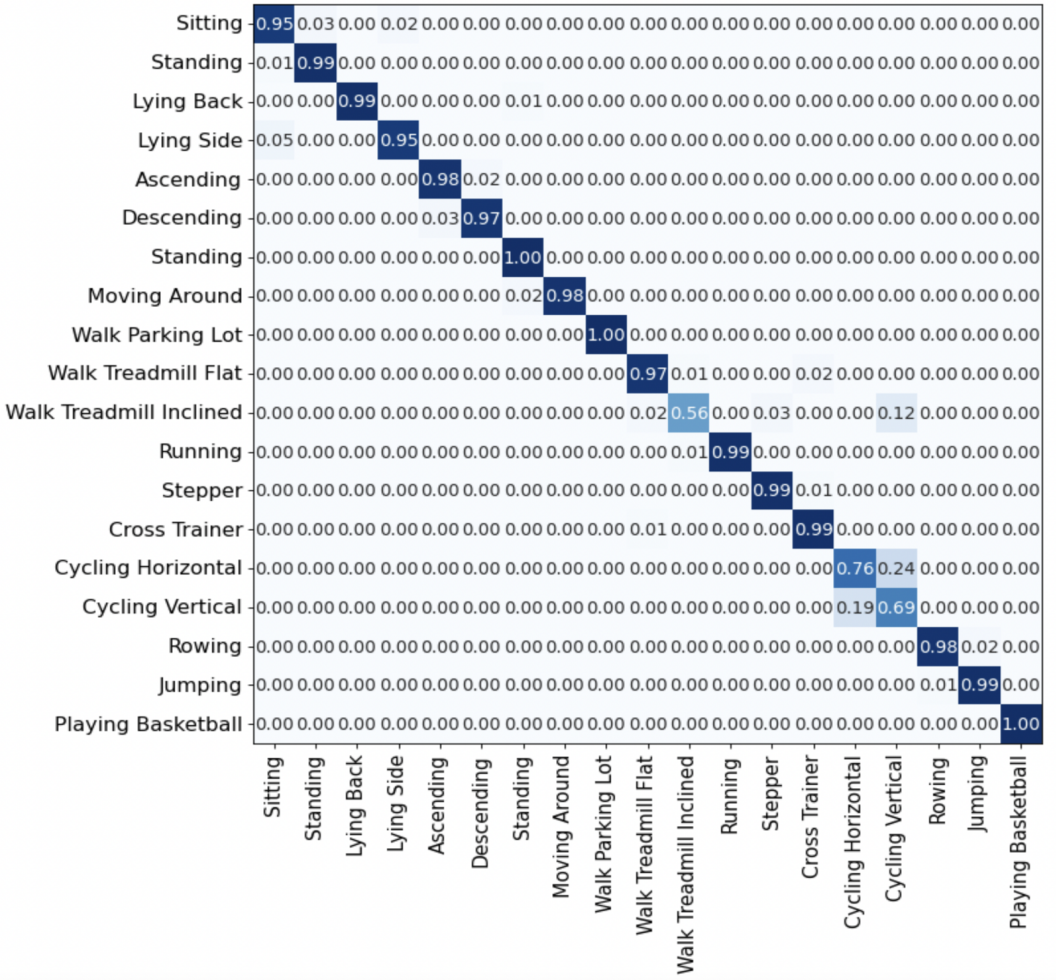


Figure 7.5: ContrasGAN

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. *shift*-GAN, DAN, JAN (a variation of DAN), and ADADM have achieved better overall performance. *shift*-GAN is also built on Bi-GAN and extends it with a kernel mean matching technique to align transformed target data with the real target data. Both ContrasGAN and *shift*-GAN are significantly 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

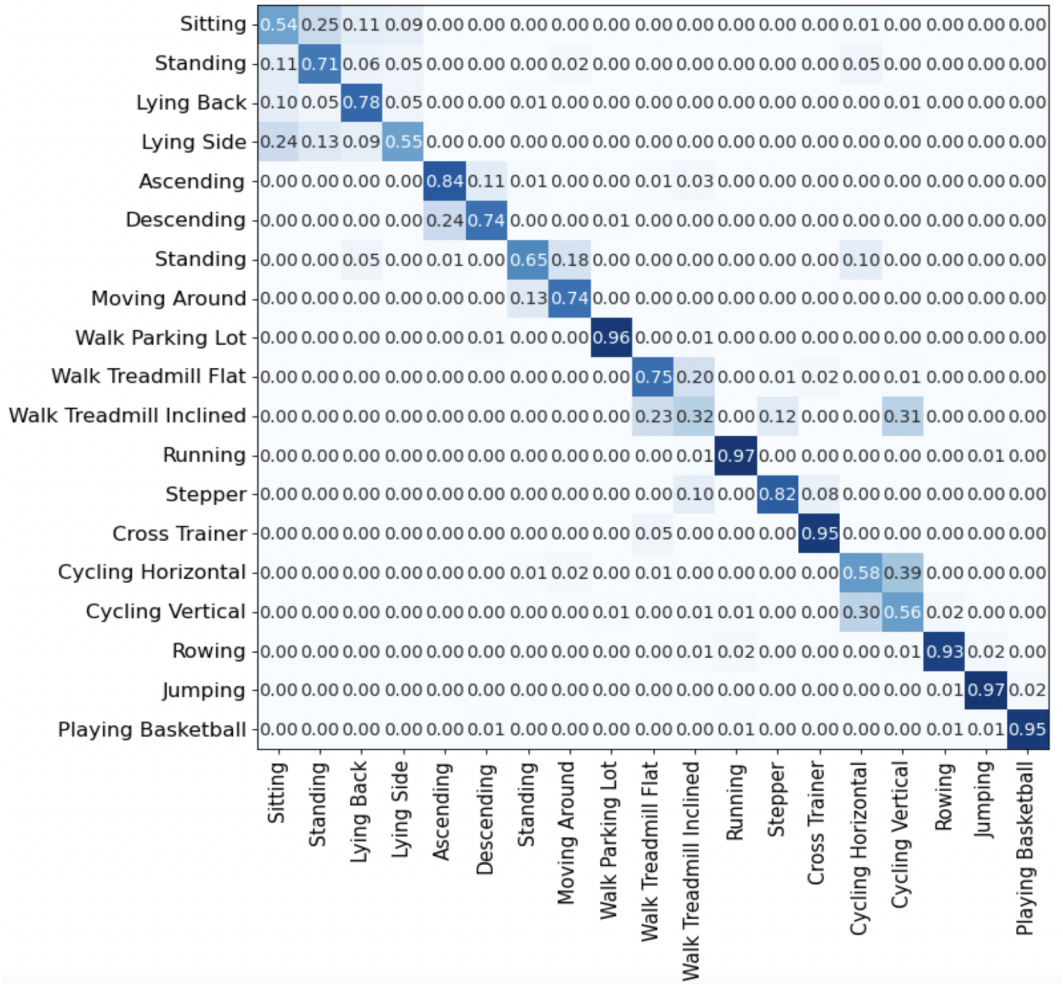


Figure 7.6: *shift*-GAN

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 the techniques do not perform well. DANN has achieved the highest micro-F1 score 72%, which is 8% higher than ContrasGAN. However, its macro-F1 score is only 49%, 14% lower than ContrasGAN. This suggests that its learning prioritises the majority classes. On the other hand, ContrasGAN, DANN and GFK can achieve more balanced accuracy across all the classes.

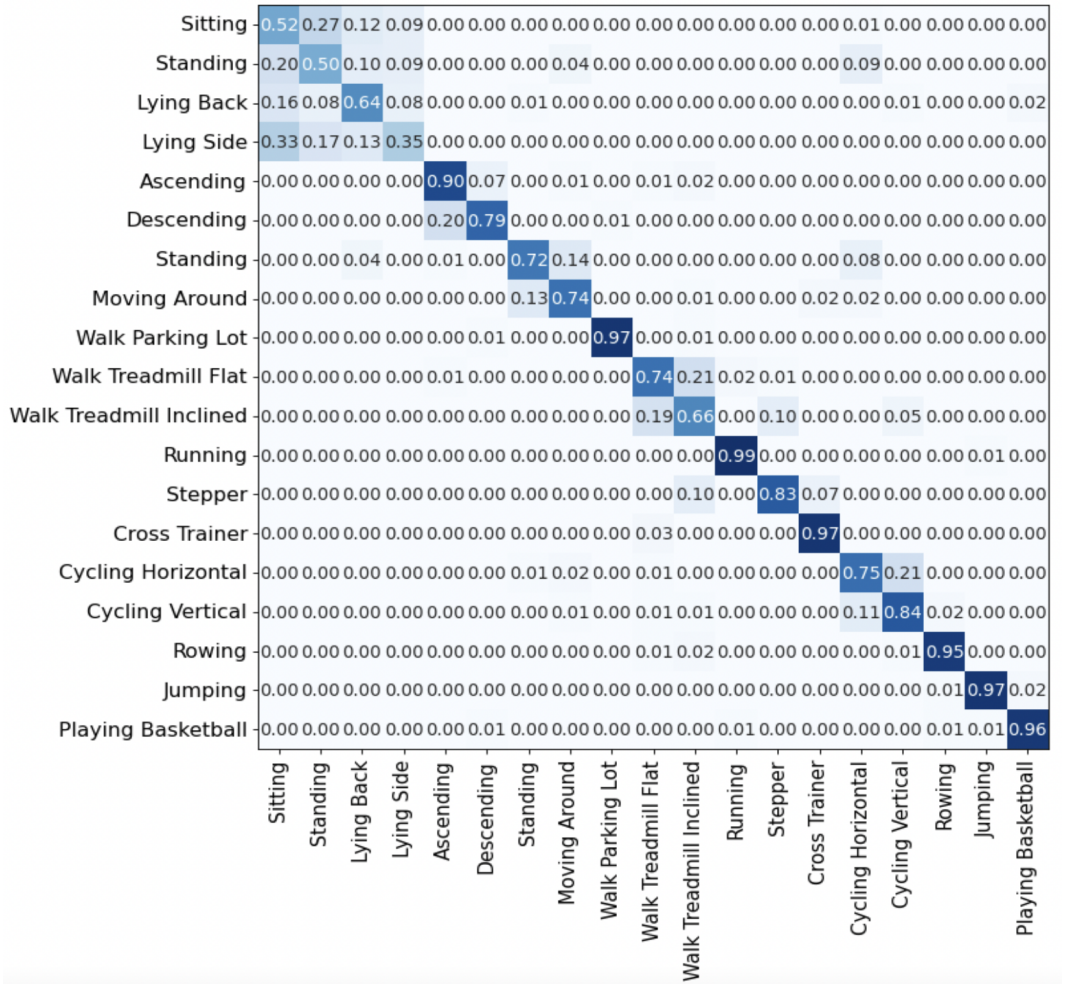


Figure 7.7: DAN

7.2.2.2 Cross-sensor Experiments

For *cross-sensor* experiments, we measure the accuracy of transferring an activity model from different types of devices. 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 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.

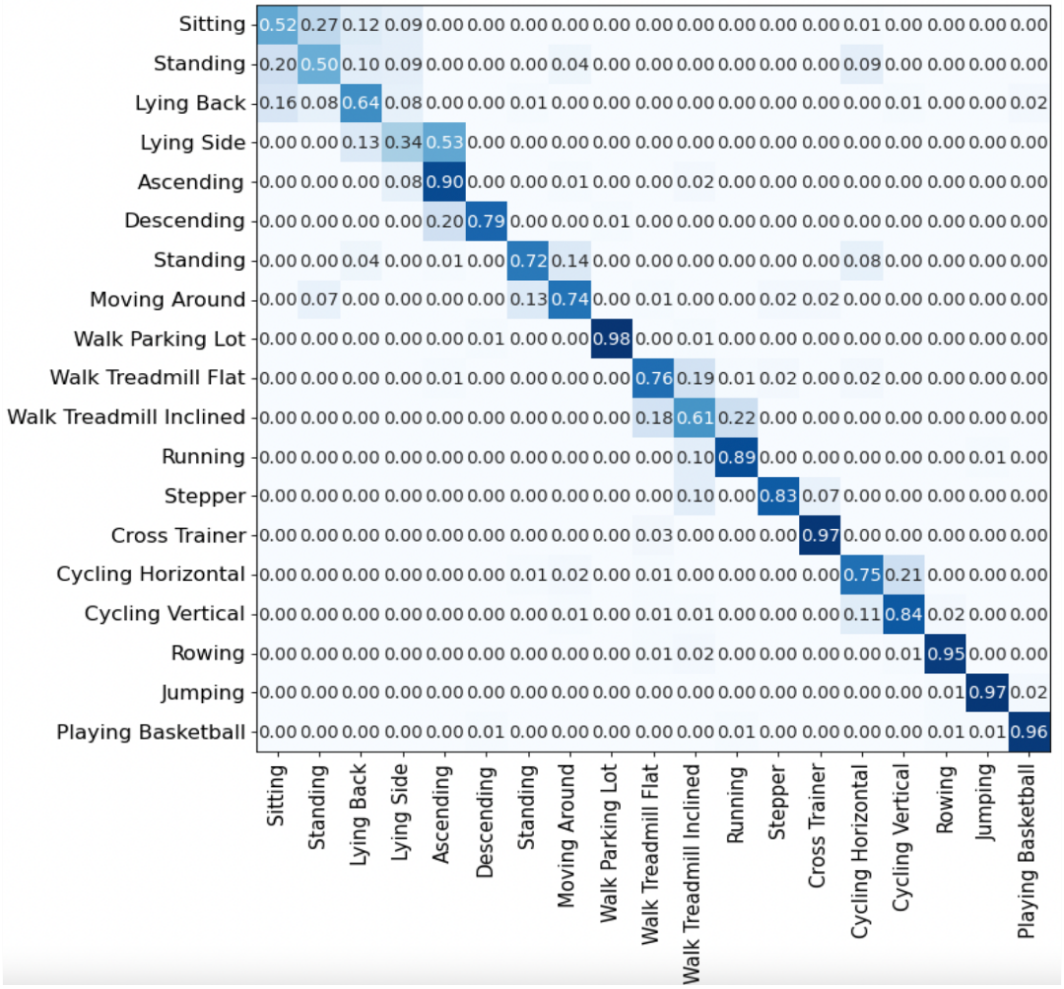


Figure 7.8: DANN

Table 7.5: Comparison of micro-F1 scores between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer datasets.

Task	ContrasDGAN	shift-GAN	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	GFK (2012)	Upper bound
WISDM-PAMAPS2	0.8	0.67	0.85	0.63	0.36	0.78	0.79	0.77	0.38	0.92
PAMAPS2-WISDM	0.88	0.80	0.86	0.51	0.2	0.84	0.9	0.89	0.47	0.98
WISDM-DSADS	0.8	0.73	0.79	0.73	0.63	0.74	0.92	0.89	0.44	0.86
DSADS-WISDM	0.8	0.84	0.79	0.74	0.64	0.86	0.88	0.77	0.63	1
PHONE-WATCH	0.68	0.71	0.55	0.49	0.15	0.46	0.42	0.38	0.33	0.81
WATCH-PHONE	0.72	0.82	0.61	0.59	0.16	0.64	0.46	0.63	0.57	0.92
W.WATCH-PAMAP	0.84	0.63	0.81	0.61	0.41	0.8	0.74	0.74	0.57	0.95
PAMAP-W.WATCH	0.85	0.72	0.87	0.69	0.57	0.81	0.74	0.82	0.61	0.92
W.WATCH-DSADS	0.84	0.73	0.79	0.68	0.53	0.71	0.71	0.62	0.61	0.92
DSADS-W.WATCH	0.89	0.71	0.79	0.69	0.55	0.7	0.72	0.76	0.65	0.97
Avg.	0.81	0.73	0.77	0.64	0.42	0.73	0.73	0.73	0.53	0.93

Table 7.6: Comparison of macro-F1 scores between ContrasGAN, and *shift*-GAN and baseline techniques on accelerometer datasets.

Task	ContrasDGAN	shift-GAN	ADADM (2020)	DADA (2019)	JAN (2017)	DANN (2016)	DeepCORAL (2016)	DAN (2015)	GFK (2012)	Upper bound
WISDM-PAMAPS2	0.8	0.62	0.82	0.51	0.2	0.77	0.75	0.76	0.26	0.92
PAMAPS2-WISDM	0.85	0.65	0.84	0.35	0.07	0.82	0.89	0.89	0.37	0.98
WISDM-DSADS	0.79	0.72	0.75	0.58	0.56	0.61	0.88	0.74	0.33	0.86
DSADS-WISDM	0.75	0.85	0.7	0.68	0.56	0.7	0.73	0.76	0.55	1
PHONE-WATCH	0.63	0.68	0.51	0.44	0.11	0.44	0.41	0.36	0.2	0.8
WATCH-PHONE	0.7	0.79	0.56	0.54	0.11	0.62	0.45	0.63	0.48	0.92
W.WATCH-PAMAP	0.84	0.60	0.81	0.58	0.26	0.76	0.73	0.71	0.52	0.92
PAMAP-W.WATCH	0.85	0.69	0.81	0.62	0.52	0.81	0.71	0.71	0.6	0.9
W.WATCH-DSADS	0.84	0.67	0.73	0.62	0.51	0.63	0.7	0.55	0.6	0.9
DSADS-W.WATCH	0.89	0.65	0.75	0.68	0.58	0.63	0.71	0.76	0.62	0.95
Avg.	0.79	0.69	0.73	0.56	0.35	0.68	0.70	0.69	0.45	0.92

Tables 7.5 and 7.6 compare the micro-F1 and macro-F1 scores between ContrasGAN and the comparison techniques. ContrasGAN outperforms all these techniques and achieves the averaged micro-F1 and macro-F1 scores as 81% and 79%, which is higher than the second-best technique (ADADM) by 4% and 7% respectively. ContrasGAN also outperforms *shift*-GAN by 8% and 10% in micro-F1 and macro- F1 scores respectively.

Due to the heterogeneity in the feature space, we cannot run the lower-bound baseline and non-deep learning-based methods. We use an upper-bound baseline to indicate the challenge on the target data. As shown in Table 7.5, PHONE-WATCH produces the lowest upper-bound accuracy (81%). Most of the comparison techniques seem to struggle: for example, GFK and DAN only reach 2% and 36% in macro-F1 scores. ContrasGAN achieves the macro-F1 scores of 63%, 12% higher than ADADM, the second-best technique. When observing the difference between micro-F1 and macro-F1 scores, ContrasGAN has the smallest difference, indicating that it can better balance the majority and minority classes.

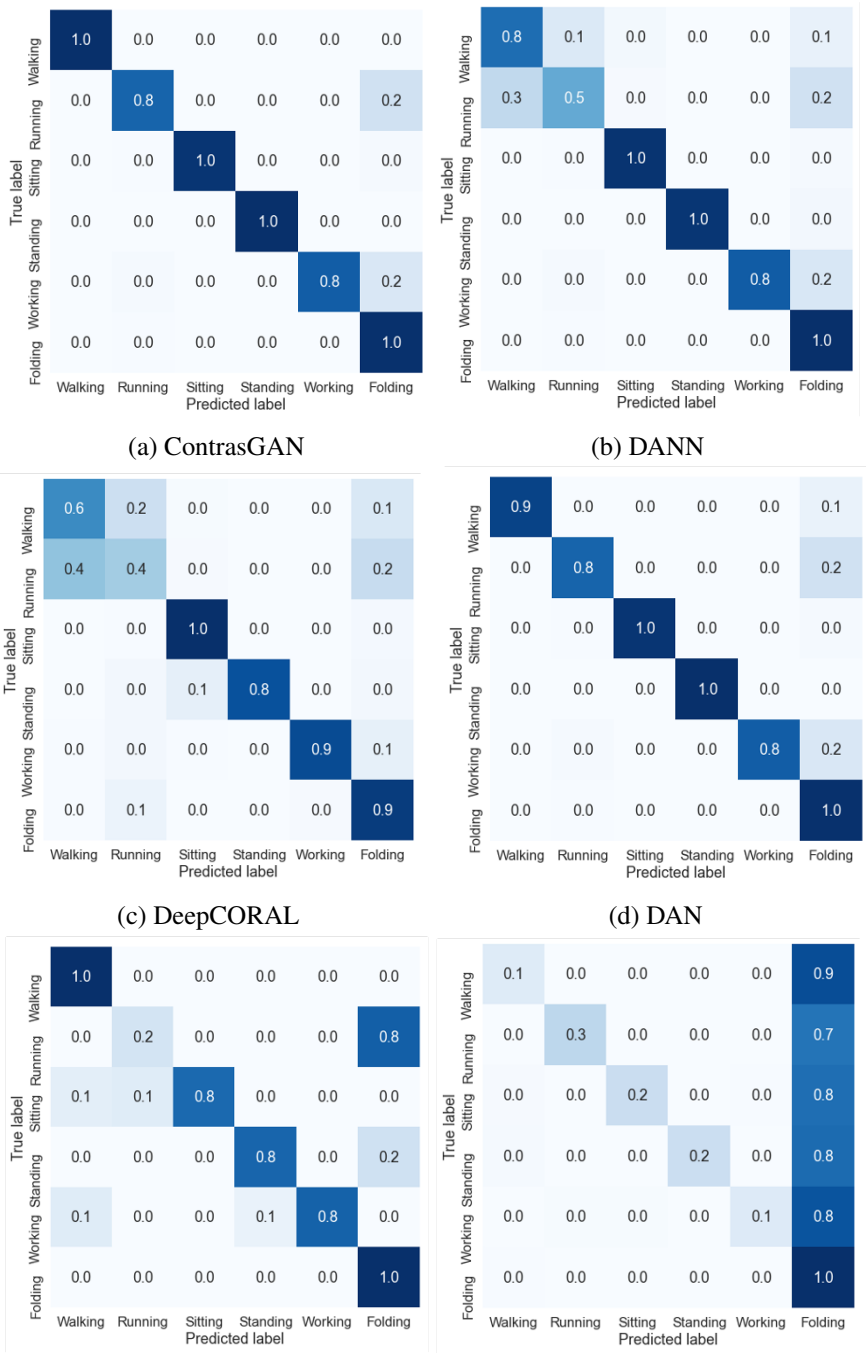
Figure 7.9 presents the confusion matrices on the PAMAP-W.PHONE task. First of all, ContrasGAN and DAN are better than DANN and DeepCORAL at differentiating similar activities, such as walking and running. Secondly, ADADM seems not stable and the performance might be affected by the mixup ratio. Thirdly, GFK cannot cope with high heterogeneity well and only predicts one activity.

7.3 Ablation, Stability and Convergence Study

The following sections present the stability and convergence of the proposed models. First, we assess the stability of UDAR by evaluating the importance of each component in its architecture. Then, for ContrasGAN and *shift*-GAN we record the loss during the training process over epochs to prove that both models converge well.

7.3.1 UDAR

In this section, we discuss how the design of each component in UDAR can impact its performance. We will start with the quality of pre-annotation steps, and then assess the advantage of UDAR over coarse-grained feature remapping (i.e., mapping feature spaces only with semantics).



(a) ContrasGAN (b) DANN

(c) DeepCORAL (d) DAN

(e) ADADM (f) GFK

Figure 7.9: Confusion matrices on the PAMAP-W.PHONE task

Quality of Pre-Annotation. The quality of pre-annotating has an important role in achieving effective domain adaptation, as the feature spaces are aligned based on whether they have the same class label. Therefore, here we aim to find an approach to achieve high accuracy in pre-annotating. To do so, we will look into how to select a classifier in generating accurate pseudo labels.

We experiment with a collection of the base classifiers, including Random Forest (RF), Support Vector Machine with RBF Kernel (SVM), and k Nearest Neighbors (kNN), and two ensemble approaches on the three base classifiers: Majority Voting (MV) [169] and Stacked Ensemble (SE). For each of them, we train the classifier with the source domain dataset, and predict activity labels on the knowledge-transferred target domain dataset. Then we select the predictions with high confidence (e.g., the confidence score is greater than 80%) and compare them with the true labels to calculate the pre-annotation accuracy.

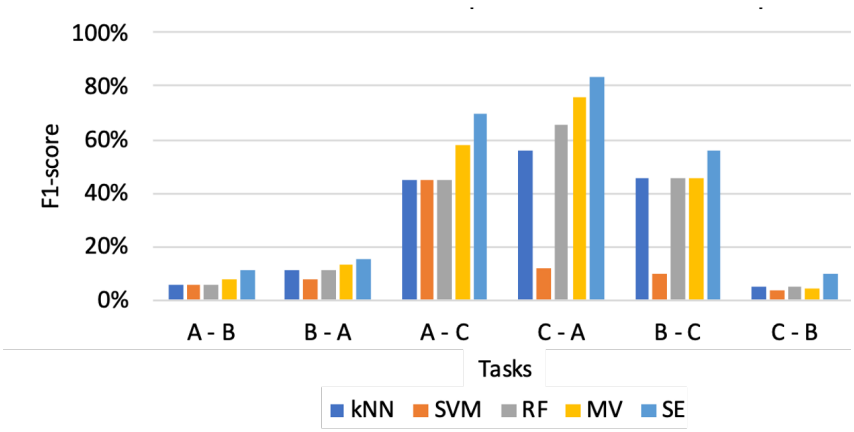


Figure 7.10: Comparison of micro-F1 scores in the pre-annotation step between SVM RBF, kNN, RF, MV and SE. The SE outperforms the other techniques and is selected as the technique for pre-annotating.

Figure 7.10 presents the micro-F1 scores of pre-annotation with the above techniques. The results demonstrate that the stacked ensemble achieves the highest accuracy in pre-annotation, with an improvement of 11% over RF, 30% over SVM, 12% over kNN, and 7% over MV. In the experiments from House B to C and from House C to A, SVM performs the worst compared to RF and kNN. The reason is that the datasets we are using are imbalanced and sensor features between activities can have subtle difference; *e.g.,* showering and toileting, and having breakfast and drinking. This problem has made the base classifiers and majority voting approaches struggle in differentiating activities with less distinctive patterns. During the majority process, we face

the problem that most of the time the classifiers will 'agree' on the same label, meaning that we will not have uncertain instances to re-annotate later on.

In the experiment from House C to A, MV achieves the same performance as the base classifiers. In most cases, the base classifiers seem to fail to find meaningful similarities across different datasets, when the datasets are much noisier. For example, the House B and C datasets are very noisy in that the activity annotation is not accurate [80] and sensor activation is unexpected for a certain activity [188]. Also, these two datasets have imbalanced class distribution; *e.g.*, House C only has 6 instances of the 'Drinking' activity. Due to these problems, the experiment results with A-B, A-C, and C-B are worse than the others. In the experiment from House C to A, MV achieves the same performance as the base classifiers while SE outperforms. In the end, we consider the stacked ensemble technique as an ideal choice to achieve high quality of generated pseudo labels.

Impact of Confidence Thresholds in Pre-annotation. As we mentioned before, the accuracy of pre-annotations can have a significant impact on the re-annotation process. To evaluate the impact, we control the confidence threshold from 50% to 85% with step size 5%, and select the target instances for the re-annotation process only when their prediction confidence is higher than the threshold. Figure 7.11 compares the micro-F1 scores of domain adaptation on different thresholds with different domain adaptation techniques on selected tasks. Again, we can see that UDAR significantly outperforms these comparison techniques on different threshold settings.

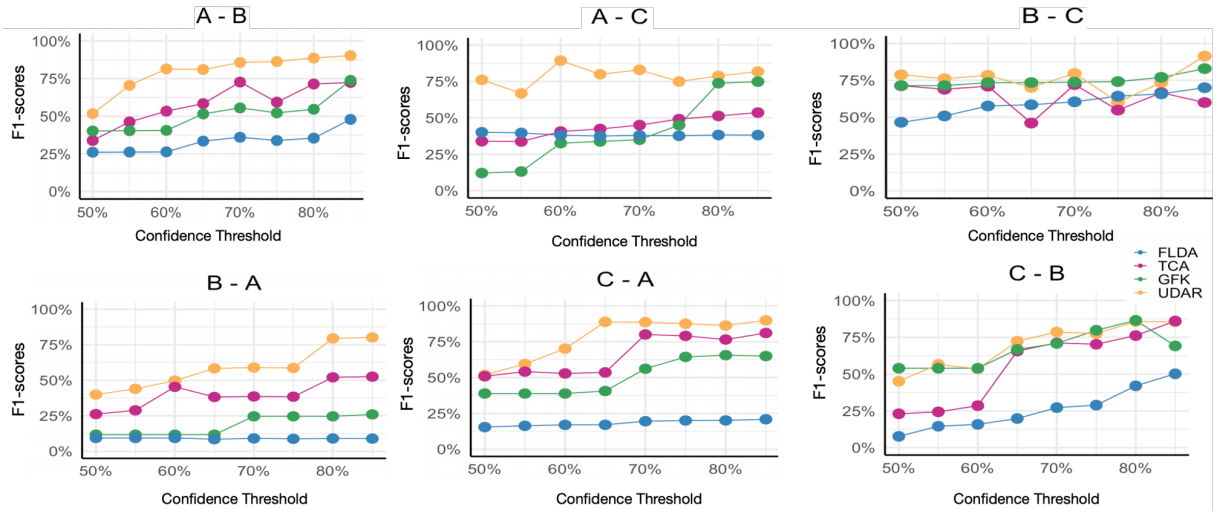


Figure 7.11: Comparison of the impact of confidence thresholds on domain adaption accuracy.

The lower the confidence threshold, the worse UDAR performs. If we set up a threshold

lower than 50% all classifiers will 'agree' on the majority class label, and we will have very few or no uncertain instances to re-annotate later on. On tasks A-B and C-A, we observe that the accuracy of TCA and GFK drops when the confidence increases, because during the pre-annotation process, most of the instances, if not all, are classified as uncertain.

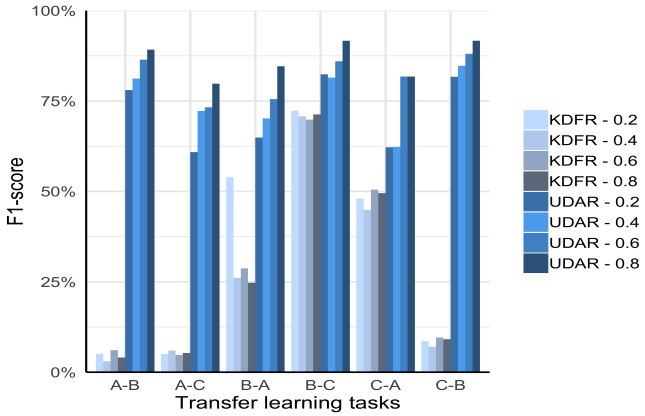


Figure 7.12: Comparison of micro-F1 scores between KDFR and VAE.

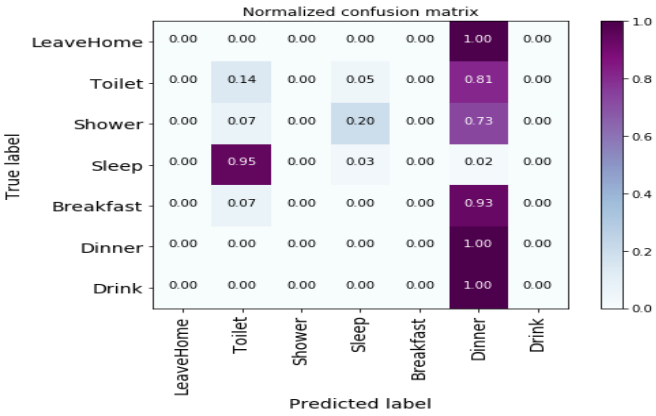


Figure 7.13: Confusion matrix of KDRF on A-B with 80% training data.

Comparison with Coarse-grained Feature Alignment. Aligning features from the two domains based on the sensor ontologies is intuitive and acts as a good baseline to see what additional benefit that VAE-based fine-grained alignment adds to our approach. We compare activity recognition accuracy between UDAR and knowledge-driven feature remapping (KDFR). Knowledge-driven feature remapping is to map sensor features based on the sensor semantics; that is, where they are deployed and which objects they are attached to. We train a stacked ensemble on a percentage p of the source domain dataset, predict labels on the target domain dataset, and evaluate the prediction accuracy. Figure 7.16 compares the micro-F1 scores between

UDAR and KDFR with different training data percentages. The label 'KDFR - 0.2' means that we predict the labels on target domain data that is transformed via KDFR alone using the stacked ensemble that is trained with 20% of the target domain dataset. We observe that UDAR achieves much better micro-F1 scores than KDFR during the pre-annotation step. This advantage is especially seen in transferring tasks A-B, A-C, and C-B, where the micro-F1 score during the pre-annotation step is lower than 15% and the performance improvement is over 50%.

These results demonstrate that the fine-grained feature space alignment and the pre-annotation process can significantly improve the performance. In terms of the low accuracy on KDFR, during the pre-annotation process, the classifiers on KDFR struggle in finding meaningful similarities between instances in the source and target domains. For example, we can see this from Figure 7.14, where all classifiers achieve very low accuracy in the tasks of A-B and C-B compared to the other tasks. This leads to a significant distribution differences between domains and increases the transferring complexity. Furthermore, Figure 7.17 presents the confusion matrix on the A-B task, where the classifier is biased towards one class; that is, the KDRF classifies most of the activities as 'dinner'. This activity activates seven sensors, more than the other activities that fire at most 4 sensors. When few sensors are activated, the original feature representation is more sparse. With knowledge remapping, the representations will not be sparse anymore as each sensor in one dataset can be mapped to a collection of sensors in the other dataset even with low similarity scores. This adds noise to the knowledge-remapped representations and decreases the performance of the classifier.

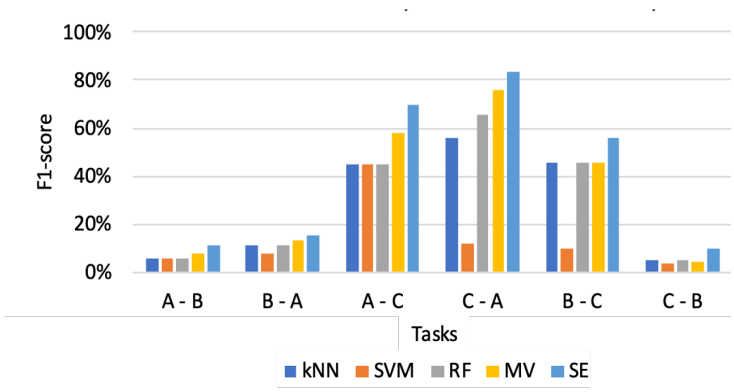


Figure 7.14: Comparison of micro-F1 scores in the pre-annotation step between SVM RBF, kNN, RF, MV and SE.

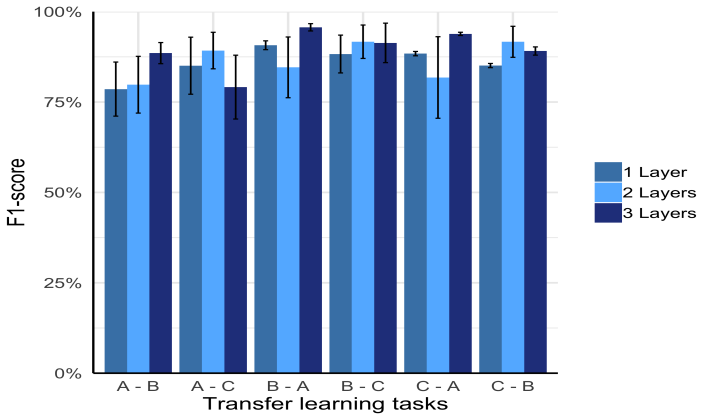


Figure 7.15: Comparison of micro-F1 scores on VAE with different number of layers.

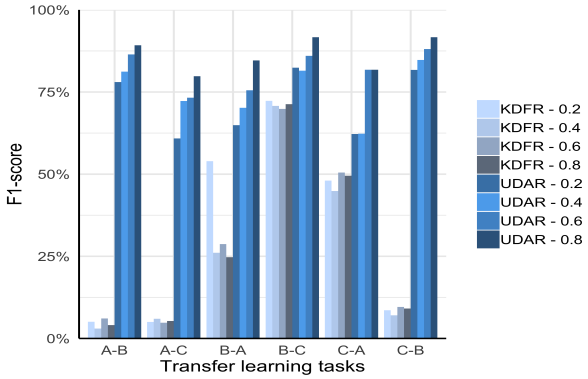


Figure 7.16: Comparison of micro-F1 scores between KDFR and VAE.

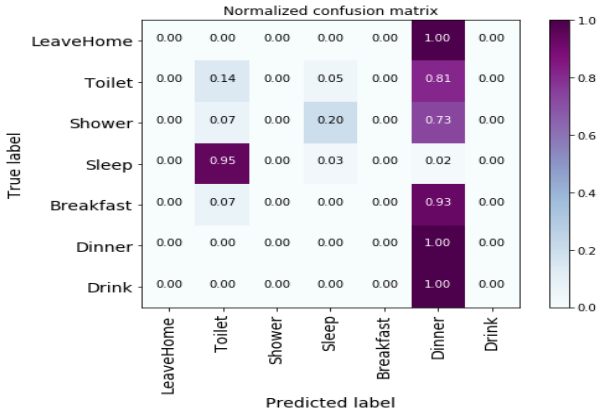
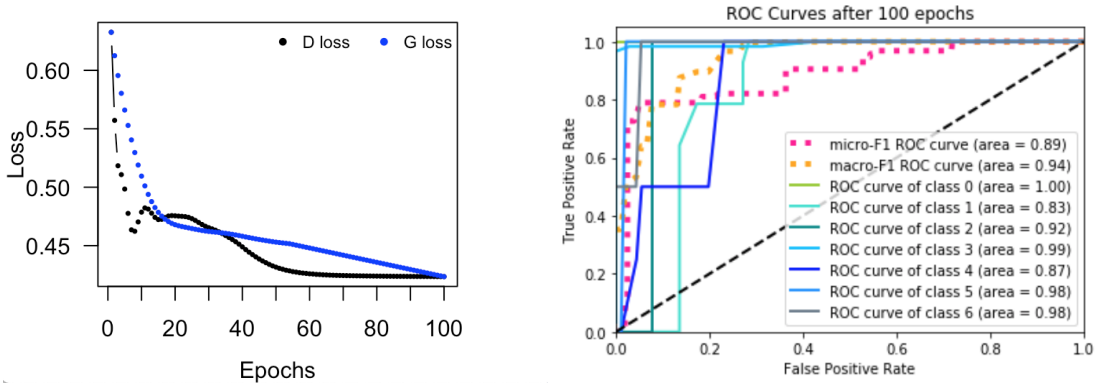


Figure 7.17: Confusion matrix of KDRF on A-B with 80% training data.

7.3.2 *shift*-GAN



(a) Loss on B-C (b) ROC on B-C

Figure 7.18: Loss performance and ROC curves for tasks B-C during training

Stability and difficulty to converge are two classic problems in GAN. These two problems can be a significant concern for unsupervised domain adaptation. Since we do not use any labels on the target domain, domain adaptation performance relies on stable feature space transformation between two domains. For this purpose, we have recorded the loss on both generators and discriminators over epochs and compared domain adaptation performance in ROC curves.

Figure 7.18 contains two plots of the performance of the discriminator and generator during the training process for task B-C. The discriminator and generator are on the primary GAN; i.e., from source to target domain. We can see that in our experiments GAN has converged well on sensor data.

We choose SVM classifier after evaluating the performance of different classifiers on 6 learning tasks: k Nearest Neighbors (kNN), Support Vector Machine with RBF Kernel (SVM), Random Forest (RF), two Neuronal Networks (NN) with different number of layers and parameters, and Naïve Bayes (NB). We train each classifier with the source domain data and we predict labels using different percentages of target domain data. We varied the percentage of testing data from 10% to 80% with a step of 10%. The results are shown in Table 7.3.2.

7.3.3 ContrasGAN

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. First,

Table 7.7: Comparison of the performance of different classifiers in binary sensor data.

Task	kNN		SVM		RF		NN - 2 Layers		NN - 3 Layers		NB	
	Micro-F	Macro-F	Micro-F	Macro-F	Micro-F	Macro-F	Micro-F	Macro-F	Micro-F	Macro-F	Micro-F	Macro-F
A - B	0.680	0.445	0.680	0.547	0.690	0.544	0.610	0.415	0.640	0.474	0.680	0.516
B - A	0.840	0.716	0.850	0.722	0.790	0.644	0.850	0.734	0.840	0.729	0.550	0.257
A - C	0.843	0.613	0.833	0.667	0.735	0.495	0.892	0.645	0.912	0.663	0.500	0.204
C - A	0.765	0.539	0.770	0.511	0.657	0.461	0.696	0.446	0.726	0.499	0.245	0.266
B - C	0.756	0.528	0.767	0.641	0.667	0.556	0.767	0.642	0.789	0.698	0.189	0.170
C - B	0.711	0.580	0.700	0.650	0.656	0.493	0.689	0.543	0.667	0.438	0.667	0.464
Average	0.766	0.570	0.767	0.623	0.699	0.532	0.751	0.571	0.762	0.583	0.472	0.313

we show the importance of each type of new loss *via* an ablation study. Figure 7.19 presents the macro-F1 scores of the ablation experiments on a subset of tasks.

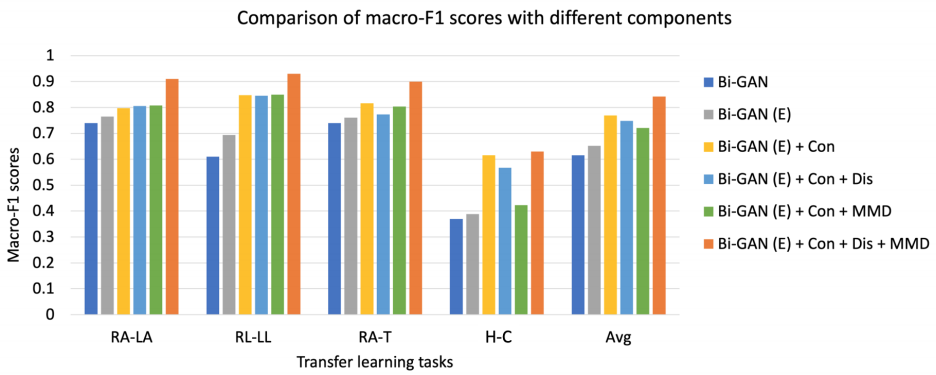
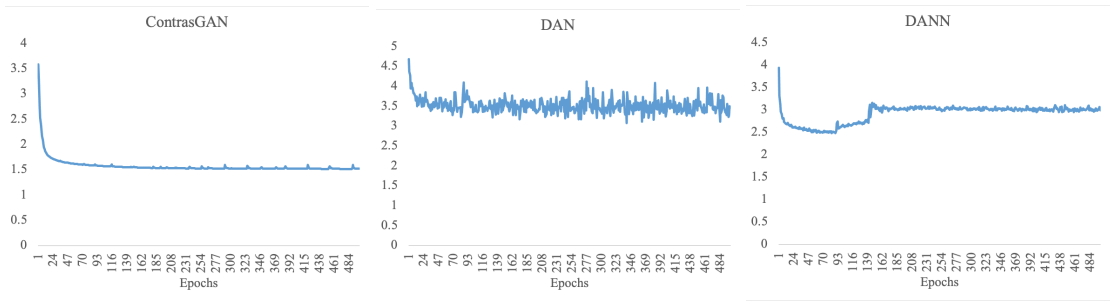


Figure 7.19: Ablation study of ContrasGAN

Based on these results, we can make the following observations. Firstly, contrastive loss significantly boosts performance, especially on task H-C, where it outperforms Bi-GAN with 25% in macro-F1 scores. This indicates that contrastive loss helps to learn discriminative features and improve class-level recognition accuracy. Secondly, the combination of contrastive, discrepancy, and MMD loss makes significant improvement, from 65% on Bi-GAN with expectation loss to 84% on ContrasGAN in the averaged macro-F1 scores. It suggests that the discrepancy and MMD loss make a contribution to transforming the global feature spaces during the fine alignment stage. Thirdly, the expectation loss, discrepancy loss, and MMD loss each individually improve the performance on these transfer learning tasks only to a certain degree.

Figure 7.20 presents the loss plots of ContrasGAN for class-level alignment, DAN, and DANN. This shows that ContrasGAN can converge smoothly and stably.



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

Figure 7.20: Comparison of loss between ContrasGAN, DAN, and DANN

7.3.4 Training Time

Training a domain adaptation model can be a challenging and expensive task depending on the size of the datasets. In Figure 7.21 we compare the training time between UDAR, *shift*-GAN and ContrasGAN and the baseline techniques for binary sensor data. Training time is similar between tasks; therefore, we only present the results on task A-B for detailed comparison. The average computational time among all techniques is 11.4 minutes. We can see that UDAR, *shift*-GAN and ContrasGAN are not computationally expensive techniques - just 2 minutes above the average. Among the deep learning techniques, DeepCORAL has the smallest training time while ADADM is 10 minutes more than the average.

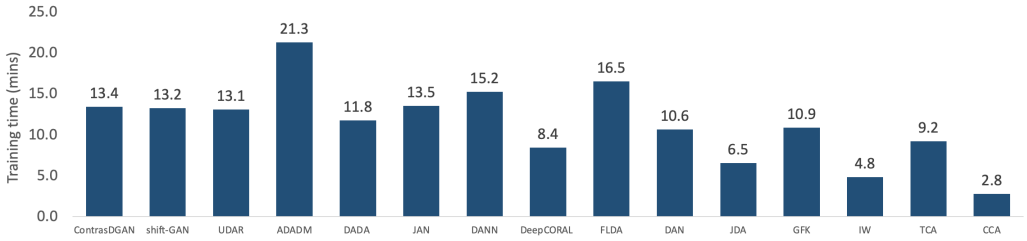


Figure 7.21: Comparison of training time between UDAR, *shift*-GAN and ContrasGAN and other techniques on task A-B.

Now in Figure 7.22 we compare the average training time for more complex tasks: RA-LA, RL-LL, RA-T and H-C. *shift*-GAN 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 H-C task is 1784 mins,

nearly 50 times the training time on ContrasGAN and significantly higher than all the other techniques.

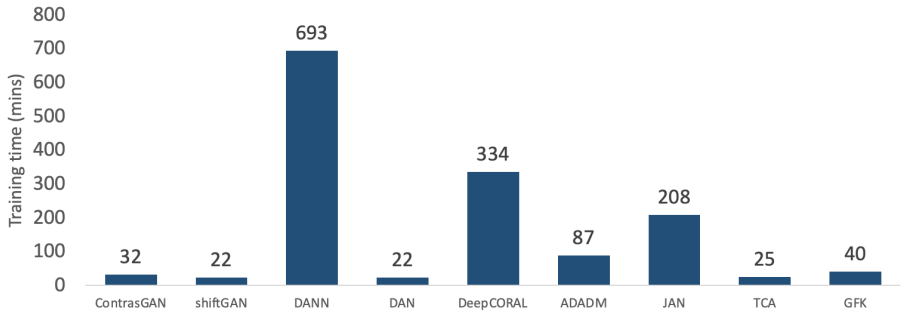


Figure 7.22: Comparison of training time between ContrasGAN and other techniques.

7.4 Impact of Training Data

We also assess the impact of training data on the effectiveness of domain adaptation. It is desirable to use less training data while achieving comparable accuracy. Therefore, in this experiment, we vary the percentage of training data in the target domain from 20% to 80% and assess the impact of the training data on the accuracy of domain adaptation.

7.4.1 Binary Sensor Data

In Figure 7.23 we average the accuracy across all tasks and all the training percentages. ContrasGAN outperforms all domain adaptation techniques. More specifically, the improvement of ContrasGAN in micro-F1 score over *shift*-GAN, UDAR, DAN, DANN, ADADM, TCA and GFK is 8%, 3%, 4%, 3%, 13%, 31% and 21%, respectively. We can also see that UDAR shows a similar performance to DAN and outperforms ADADM, TCA and GFK. *shift*-GAN achieves better micro-F1 than TCA, GFK and ADADM in 23%, 13% and 4%, respectively. However, shift-GAN performs worse than UDAR, DAN, and DANN in 5%, 5% and 5%, respectively, across all tasks and different training percentages.

Figures 7.24 and 7.25 compare the micro-F1 and macro-F1 scores of domain adaptation on six transfer learning tasks between ContrasGAN, *shift*-GAN and UDAR and the other techniques. The x-axis indicates the percentage of the training data. From the results, we observe that UDAR achieves better micro-F1 and macro-F1 scores across various learning tasks. UDAR and GFK

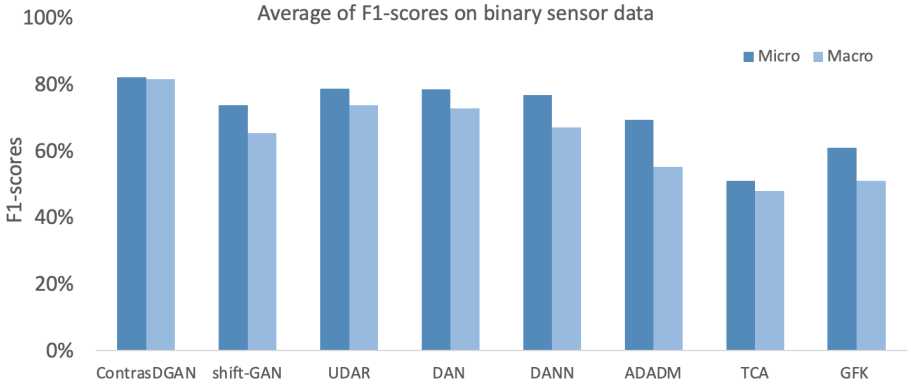


Figure 7.23: Average of micro-F1 and macro-F1 scores across all tasks over different training percentage.

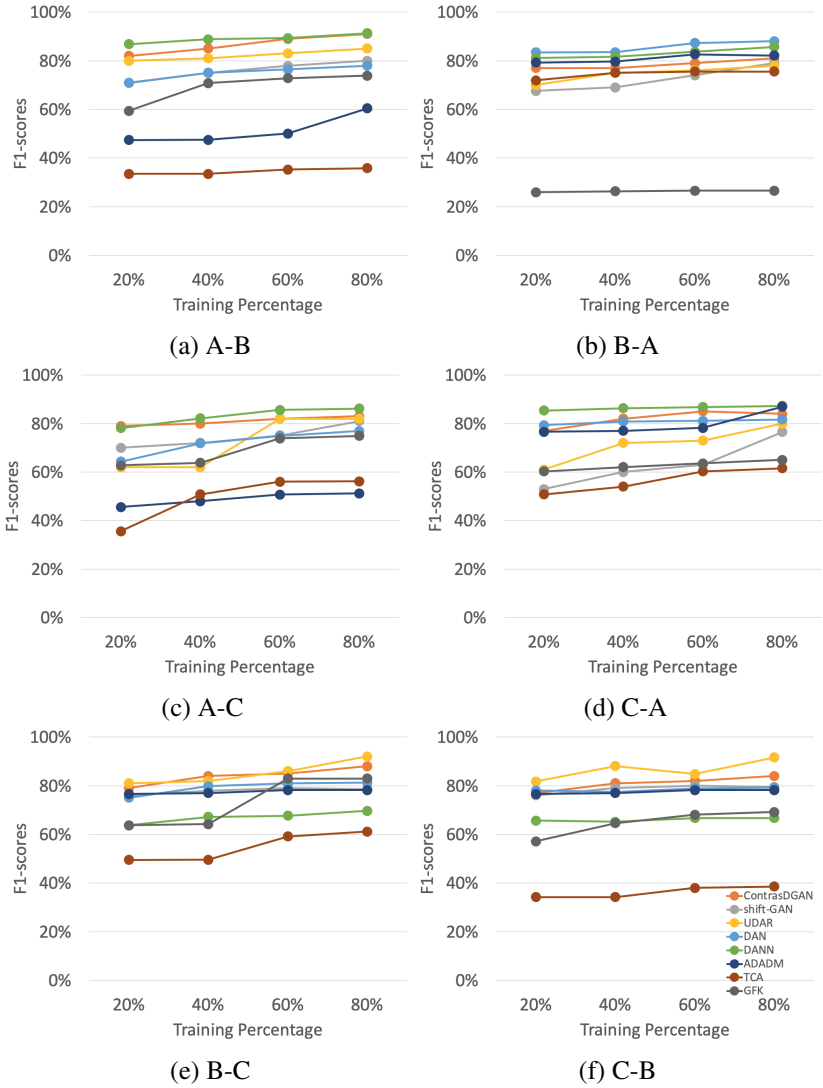


Figure 7.24: Comparison of micro-F1 scores (%) of domain adaptation between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary sensor data.

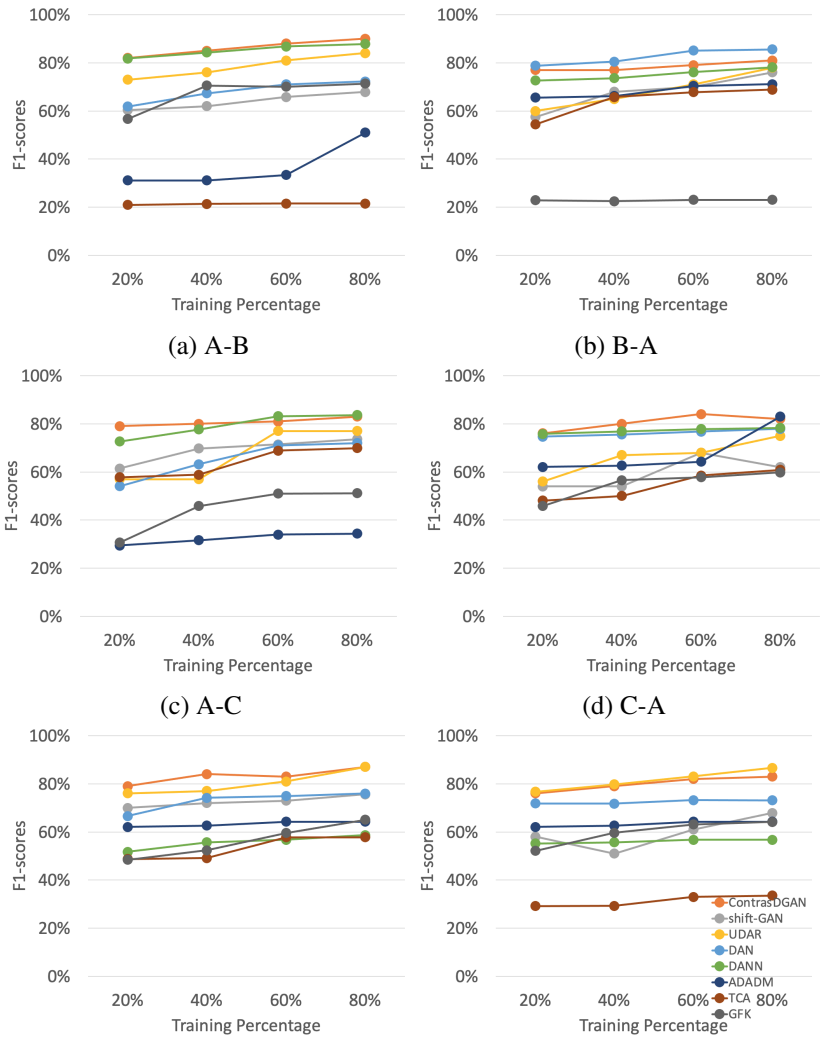


Figure 7.25: Comparison of macro-F1 scores (%) of domain adaptation between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary sensor data.

are stable and can achieve good domain adaptation independently of the percentage of training data. In contrast, TCA presents a higher variance especially on task C-B where both datasets are very noisy. GKF require expensive computation for subspace projection and hyper-parameter selection. This alignment becomes more difficult when the training dataset is small. ADADM seems to suffer from negative transfer, which sometimes produces lower accuracy, for example, tasks A-B and A-C. *shift*-GAN has a very stable performance except in task C-A where the accuracy decreases drastically with 20% of training data. This means that *shift*-GAN struggles in finding good discriminative features especially when little data is available and when the data is noisy.

Table 7.8: Comparison of average micro-F1 scores between ContrasGAN, *shift*-GAN and UDAR and baseline techniques on binary datasets across all training percentages.

Task	Lower bound	ContrasDGAN	shift-GAN	UDAR	DAN	DANN	ADADM	TCA	GFK	Upper bound
A-B	0.29	0.87	0.76	0.82	0.75	**0.89**	0.51	0.35	0.69	0.85
B-A	0.68	0.79	0.72	0.75	**0.86**	0.83	0.81	0.75	0.26	0.86
A-C	0.82	0.81	0.75	0.72	0.72	**0.83**	0.49	0.50	0.69	0.86
C-A	0.78	0.82	0.63	0.72	0.81	**0.86**	0.80	0.57	0.63	0.82
B-C	0.80	0.84	0.78	**0.85**	0.79	0.67	0.78	0.55	0.73	0.85
C-B	0.49	0.81	0.79	**0.87**	0.78	0.66	0.78	0.36	0.65	0.83

Table 7.9: Comparison of average macro-F1 scores between ContrasGAN, *shift*-GAN and UDAR and baseline techniques on binary datasets across all training percentages.

Task	Lower bound	ContrasDGAN	shift-GAN	UDAR	DAN	DANN	ADADM	TCA	GFK	Upper bound
A-B	0.23	**0.86**	0.64	0.79	0.68	0.85	0.37	0.21	0.67	0.84
B-A	0.62	0.79	0.68	0.69	**0.82**	0.75	0.68	0.64	0.23	0.86
A-C	0.81	**0.81**	0.69	0.67	0.65	0.79	0.32	0.64	0.45	0.84
C-A	0.74	**0.81**	0.60	0.67	0.76	0.77	0.68	0.54	0.55	0.81
B-C	0.78	**0.83**	0.73	0.80	0.73	0.56	0.63	0.53	0.56	0.85
C-B	0.47	0.80	0.59	**0.82**	0.72	0.56	0.63	0.31	0.60	0.82

In Tables 7.10 and 7.11 we average micro-F1 and macro-F1 scores across all training percentages for each task and we highlight the best micro-F1 and macro-F1 scores among all the domain adaptation techniques. ContrasGAN achieves the best macro-F1 scores on 4 out of 6 transfer tasks. The macro-F1 scores are 16% (*shift*-GAN), 8% (UDAR), 9% (DAN), 10% (DANN), 26% (ADADM), 33% (TCA) and 31% (GFK) lower than ContrasGAN. Although DANN performs better in tasks A-C (83% vs. 84%) when using 80% of the training data, ContrasGAN is more stable and outperforms DANN when the percentage of training data decreases. However in terms of micro-F1 scores, ContrasGAN is outperformed by DANN in C-A (86% vs. 82%) and B-A (83% vs. 79%) tasks, by DAN performs in B-A task (86% vs. 79%) and by UDAR in C-B task (87% vs. 81%). This means ContrasGAN achieves more balanced accuracy on each class than the other methods.

7.4.2 Accelerometer Sensor Data

In Figures 7.26 and 7.27 we present the micro-F1 and macro-F1 scores across different training percentages for each learning task. All techniques are more stable in the accelerometer experiments independently of the percentage of training data compared to the binary ones. This is probably due to the feature representations of the datasets. The binary sensor data collected in the in-the-wild real-world environments is more sparse, noisy, and imbalanced across classes while the acceleration sensor data curated in the controlled environments is balanced in class

distribution.

In Tables 7.10 and 7.11 we can see the average micro-F1 and macro-F1 scores across all training for each learning model. On average, ContrasGAN has the best classification accuracies despite the amount of training data. This indicates that contrastive learning can provide better class-discriminative features. The closest competitor to ContrasGAN is DAN, which performs 3% and 6% worst in micro-F1 and macro-F1 scores.

ContrasGAN performs best on 10 out of 15 learning tasks. Note that while ContrasGAN and DANN improve over the lower bound in task LA-RA, the other methods underperform. ADADM is very unstable and performs similar to non-deep learning base methods - the domain mix up adds noise and complexity to the transfer learning task.

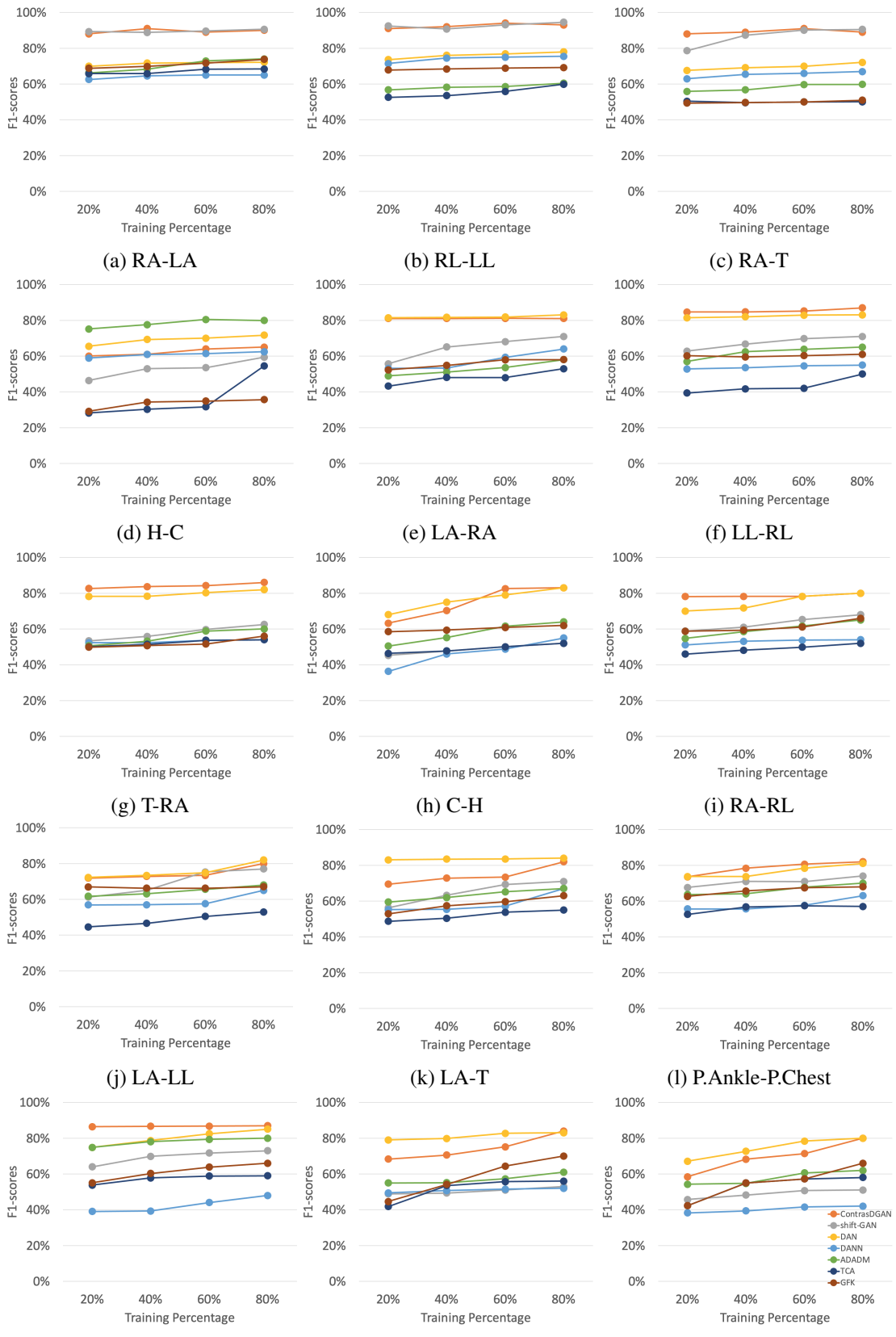
It is also interesting to point out that the improvement of DANN, TCA an GKF over the lower bound is limited by only 4%, 1% and 7%, respectively. This observation may direct some future work on studying under what situations deep learning and non-deep learning techniques can be used.

Table 7.10: Comparison of average micro-F1 scores between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer sensor datasets across all training percentages.

Task	Lower bound	ContrasDGAN	shift-GAN	DAN	DANN	ADADM	TCA	GFK	Upper bound
RA-LA	0.68	0.90	0.90	0.71	0.64	0.70	0.67	0.71	0.93
RL-LL	0.80	0.93	0.93	0.76	0.74	0.58	0.55	0.69	0.95
RA-T	0.46	0.89	0.87	0.70	0.65	0.58	0.50	0.50	0.94
H-C	0.39	0.63	0.53	0.69	0.61	0.78	0.36	0.34	0.83
LA-RA	0.67	0.81	0.65	0.82	0.57	0.53	0.48	0.56	0.93
LL-RL	0.58	0.85	0.68	0.82	0.54	0.62	0.43	0.60	0.89
T-RA	0.43	0.84	0.58	0.80	0.53	0.56	0.52	0.52	0.98
C-H	0.49	0.75	0.49	0.76	0.47	0.58	0.49	0.60	0.94
RA-LA	0.43	0.79	0.63	0.75	0.53	0.60	0.49	0.61	0.92
LA-LL	0.57	0.75	0.70	0.76	0.59	0.65	0.49	0.67	0.97
LA-T	0.54	0.74	0.65	0.83	0.59	0.63	0.52	0.58	0.99
P.Ankle-P.Chest	0.36	0.79	0.71	0.77	0.58	0.66	0.56	0.66	0.96
P.Ankle-P.hand	0.34	0.87	0.70	0.80	0.43	0.78	0.57	0.61	0.95
P.Chest-P.Ankle	0.46	0.75	0.51	0.81	0.51	0.57	0.52	0.58	0.96
P.hand-P.Ankle	0.56	0.70	0.49	0.75	0.40	0.58	0.53	0.55	0.96
Avg	**0.52**	**0.80**	**0.67**	**0.77**	**0.56**	**0.63**	**0.51**	**0.59**	**0.94**

7.5 Robustness to Sensor Noise

In this section, we further examine how sensor noise affects domain adaptation. We inject random Gaussian noise into the target domain data to simulate the real-world situation where the



(a) RA-LA (b) RL-LL (c) RA-T

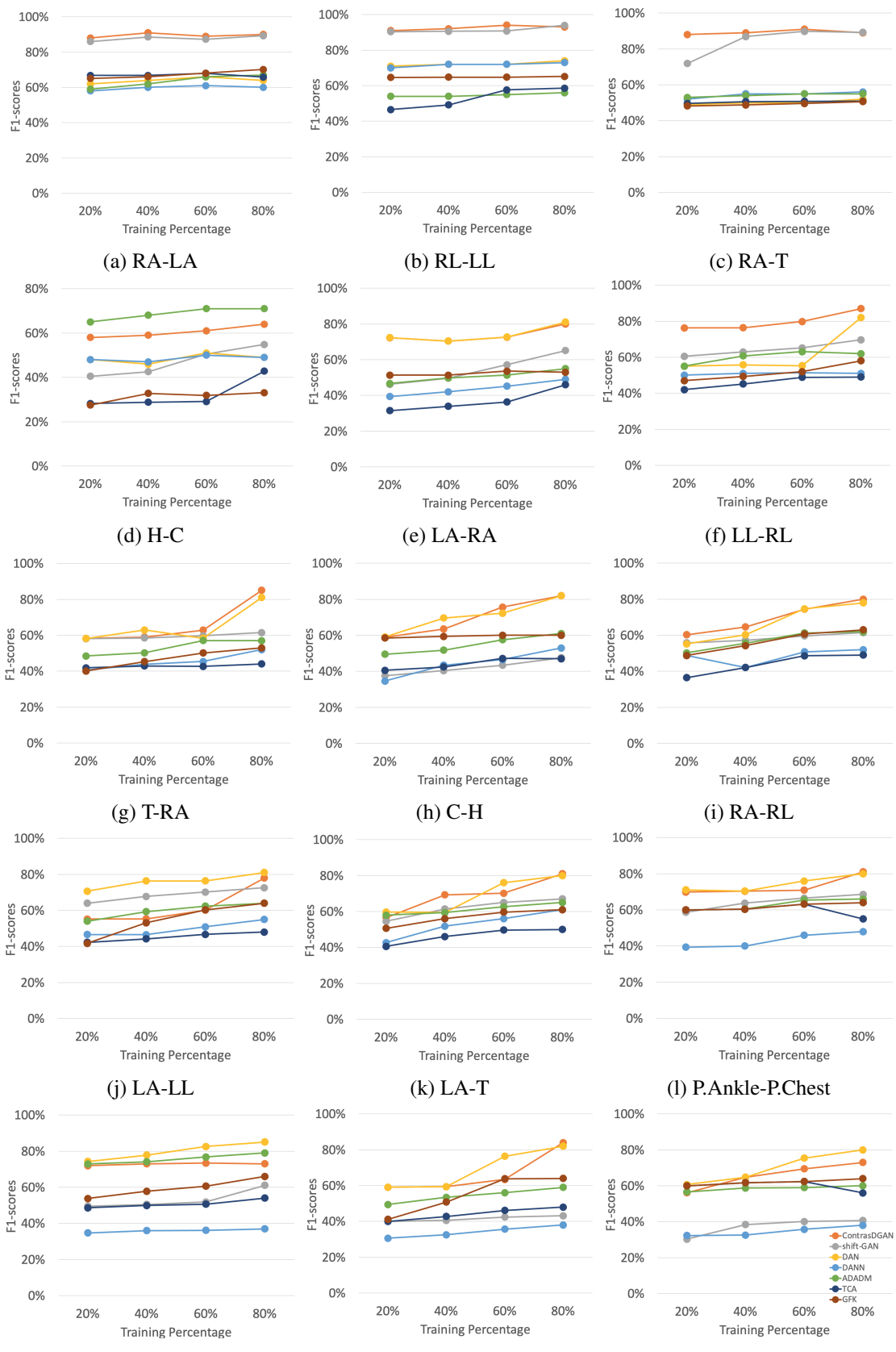
(d) H-C (e) LA-RA (f) LL-RL

(g) T-RA (h) C-H (i) RA-RL

(j) LA-LL (k) LA-T (l) P.Ankle-P.Chest

(m) P.Ankle-P.hand (n) P.Chest-P.Ankle (o) P.hand-P.Ankle

Figure 7.26: Comparison of micro-F1 scores (%) of domain adaptation between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer data.



(a) RA-LA (b) RL-LL (c) RA-T

(d) H-C (e) LA-RA (f) LL-RL

(g) T-RA (h) C-H (i) RA-RL

(j) LA-LL (k) LA-T (l) P.Ankle-P.Chest

(m) P.Ankle-P.hand (n) P.Chest-P.Ankle (o) P.hand-P.Ankle

Figure 7.27: Comparison of macro-F1 scores (%) of domain adaptation between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer data.

Table 7.11: Comparison of average macro-F1 scores between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer sensor datasets across all training percentages.

Task	Lower bound	ContrasDGAN	shift-GAN	DAN	DANN	ADADM	TCA	GFK	Upper bound
RA-LA	0.67	0.90	0.88	0.64	0.60	0.64	0.67	0.67	0.93
RL-LL	0.79	0.93	0.91	0.72	0.72	0.55	0.53	0.65	0.95
RA-T	0.43	0.89	0.84	0.50	0.55	0.54	0.50	0.49	0.94
H-C	0.34	0.61	0.47	0.49	0.49	0.69	0.32	0.31	0.82
LA-RA	0.52	0.74	0.55	0.74	0.44	0.51	0.37	0.52	0.91
LL-RL	0.52	0.80	0.65	0.62	0.51	0.60	0.46	0.52	0.89
T-RA	0.26	0.66	0.59	0.65	0.45	0.53	0.43	0.47	0.97
C-H	0.28	0.70	0.42	0.71	0.44	0.55	0.44	0.60	0.91
RA-LA	0.25	0.70	0.59	0.67	0.48	0.57	0.44	0.57	0.88
LA-LL	0.40	0.62	0.69	0.76	0.50	0.60	0.45	0.55	0.86
LA-T	0.43	0.69	0.62	0.69	0.53	0.61	0.47	0.57	0.88
P.Ankle-P.Chest	0.30	0.73	0.64	0.74	0.43	0.63	0.60	0.62	0.72
P.Ankle-P.hand	0.28	0.73	0.53	0.80	0.36	0.76	0.51	0.60	0.83
P.Chest-P.Ankle	0.41	0.66	0.42	0.69	0.34	0.54	0.44	0.55	0.75
P.hand-P.Ankle	0.50	0.66	0.37	0.70	0.35	0.59	0.60	0.62	0.84
Avg	**0.43**	**0.73**	**0.61**	**0.68**	**0.48**	**0.59**	**0.48**	**0.55**	**0.87**

environment to be adapted to is compromised with unexpected sensor noise. On the test data of the target domain, we randomly select a number of sensors, and for each randomly selected sensor, we inject it with Gaussian noise. The percentage of sensors is chosen from 25% to 100% with a step size of 25%. The mean and variance of Gaussian noise are randomly sampled between 0 and 1.

7.5.1 Binary Sensor Data

Figures 7.28 and 7.29 compare the accuracy of UDAR, *shift*-GAN, ContrasGAN and the existing techniques on different levels of sensor noise. It is evident that ContrasGAN outperforms all the domain adaptation techniques and it is very stable even in the presence of noise. ContrasGAN achieves better accuracy than DAN in task A-C when only 25% of the sensor features are affected by noise; however, DAN is less sensitive to noise and outperforms ContrasGAN when more sensors are injected with noise.

shift-GAN have achieved higher macro-F1 scores than DAN (11.7%), ADADM (1.8%), GFK (28.2%), and TCA (34.2%). This further demonstrates that using GAN for domain adaptation plus KMM for shift correction are stable and effective methods in domain adaptation tasks. ADADM, mixing up the samples, can be more robust in dealing with noise, as the accuracy on ADADM does not vary much with different noise effects. DAN achieves better performance on

H-C since concatenating source and target features will lead to more robust feature learning.

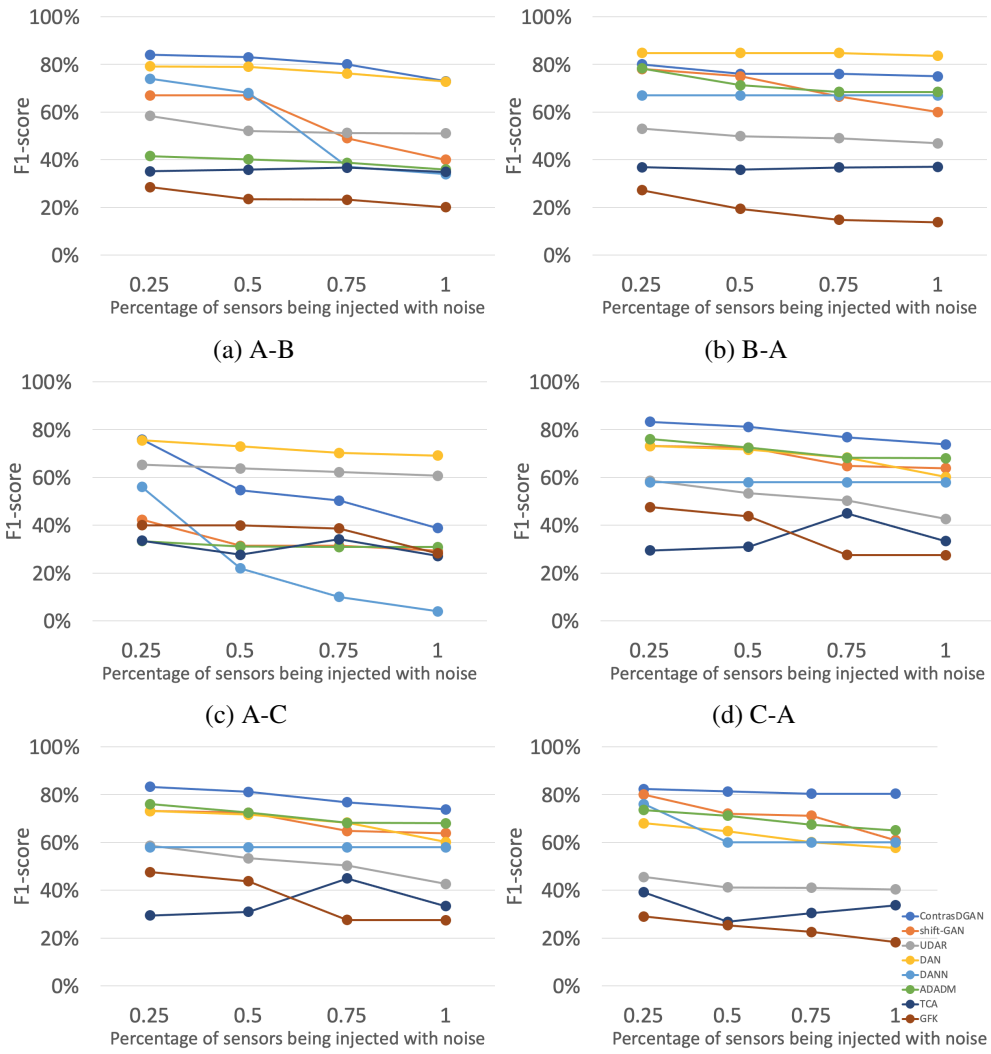
UDAR achieves on average much better performance than TCA (53% vs. 34%) and GFK (53% vs. 28%). UDAR achieves the most stable results during domain adaptation when noise is injected into the sensor features. The overall improvement of UDAR is 32.1%, and 29.7% over TCA, and GFK respectively.

It is also evident that classic techniques are more sensitive to noise. For example, TCA degrades abruptly when noise is introduced in binary sensor data. This is because TCA is unable to find a feature representation for each activity in both domains and it is biased towards one class; that is, after the domain adaptation process the feature representation is not meaningful, which makes the SVM classifier struggle in distinguishing between classes and will only predict one. GFK is mainly affected by the noise and the size of the dataset.

7.5.2 Accelerometer Sensor Data

Tables 7.12 and 7.13 report the average micro-F1 and macro-F1 scores across all percentages of sensor noise for each learning task and we highlight the best micro-F1 and macro-F1 scores among all the domain adaptation techniques. Clearly, ContrasGAN yields better accuracy than the other methods. It has an improvement of 28% and 35% in micro-F1 and macro- F1 scores, respectively, over the lower bound and it outperforms in 9 out of 15 learning tasks. One key factor that may contribute to the superiority of our method is a class-discriminative adaptation that models better the domain shift between the source and the target domain. The second-best method is DAN, which performs 4% and 6% lower than ContrasGAN in micro-F1 and macro-F1 scores respectively.

shift-GAN has achieved higher micro-F1 scores than DANN (11%), ADADM (6%), TCA (12%) and GFK (9%). This demonstrates the robustness and capability of KMM for shift correction even in presence of noise. The poor performance of DANN highlights its weaknesses - learning domain-invariant features is not sufficient to guarantee successful domain adaptation. It is interesting to note that non-deep learning techniques, specifically TCA and GFK, improve very little over the lower bound; 3% (TCA) and 6% (GFK) in micro-F1 scores and 8% (TCA) and 8% (GFK) in macro-F1 scores. This asserts the complexity of the learning process under the presence of noise.



(a) A-B (b) B-A

(c) A-C (d) C-A

(e) B-C (f) C-B

Figure 7.28: Comparison of micro-F1 scores (%) of domain adaptation between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary sensor data injected with Gaussian noise.

In Figures 7.30 and 7.30, we show the micro-F1 and macro-F1 scores on different levels of sensor noise. The x-axis represents the percentage of sensor features injected with Gaussian noise and the y-axis represents the accuracy. Clearly, adding noise to the sensor features makes domain adaptation more difficult. However, all techniques are very stable independently of the percentage of noise injected.

7.6 Summary

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

Table 7.12: Comparison of average micro-F1 scores between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer datasets across different percentage of sensor noise.

Task	Lower bound	ContrasDGAN	shift-GAN	DAN	DANN	ADADM	TCA	GFK	Upper bound
RA-LA	0.58	0.73	0.74	0.62	0.65	0.67	0.65	0.68	0.87
RL-LL	0.66	0.70	0.80	0.73	0.69	0.56	0.47	0.64	0.89
RA-T	0.37	0.69	0.84	0.63	0.64	0.51	0.46	0.44	0.84
H-C	0.39	0.49	0.52	0.54	0.61	0.50	0.51	0.24	0.85
LA-RA	0.43	0.80	0.67	0.81	0.35	0.51	0.52	0.52	0.87
LL-RL	0.49	0.85	0.67	0.76	0.51	0.53	0.47	0.51	0.84
T-RA	0.49	0.84	0.57	0.75	0.41	0.54	0.46	0.50	0.83
C-H	0.45	0.82	0.49	0.78	0.52	0.53	0.47	0.56	0.85
RA-LA	0.41	0.67	0.63	0.74	0.51	0.56	0.47	0.44	0.83
LA-LL	0.47	0.74	0.73	0.74	0.49	0.59	0.46	0.60	0.85
LA-T	0.46	0.77	0.64	0.76	0.50	0.60	0.48	0.52	0.88
P.Ankle-P.Chest	0.56	0.80	0.66	0.77	0.61	0.67	0.51	0.60	0.87
P.Ankle-P.hand	0.51	0.86	0.64	0.76	0.42	0.69	0.59	0.56	0.84
P.Chest-P.Ankle	0.44	0.82	0.40	0.64	0.46	0.55	0.53	0.64	0.85
P.hand-P.Ankle	0.36	0.73	0.35	0.68	0.35	0.49	0.44	0.51	0.88
Avg	**0.47**	**0.76**	**0.62**	**0.72**	**0.51**	**0.57**	**0.50**	**0.53**	**0.86**

Table 7.13: Comparison of average macro-F1 scores between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer datasets across different percentage of sensor noise.

Task	Lower bound	ContrasDGAN	shift-GAN	DAN	DANN	ADADM	TCA	GFK	Upper bound
RA-LA	0.51	0.73	0.73	0.60	0.61	0.64	0.63	0.62	0.82
RL-LL	0.59	0.70	0.78	0.70	0.71	0.53	0.44	0.58	0.84
RA-T	0.30	0.69	0.71	0.61	0.63	0.48	0.44	0.39	0.79
H-C	0.32	0.49	0.51	0.51	0.58	0.48	0.49	0.19	0.80
LA-RA	0.36	0.80	0.66	0.78	0.32	0.48	0.49	0.47	0.82
LL-RL	0.42	0.84	0.66	0.73	0.48	0.50	0.45	0.46	0.79
T-RA	0.42	0.84	0.56	0.72	0.38	0.52	0.44	0.45	0.78
C-H	0.38	0.81	0.48	0.75	0.48	0.50	0.45	0.51	0.80
RA-LA	0.34	0.67	0.62	0.71	0.48	0.53	0.45	0.39	0.78
LA-LL	0.40	0.74	0.72	0.71	0.46	0.56	0.44	0.54	0.80
LA-T	0.39	0.77	0.63	0.73	0.46	0.57	0.46	0.47	0.83
P.Ankle-P.Chest	0.49	0.79	0.65	0.74	0.58	0.64	0.49	0.55	0.82
P.Ankle-P.hand	0.44	0.85	0.63	0.74	0.39	0.66	0.57	0.51	0.79
P.Chest-P.Ankle	0.37	0.81	0.39	0.62	0.42	0.52	0.51	0.59	0.80
P.hand-P.Ankle	0.29	0.72	0.34	0.66	0.32	0.46	0.45	0.45	0.83
Avg	**0.40**	**0.75**	**0.61**	**0.69**	**0.49**	**0.54**	**0.48**	**0.48**	**0.81**

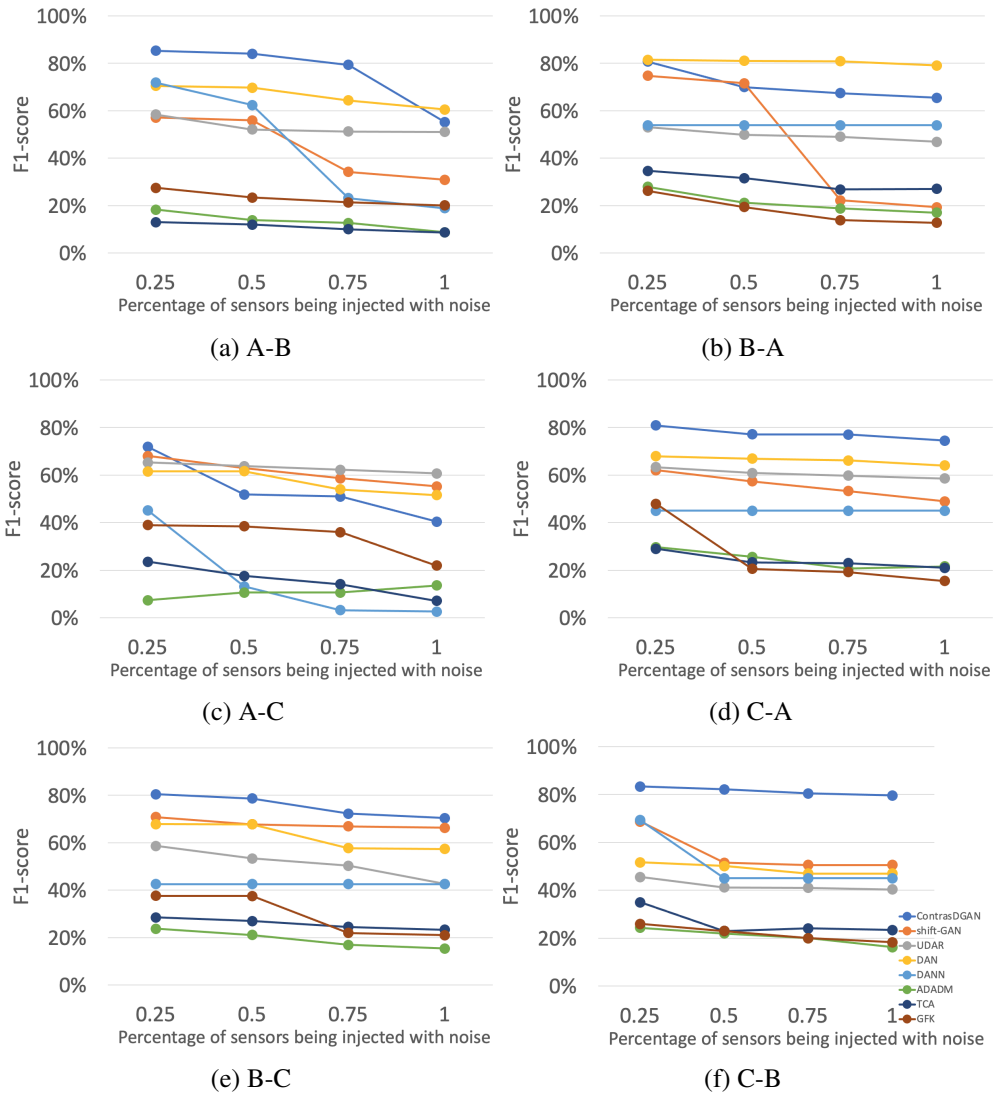
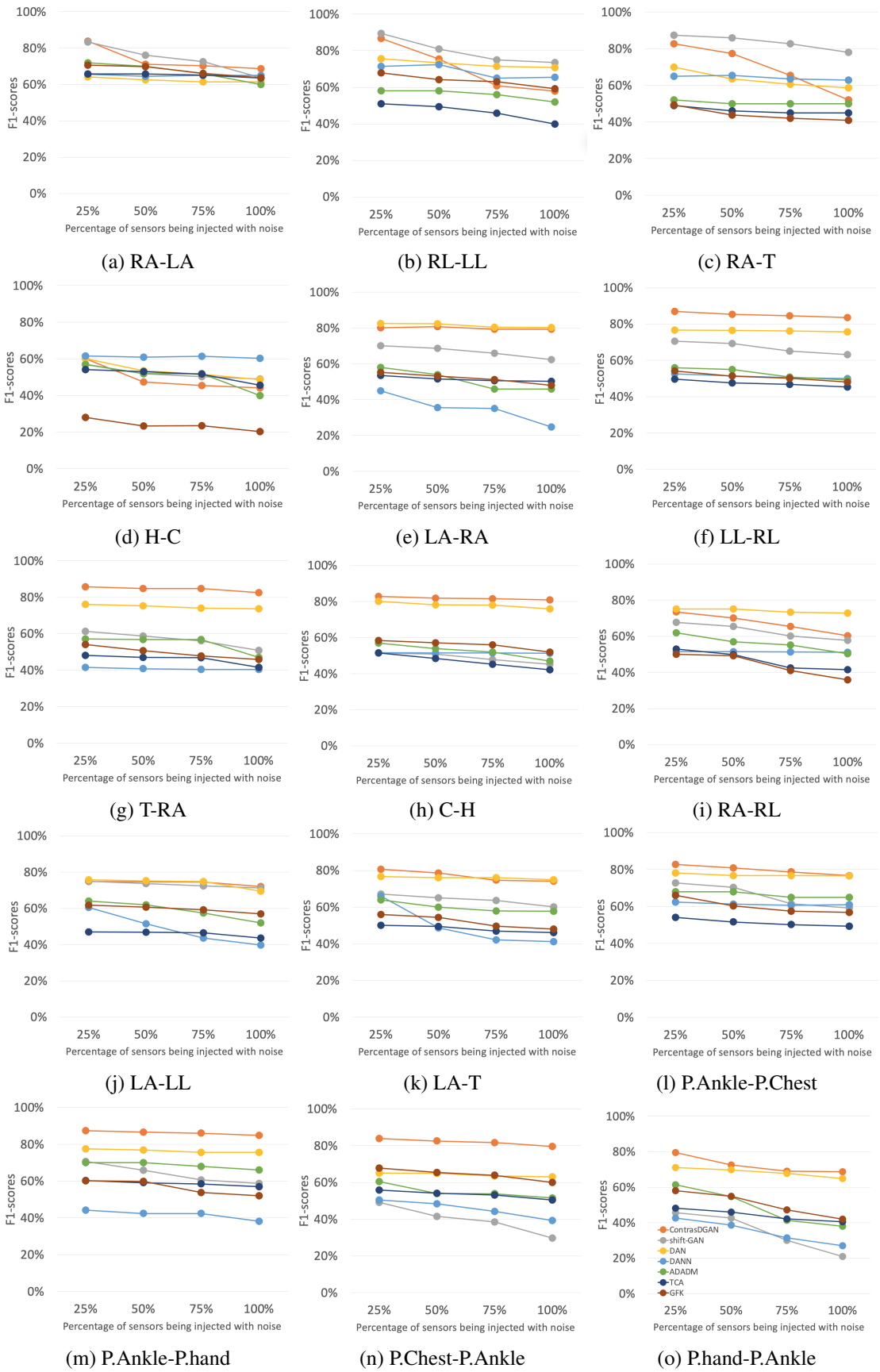


Figure 7.29: Comparison of macro-F1 scores (%) of domain adaptation between ContrasGAN, *shift*-GAN, UDAR and baseline techniques on binary sensor data injected with Gaussian noise.

- ContrasGAN outperforms the baseline techniques on all the transfer learning tasks and datasets with a low computational footprint.
- Contrastive learning helps capture distinctive class-level features, which improve recognition accuracy.
- 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-sensor experiments, when dealing with the most challenging tasks in heteroge-



(a) RA-LA (b) RL-LL (c) RA-T

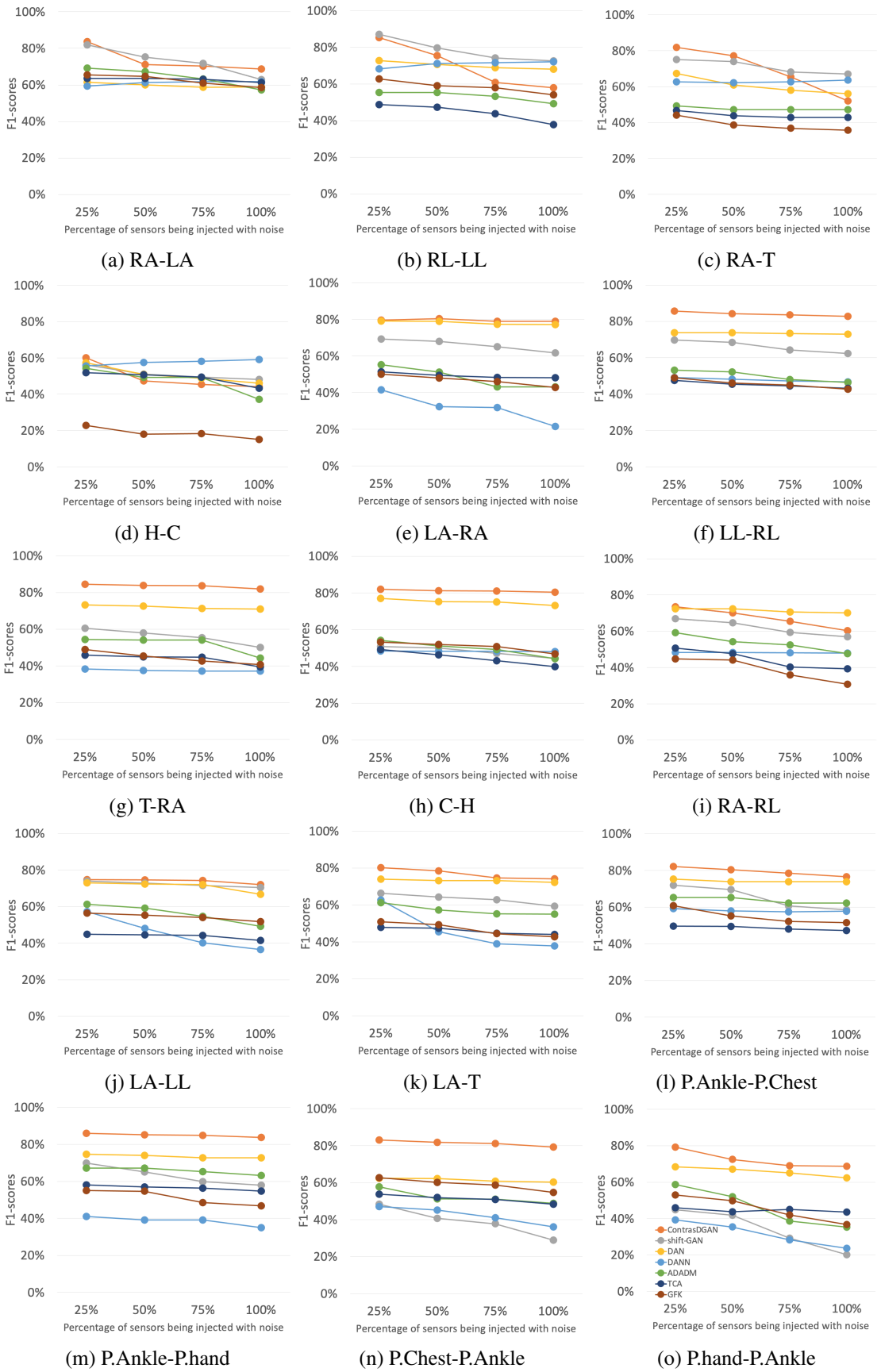
(d) H-C (e) LA-RA (f) LL-RL

(g) T-RA (h) C-H (i) RA-RL

(j) LA-LL (k) LA-T (l) P.Ankle-P.Chest

(m) P.Ankle-P.hand (n) P.Chest-P.Ankle (o) P.hand-P.Ankle

Figure 7.30: Comparison of micro-F1 scores (%) of domain adaptation between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer data with sensor noise.



(a) RA-LA (b) RL-LL (c) RA-T

(d) H-C (e) LA-RA (f) LL-RL

(g) T-RA (h) C-H (i) RA-RL

(j) LA-LL (k) LA-T (l) P.Ankle-P.Chest

(m) P.Ankle-P.hand (n) P.Chest-P.Ankle (o) P.hand-P.Ankle

Figure 7.31: Comparison of macro-F1 scores (%) of domain adaptation between ContrasGAN and *shift*-GAN and baseline techniques on accelerometer data with sensor noise.

neous feature spaces and imbalanced class distribution, ContrasGAN can still produce a consistently good performance, while some of the existing techniques have less stable performance and are more sensitive to the difficulty level of tasks.

- *shift*-GAN is computationally efficient compared to baseline techniques and it has demonstrated its strength of performing a fine-grained feature space alignment.
- UDAR has demonstrated superior performance on domain adaptation over deep learning and non-deep learning-based methods across a range of tasks on binary sensor data, each with different sensor deployments and room layouts.
- UDAR has achieved consistent improvement over the other domain adaptation techniques in the presence of sensor noise and when training data is scarce.

In Table 7.14 we also summarise the main difference between UDAR, *shift*-GAN, ContrasGAN and the best performing comparison techniques, which are DAN, DANN, ADADM, GFK and TCA. The main advantage of ContrasGAN is that it uses contrastive learning to better discriminate samples from different class labels. The results of the experiments show that class-discriminative adaptation improves accuracy. In contrast, *shift*-GAN focuses on the translation between individual samples and does not consider the classes.

The disadvantage of DAN is that it might not perform very well in feature level transfer and the assumption that source and target domain share classifiers might not always be suitable in different adaptation scenarios. DANN learns domain-invariant features to perform domain adaptation. However, it is not clear if the aligned representations and small source error are sufficient assumptions to guarantee a good generalization on the target domain. The results confirm that DANN does not necessarily perform well in the target domain. The disadvantage of ADADM is that it hypothesises that mixing transformed source and real target data can improve domain adaptation. However, the learnt representation can add more complexity to the model, which will result in low performance.

The main advantage of UDAR is that UDAR performs 2-stage intra-class alignment and focuses on capturing intra-class variation using a latent subspace associated with each class. In contrast, GFK computes an *infinite* number of subspaces to obtain the overall new feature representations. TCA assumes that if two domains are related to each other, then there may be

common components between them. These common components may contain less discriminative information with activities that deploy the same sensors.

Table 7.14: Comparison between domain adaptation techniques

Technique	Domain adaptation approach
ContrasGAN	Extends Bi-directional Generative Adversarial Network (Bi-GAN) and includes 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, whereas the inter-class domain discrepancy is maximised to push the representations of each other further away from the decision boundary.
shift-GAN	Extends Bi-directional Generative Adversarial Network (Bi-GAN) that allows transforming from one heterogeneous feature space to another and vice versa through two GAN models. To improve the matching, the model employs Kernel Mean Matching (KMM) to enable covariate shift correction between transformed source data and original target data.
UDAR	Performs 2-stage intra-class alignment. Variational Autoencoders (VAE) captures intra-class variation using a latent subspace associated with each class. Data from the target domain $\mathcal{T}$ is then mapped to the learned embedding. The latent probability distribution function of $\mathcal{T}$ is aligned to that of the source domain $\mathcal{S}$ by matching their means and the eigenvalues of their covariance using the KL divergence.
DAN	Fine-tunes a CNN model on the source labelled samples and introduces MK-MMD-based multi-layer adaptation regulariser to perform layerwise matching so that the source and target domain are as similar as possible under the hidden representations of fully connected layers.
DANN	Learns domain-invariant features by combining domain adaptation with feature learning. It focuses on the $\mathcal{H}$-divergence that relies on the capacity of the hypothesis class $\mathcal{H}$ to distinguish between samples generated by the source domain $\mathcal{D}_s$ and samples generated by the target domain $\mathcal{D}_t$.
ADADM	Advances adversarial learning by mixing transformed source and real target domain samples to train a more robust generator. This is done by using a variant of VAE-GAN.
GFK	Constructs an infinite-dimensional feature space $\mathcal{H}^{\infty}$ that aggregates information on the source domain $\mathcal{S}$, and the target domain $\mathcal{T}$. This is done by extracting the difference in angles between the principal components of the source and target domains. The kernel implicitly maps the data onto all possible subspaces on the geodesic path between domains.
TCA	Learns *transfer components* across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD). This set of common transfer components underlie both domains such that the distance across domains is reduced in a RKHS.

In summary, in this chapter, we discussed and compared our approaches to several competitive methods in domain adaptation for human activity recognition. We discussed the advantages and disadvantages of each technique. In addition, we also provided detailed analysis to validate each component of the architecture in our approaches and their convergence. The next chapter summarises the main contributions of this thesis and discusses future work.

Chapter 8

Conclusion and Future Work

This chapter concludes the thesis, summarises the main contributions and discusses future research directions. The overall goal of this thesis is to learn transferable features to perform unsupervised domain adaptation for human activity recognition. Our methods are evaluated over various experiments using real-world datasets commonly used in human activity recognition and demonstrated better, or at least comparable, accuracy than existing domain adaptation techniques. Based on the results of the experiments, we can answer the following research questions:

- **Q1** Is it possible to relieve the annotation burden on individual users but still be able to build a robust activity recognition model by sharing and transferring activity models across users, even though the sensor deployments and operating environments are different?

  We can conclude that a knowledge-driven approach and deep learning-based model can provide accurate domain adaptation and in most cases outperform existing techniques.

- **Q2** The amount of training data can affect the performance of the model. Can the domain adaptation model achieve high accuracy with little training data?

  We have found that our models are stable and can achieve competitive recognition accuracy regardless of the amount of training data.

- **Q3** The performance of the sensors can vary over time affecting drastically the sensor features. Is it possible to develop a system that performs robustly in the face of sensor noise?

We are able to conclude that adding noise to sensor features adds complexity to the domain adaptation process. Nevertheless, our methods are stable independently of the percentage of Gaussian noise injected and outperform the baseline domain adaptation techniques.

- **Q4** Is it possible to better discriminate samples from different class labels leading to more class-discriminative adaptation?

  We can conclude that contrastive learning improves recognition accuracy significantly. It is robust in noisy data and achieves competitive accuracy even when few training data is available. However, a well-annotated domain is required to transfer the activity model from the source domain to many other unlabelled domains.

The remainder of this chapter summarises our contributions and then discusses future work.

8.1 Summary of Contributions

We summarise the main contributions of this thesis as follows:

1. We have designed a workflow that combines knowledge- and data-driven techniques in performing domain adaptation at different stages. We build on a general ontology for smart home datasets and achieve coarse-grained feature space remapping to link heterogeneous datasets without the need for labelled data in the target domain. This approach uses Variational Autoencoder (VAE) to perform fine-grained feature space alignment.

2. We have developed two GAN-based unsupervised domain adaptation models that do not need any extra knowledge engineering effort to align the source and target domains. The first model called *shift*-GAN integrates bidirectional generative adversarial networks (Bi-GAN) and kernel mean matching (KMM) to learn intrinsic, robust feature transfer between two heterogeneous domains. The second model called ContrasGAN uses bi-directional generative adversarial networks for heterogeneous feature transfer and contrastive learning to capture distinctive features between classes.

3. We have extended Bi-GAN by not just performing one-to-one instance translation but one-to-many instance translation along with an instance selection process to allow more robust domain adaptation.

4. We have performed an extensive empirical evaluation on third-party, real-world datasets that have different spatial layouts and sensor deployments. We have designed different experiments on assessing the effectiveness and robustness of domain adaptation. The results have demonstrated the robustness of our models as they have consistently achieved better, or at least comparable accuracy to baseline domain adaptation techniques.

5. We provide a comprehensive review of human activity recognition and domain adaptation techniques. In terms of human activity recognition, we present different sensor technologies, we provide an overview of the main challenges and future research directions. We compare and evaluate non-deep learning and deep learning-based domain adaptation techniques and we discuss important challenges of domain adaptation in HAR.

6. We design and perform other HAR specific experiments on sensor noise and sensitivity to training data to test the robustness and performance of our models and existing ones. These challenges are more real-world scenarios. Addressing them can improve the credibility of domain adaptation techniques.

8.2 Future Work

Domain adaptation aims to learn domain adaptive representations from the source domain towards representing samples from the target domain. It is present in various real-world applications and has been an energetic research field. To develop robust algorithms, we have to carefully examine the relationship between the source and target domain to understand which method is suitable under which circumstances.

Our experiment results have shown that some non-deep learning and deep learning-based techniques show little improvement over the lower bound. This issue is more evident in the presence of noise and when few training data is available. Several questions arise with this finding: 1) *when do we need domain adaptation?*, 2) *how can we determine if a model trained in*

a source domain may be adapted to our target domain? and 3) *is it possible to build a robust system that can reliably identify the best model to train in the source domain?*. Addressing these questions, we can improve the universality and interpretability of domain adaptation techniques.

Also, the poor performance of some deep learning methods can be related to negative transfer; a typical problem in domain adaptation where the model overfits in the source domain and the transfer knowledge from the source domain can have a negative impact on the target domain. This specific research area is of interest for the robustness of our models, *how to overcome negative transfer to transfer useful knowledge from the source to target domain?*.

One specific research area of domain adaptation that has received little attention is co-adaptation of multiple but heterogeneous domains. We hypothesise that our approaches have generalisation capability as the feature mapping function learnt can be activity-independent. This generalisation capability will enable scalability in domain adaptation techniques.

It is also interesting to consider the implications for this thesis outside human activity recognition. For example, transfer learning is needed in many applications where there are differences between training and deployment scenarios and 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.

Bibliography

[1] D. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *To appear. ACM Computing Surveys*.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[3] A. Akbari and R. Jafari. Transferring activity recognition models for new wearable sensors with deep generative domain adaptation. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, IPSN '19, pages 85–96, New York, NY, USA, 2019. ACM.

[4] U. Akdemir, P. Turaga, and R. Chellappa. An ontology based approach for activity recognition from video. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, page 709–712, New York, NY, USA, 2008. Association for Computing Machinery.

[5] R. Alaiz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Improving classification under changes in class and within-class distributions. In J. Cabestany, F. Sandoval, A. Prieto, and J. M. Corchado, editors, *Bio-Inspired Systems: Computational and Ambient Intelligence*, pages 122–130, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[6] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. C. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *CoRR*, abs/1802.10151, 2018.

[7] B. Almaslukh. An effective deep autoencoder approach for online smartphone-based human activity recognition. *International Journal of Computer Science and Network Security*, 17, 04 2017.

[8] K. Altun and B. 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.

[9] K. Altun, B. Barshan, and O. Tunael. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605 – 3620, 2010.

[10] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, pages 1–17, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[11] L. Bashmal, Y. Bazi, H. Alhichri, M. Alrahhal, N. Ammour, and N. Alajlan. Siamese-gan: Learning invariant representations for aerial vehicle image categorization. *Remote Sensing*, 10, 02 2018.

[12] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1–2):151–175, May 2010.

[13] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

[14] Y. Bengio. Deep learning of representations: Looking forward. *CoRR*, abs/1305.0445, 2013.

[15] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and M. T. Kechadi. Human activity recognition with convolutional neural netowrks. *CoRR*, abs/1906.01935, 2019.

[16] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *J. Mach. Learn. Res.*, 10:2137–2155, 2009.

[17] M. Borga. Canonical correlation: a tutorial, 2001.

[18] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, 07 2006.

[19] O. Brdiczka, P. Reignier, and J. L. Crowley. Detecting individual activities from video in a smart home. In B. Apolloni, R. J. Howlett, and L. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 363–370, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[20] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing daily activities with rfid-based sensors. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, UbiComp '09, page 51–60, New York, NY, USA, 2009. Association for Computing Machinery.

[21] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3), Jan. 2014.

[22] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar. A systematic study of unsupervised domain adaptation for robust human-activity recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(1), Mar. 2020.

[23] D. Chen, J. Yang, and H. D. Wactlar. Towards automatic analysis of social interaction patterns in a nursing home environment from video. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '04, page 283–290, New York, NY, USA, 2004. Association for Computing Machinery.

[24] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):790–808, Nov 2012.

[25] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong, and M. Poland. A logical framework for behaviour reasoning and assistance in a smart home. *International Journal of Assistive Robotics and Mechatronics*, 9(4):20–34, Dec. 2008.

[26] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, 2012.

[27] M. Chen, Z. E. Xu, K. Q. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *CoRR*, abs/1206.4683, 2012.

[28] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *CoRR*, abs/1711.09020, 2017.

[30] D. Cook, K. Feuz, and N. Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36:537–556, 09 2013.

[31] D. Cook, K. D. Feuz, and N. C. Krishnan. Transfer learning for activity recognition: A survey. *Knowl. Inf. Syst.*, 36(3):537–556, Sept. 2013.

[32] D. Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48 5:480–5, 2009.

[33] D. Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, 48:480–485, 2009.

[34] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2013.

[35] P. Cottone, S. Gaglio, G. L. Re, and M. Ortolani. User activity recognition for energy saving in smart homes. *Pervasive and Mobile Computing*, 16(Part A):156 – 170, 2015.

[36] Z. Cui, H. Chang, S. Shan, and X. Chen. Generalized unsupervised manifold alignment. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors,

Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014.

[37] S. Dai, Y. Cheng, Y. Zhang, Z. Gan, J. Liu, and L. Carin. Contrastively smoothed class alignment for unsupervised domain adaptation, 2019.

[38] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 193–200, New York, NY, USA, 2007. Association for Computing Machinery.

[39] J. Davis and P. Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 217–224, New York, NY, USA, 2009. Association for Computing Machinery.

[40] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi. Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey. *IEEE Access*, 8:210816–210836, 2020.

[41] T. Diethe, D. Hardoon, and J. Shawe-Taylor. Multiview fisher discriminant analysis. *NIPS Workshop on Learning from Multiple Sources*, 12 2008.

[42] C. B. Do and A. Y. Ng. Transfer learning for text classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.

[43] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[44] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[45] L. Fang, J. Ye, and S. Dobson. Discovery and recognition of emerging human activities using a hierarchical mixture of directional statistical models. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019.

[46] A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, L. E. Miller, and S. A. Solla. Adversarial domain adaptation for stable brain-machine interfaces, 2018.

[47] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[48] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, page 2960–2967, USA, 2013. IEEE Computer Society.

[49] K. D. Feuz and D. J. Cook. Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr). *ACM Trans. Intell. Syst. Technol.*, 6(1):3:1–3:27, Mar. 2015.

[50] K. Fishkin, M. Philipose, and A. Rea. Hands-on rfid: wireless wearables for detecting use of objects. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 38–41, 2005.

[51] A. Gaddam, S. C. Mukhopadhyay, and G. S. Gupta. Elder care based on cognitive sensor network. *IEEE Sensors Journal*, 11(3):574–581, 2011.

[52] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189. JMLR.org, 2015.

[53] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, Jan. 2016.

[54] B. Georis, M. Maziere, and M. Thonnat. A video interpretation platform applied to bank agency monitoring. pages 46 – 50, 03 2004.

[55] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.

[56] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE Computer Society, 2012.

[57] G. Gordalina, J. Figueiredo, R. Martinho, R. Rijo, P. Correia, P. Assunção, A. Seco, G. Pires, L. Oliveira, and R. Fonseca-Pinto. Tracking human routines towards adaptive monitoring: the movida.domus platform. *Procedia Computer Science*, 138:41–48, 2018. CENTERIS 2018 - International Conference on ENTERprise Information Systems / ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018.

[58] K. 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.

[59] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, B. Schölkopf, J. Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning, 131-160 (2009)*, 01 2009.

[60] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu. Optimal kernel choice for large-scale two-sample tests. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1205–1213, Red Hook, NY, USA, 2012. Curran Associates Inc.

[61] H. Hachiya, M. Sugiyama, and N. Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomput.*, 80(C):93–101, Mar. 2012.

[62] 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.

[63] M. Haescher. Multi-sensory environment analysis and human activity recognition via wearable technologies. 03 2014.

[64] A. Hakeem and M. Shah. Ontology and taxonomy collaborated framework for meeting classification. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 219–222 Vol.4, 2004.

[65] N. Y. Hammerla, S. Halloran, and T. Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *CoRR*, abs/1604.08880, 2016.

[66] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: a programmable pervasive space. *Computer*, 38(3):50–60, 2005.

[67] M. R. Hodges and M. E. Pollack. An 'object-use fingerprint': The use of electronic sensors for human identification. In J. Krumm, G. D. Abowd, A. Seneviratne, and T. Strang, editors, *UbiComp 2007: Ubiquitous Computing*, pages 289–303, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[68] D. Hollosi, J. Schröder, S. Goetze, and J.-E. Appell. Voice activity detection driven acoustic event classification for monitoring in smart homes. In *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*, pages 1–5, 2010.

[69] H. Hotelling. *Relations Between Two Sets of Variates*, pages 162–190. Springer New York, New York, NY, 1992.

[70] L. Hu, M. Kan, S. Shan, and X. Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2018.

[71] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 601–608, Cambridge, MA, USA, 2006. MIT Press.

[72] Z. Hussain, M. Sheng, and W. E. Zhang. Different approaches for human activity recognition: A survey. *CoRR*, abs/1906.05074, 2019.

[73] H. D. III. Frustratingly easy domain adaptation. *CoRR*, abs/0907.1815, 2009.

[74] A. Jayatilaka and D. C. Ranasinghe. Real-time fluid intake gesture recognition based on batteryless uhf rfid technology. *Pervasive and Mobile Computing*, 34:146–156, 2017. Pervasive Computing for Gerontechnology.

[75] W. Jitkrittum, P. Sangkloy, M. W. Gondal, A. Raj, J. Hays, and B. Schölkopf. Kernel mean matching for content addressability of GANs. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3140–3151, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[76] C. Jobanputra, J. Bavishi, and N. Doshi. Human activity recognition: A survey. *Procedia Computer Science*, 155:698–703, 2019. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

[77] A. Karpov, L. Akarun, H. Yalçın, A. Ronzhin, B. Demiroz, A. Çoban, and M. Železný. Audio-visual signal processing in a multimodal assisted living environment. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1023–1027, 01 2014.

[78] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[79] T. L. M. Kasteren, G. Englebienne, and B. J. A. Kröse. Human activity recognition from wireless sensor network data: Benchmark and software. In L. Chen, C. D. Nugent, J. Biswas, J. Hoey, and I. Khalil, editors, *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, pages 165–186. Atlantis Press, 2011.

[80] T. L. M. Kasteren, G. Englebienne, and B. J. A. Kröse. Human activity recognition from wireless sensor network data: Benchmark and software. In L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, editors, *Activity Recognition in Pervasive Intelligent Environments*,

volume 4 of *Atlantis Ambient and Pervasive Intelligence*, chapter 8, pages 165–186. Atlantis Press, Paris, France, 2011.

[81] M. A. A. H. Khan and N. Roy. Transact: Transfer learning enabled activity recognition. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 545–550, March 2017.

[82] C. D. Kidd, R. J. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. D. Mynatt, T. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In N. A. Streitz, J. Siegel, V. Hartkopf, and S. Konomi, editors, *CoBuild*, volume 1670 of *Lecture Notes in Computer Science*, pages 191–198. Springer, 1999.

[83] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9(1):48–53, Jan. 2010.

[84] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[85] W. M. Kouw. An introduction to domain adaptation and transfer learning. *CoRR*, abs/1812.11806, 2018.

[86] W. M. Kouw, L. J. van der Maaten, J. H. Krijthe, and M. Loog. Feature-level domain adaptation. *Journal of Machine Learning Research*, 17(171):1–32, 2016.

[87] F. Krüger, K. Yordanova, V. Köppen, and T. Kirste. Towards tool support for computational causal behavior models for activity recognition. 09 2012.

[88] K. Kunze, M. Barry, E. A. Heinz, P. Lukowicz, D. Majoe, and J. Gutknecht. Towards recognizing tai chi - an initial experiment using wearable sensors. In *3rd International Forum on Applied Wearable Computing 2006*, pages 1–6, 2006.

[89] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1354–1361, 2012.

[90] N. D. Lane, P. Georgiev, and L. Qendro. Deepear: Robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 283–294, New York, NY, USA, 2015. Association for Computing Machinery.

[91] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, Third 2013.

[92] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

[93] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[94] Y. Li, D. Shi, B. Ding, and D. Liu. Unsupervised feature learning for human activity recognition using smartphone sensors. In R. Prasath, P. O'Reilly, and T. Kathirvalavakumar, editors, *Mining Intelligence and Knowledge Exploration*, pages 99–107, Cham, 2014. Springer International Publishing.

[95] M. Liu and O. Tuzel. Coupled generative adversarial networks. *CoRR*, abs/1606.07536, 2016.

[96] M. Long, Y. Cao, J. Wang, and M. 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.

[97] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[98] M. Long, H. Zhu, J. Wang, and M. 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.

[99] T. Maekawa and S. Watanabe. Unsupervised activity recognition with user's physical characteristics data. In *2011 15th Annual International Symposium on Wearable Computers*, pages 89–96, June 2011.

[100] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017.

[101] H. P. Martínez, G. N. Yannakakis, and J. Hallam. Don't classify ratings of affect; rank them! *IEEE Transactions on Affective Computing*, 5(3):314–326, 2014.

[102] I. Maurtua, P. T. Kirisci, T. Stiefmeier, M. L. Sbodio, and H. Witt. A wearable computing prototype for supporting training activities in automotive production. In *4th International Forum on Applied Wearable Computing 2007*, pages 1–12, 2007.

[103] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, Nov. 1995.

[104] M. Mirza and S. Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.

[105] T. M. Mitchell. The need for biases in learning generalizations. Technical report, 1980.

[106] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

[107] M. E. Mlinac and M. C. Feng. Assessment of Activities of Daily Living, Self-Care, and Independence. *Archives of Clinical Neuropsychology*, 31(6):506–516, 08 2016.

[108] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[109] L. H. Morsing, O. A. Sheikh-Omar, and A. Iosifidis. Supervised domain adaptation: A graph embedding perspective and a rectified experimental protocol, 2020.

[110] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5716–5726, 2017.

[111] K. Mu, et, B. Sriperumbudur, K. Fukumizu, A. Gretton, and B. Schölkopf. Kernel mean shrinkage estimators. *Journal of Machine Learning Research*, 17(48):1–41, 2016.

[112] N. T. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *BMVC '06*, pages 126.1–126.10, 2006.

[113] B. Ni, P. Moulin, X. Yang, and S. Yan. Motion part regularization: Improving action recognition via trajectory group selection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3698–3706, 2015.

[114] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion*, 46:147–170, 2019.

[115] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651. PMLR, 06–11 Aug 2017.

[116] G. O. of Science. Future of ageing population. *Foresight*, 2016.

[117] S. Oniga and J. Sütő. Human activity recognition using neural networks. In *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*, pages 403–406, 2014.

[118] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[119] 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.

[120] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[121] D. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 44–51, 2005.

[122] M. Perkowitz, M. Philipose, K. Fishkin, and D. Patterson. Mining models of human activities from the web. pages 573–582, 05 2004.

[123] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[124] T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1729–1734. AAAI Press, 2011.

[125] A. Pounds-Cornish and A. Holmes. The idorm - a practical deployment of grid technology. pages 470– 470, 06 2002.

[126] X. Qin, Y. Chen, J. Wang, and C. Yu. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), Dec. 2019.

[127] B. Quanz and J. Huan. Large margin transductive transfer learning. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1327–1336, New York, NY, USA, 2009. Association for Computing Machinery.

[128] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[129] V. Radu, C. Tong, S. Bhattacharya, N. Lane, C. Mascolo, M. Marina, and F. Kawsar. Multimodal deep learning for activity and context recognition. In *Proceedings of Ubicomp '18*, 2018.

[130] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 759–766, New York, NY, USA, 2007. Association for Computing Machinery.

[131] S. R. Ramamurthy and N. Roy. Recent trends in machine learning for human activity recognition—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8, 2018.

[132] P. Rashidi and D. J. Cook. D.j.: Multi home transfer learning for resident activity discovery and recognition. In *In: KDD Knowledge Discovery from Sensor Data*, pages 56–63, 2010.

[133] 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.

[134] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *Proceedings of PerCom '15*, pages 149–154, 2015.

[135] A. Rosales and J. Ye. Unsupervised domain adaptation for activity recognition across heterogeneous datasets. *Pervasive and Mobile Computing*, 2020.

[136] F. M. Rueda, S. Lüdtke, M. Schröder, K. Yordanova, T. Kirste, and G. A. Fink. Combining symbolic reasoning and deep learning for human activity recognition. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 22–27, 2019.

[137] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 213–226, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[138] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *2018 IEEE Conference on Computer Vision and*

Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 3723–3732. IEEE Computer Society, 2018.

[139] N. Saleheen, A. A. Ali, S. M. Hossain, H. Sarker, S. Chatterjee, B. Marlin, E. Ertin, M. al'Absi, and S. Kumar. puffmarker: A multi-sensor approach for pinpointing the timing of first lapse in smoking cessation. In *Proceedings of UbiComp '15*, pages 999–1010, 2015.

[140] A. R. Sanabria and J. Ye. Unsupervised domain adaptation for activity recognition across heterogeneous datasets. *Pervasive and Mobile Computing*, 64:101147, 2020.

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

[142] M. Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press, Cambridge, MA, USA, 1997.

[143] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109, 08 2014.

[144] K. Shirahama, M. Grzegorzek, and L. Koping. Codebook approach for sensor-based human activity recognition. 2016.

[145] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger. Human activity recognition using recurrent neural networks. *CoRR*, abs/1804.07144, 2018.

[146] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors*, 19(14), 2019.

[147] T. Starner and M. Group. Visual recognition of american sign language using hidden markov models. 05 1995.

[148] M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics Decisions*, 23:249–279, 01 2005.

[149] B. Sun, J. Feng, and K. Saenko. Correlation alignment for unsupervised domain adaptation. *CoRR*, abs/1612.01939, 2016.

[150] B. Sun and K. Saenko. Deep CORAL: correlation alignment for deep domain adaptation. *CoRR*, abs/1607.01719, 2016.

[151] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham, 2016. Springer International Publishing.

[152] J. T. Sunny, S. George, and J. J. Kizhakkethottam. Applications and challenges of human activity recognition using sensors in a smart environment. 2015.

[153] N. Suzuki, Y. Watanabe, and A. Nakazawa. Gan-based style transformation to improve gesture-recognition accuracy. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), Dec. 2020.

[154] H. Tang and K. Jia. Discriminative adversarial domain adaptation. *CoRR*, abs/1911.12036, 2019.

[155] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, pages 158–175, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[156] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. Cir. and Sys. for Video Technol.*, 18(11):1473–1488, Nov. 2008.

[157] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[158] M. Vacher, F. Portet, A. Fleury, and N. Noury. Development of audio sensing technology for ambient assisted living: Applications and challenges. *IJEHMC*, 2:35–54, 03 2011.

[159] L. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:1–21, 2014.

[160] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *UbiComp '08*, pages 1–9, New York, NY, USA, 2008. ACM.

[161] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. Transferring knowledge of activity recognition across sensor networks. In P. Floréen, A. Krüger, and M. Spasojevic, editors, *Proceedings of the 8th International Conference on Pervasive Computing*, pages 283–300, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[162] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. *Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software*, pages 165–186. Atlantis Press, Paris, 2011.

[163] P. Vepakomma, D. De, S. K. Das, and S. Bhansali. A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6, 2015.

[164] M. Vrigkas, C. Nikou, and I. A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

[165] K. Walse, R. Dharaskar, and V. M. Thakare. *PCA Based Optimal ANN Classifiers for Human Activity Recognition Using Mobile Sensors Data*, volume 50, pages 429–436. 01 2016.

[166] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu. Human activity recognition in a smart home environment with stacked denoising autoencoders. In S. Song and Y. Tong, editors, *Web-Age Information Management*, pages 29–40, Cham, 2016. Springer International Publishing.

[167] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *CoRR*, abs/1707.03502, 2017.

[168] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 2018.

[169] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018.

[170] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, Mar 2018.

[171] Y. Wang, S. Cang, and H. Yu. A survey on wearable sensor modality centred human activity recognition in health care. *Expert Systems with Applications*, 137:167–190, 2019.

[172] G. M. Weiss, K. Yoneda, and T. Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

[173] D. H. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In H. W. Gellersen, R. Want, and A. Schmidt, editors, *Pervasive Computing*, pages 62–79, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[174] G. Wilson, J. R. Doppa, and D. 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.

[175] W. Wobcke. Two Logical Theories of Plan Recognition. *Journal of Logic and Computation*, 12(3):371–412, 06 2002.

[176] C. R. Wren and E. M. Tapia. Toward scalable activity recognition for sensor networks. In M. Hazas, J. Krumm, and T. Strang, editors, *Location- and Context-Awareness*, pages 168–185, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[177] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 5:975–1005, Dec. 2004.

[178] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94)*, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[179] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6502–6509. AAAI Press, 2020.

[180] X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2492–2501, 2019.

[181] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, page 188–197, New York, NY, USA, 2007. Association for Computing Machinery.

[182] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2030–2037, 2010.

[183] J. Ye. Slearn: Share learning human activity labels across multiple datasets. In *Proceedings of PerCom '18*, 2018. To appear.

[184] J. Ye. Slearn: Shared learning human activity labels across multiple datasets. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10, March 2018.

[185] J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: a review. *Pervasive and mobile computing*, 8:36–66, 2012.

[186] J. Ye, G. Stevenson, and S. Dobson. Usmart: An unsupervised semantic mining activity recognition technique. *ACM Trans. Interact. Intell. Syst.*, 4(4):16:1–16:27, Nov. 2014.

[187] J. Ye, G. Stevenson, and S. Dobson. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47 – 70, 2015.

[188] J. Ye, G. Stevenson, and S. Dobson. Detecting abnormal events on binary sensors in smart home environments. *Pervasive and Mobile Computing*, 33:32 – 49, 2016.

[189] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV 17*, 2017.

[190] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. *CoRR*, abs/1704.02510, 2017.

[191] K. Yordanova, S. Bader, C. Heine, S. Teipel, and T. Kirste. Towards a situation model for assessing challenging behaviour of people with dementia. In *Proceedings of the 3rd International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '16, New York, NY, USA, 2016. Association for Computing Machinery.

[192] K. Yordanova and T. Kirste. Learning models of human behaviour from textual instructions. 02 2016.

[193] F. Zenke, B. Poole, and S. Ganguli. Improved multitask learning through synaptic intelligence. *CoRR*, abs/1703.04200, 2017.

[194] D. Zhai, B. Li, H. Chang, S. Shan, X. Chen, and W. Gao. Manifold alignment via corresponding projections. In *BMVC*, 2010.

[195] J. Zhang, W. Li, P. Ogunbona, and D. Xu. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Comput. Surv.*, 52(1), Feb. 2019.

[196] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li. A review on human activity recognition using vision-based method. *Journal of Healthcare Engineering*, 2017, 2017.

[197] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li. A review on human activity recognition using vision-based method. *Journal of Healthcare Engineering*, 2017:1–31, 07 2017.

[198] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer. Multi-source distilling domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12975–12983, Apr. 2020.

[199] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. Cross-people mobile-phone based activity recognition. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2545–2550. AAAI Press, 2011.

[200] J. Zheng, Z. Jiang, P. J. Phillips, and R. Chellappa. Cross-view action recognition via a transferable dictionary pair. *IEEE Transactions on Image Processing*, 25, 11 2012.

[201] V. W. Zheng, D. H. Hu, and Q. Yang. Cross-domain activity recognition. In *Proceedings of the 11th international conference on Ubiquitous computing (Ubicomp '09)*, pages 61–70. ACM, 2009.

[202] F. Zhu and L. Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *Int. J. Comput. Vision*, 109(1–2):42–59, Aug. 2014.

[203] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[204] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.