OVERCOMING DATA SCARCITY IN IMU-BASED HUMAN MOTION ANALYSIS

OVERCOMING DATA SCARCITY IN IMU-BASED HUMAN MOTION ANALYSIS

BY

YUJIAO HAO, M.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTING & SOFTWARE
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2022)	McMaster University
(Computing & Software)	Hamilton, Ontario, Canada

TITLE:	Overcoming Data Scarcity in IMU-based Human Motion Analysis
AUTHOR:	Yujiao Hao M.Sc., (Computer Science) Northeastern University, Shenyang, China
SUPERVISOR:	Prof. Rong Zheng
NUMBER OF PAGES:	xv, 154

Abstract

Deep learning techniques enable the automatic analysis and interpretation of human motion from wearable sensor. However, despite extensive research efforts, the absence of vast cleanly labeled human motion sensor data in free-living setting still hindered the applications of deep models in the real world. Human motion data is known to have a large variance among subjects and devices. As the result, there is a significant performance gap between models trained with and without part of test subjects' data. In addition, due to the difficulty in labeling wearable sensor data post hoc, existing public datasets are either collected from scripted activities under controlled settings or contain severe label noises when collected through crowdsourcing. Moreover, since collecting motion data from frail populations such as older adults with impaired mobility can be physically demanding or even cause safety concerns, the data scarcity problem becomes more severe. In this dissertation, we aim to address these challenges through a multi-pronged approach.

First, we investigate domain adaptation techniques to handle the subject variance and device diversity in wearable sensor-based human activity recognition (HAR). We propose an invariant feature learning framework (IFLF) that extracts common information shared across subjects and devices. It incorporates two learning paradigms: 1) meta-learning to capture robust features across multiple source domains and adapt trained model to a target domain with similarity-based data selection; and 2) multi-task learning to deal with data shortage and enhance overall performance via knowledge sharing among different domains. Experimental results demonstrate that IFLF is effective in handling both subject and device variations across popular open datasets and an in-house dataset from older adults.

Inertial measurement units (IMU) datasets collected in naturalistic settings are often fraught with labeling noise due to misaligned onsets, the presence of concurrent activities, unpredictable terrains or human errors. However, state-of-the-art learning with label noise (LNL) approaches fail to converge due to the presence of subject variations. As a second contribution, we propose VALERIAN, an invariant feature learning for in-the-wild domain adaptation method for wearable sensor-based HAR. It consists of three components: self-supervised pre-training, invariant feature learning with noisy labels, and fast adaptation to new subjects. By training a multi-task model with separate task-specific layers for each subject, VALERIAN allows noisy labels to be dealt with individually for each subject while benefiting from shared feature representation across subjects. Experimental results show that VALERIAN significantly outperforms baseline approaches.

Simulating IMU data from other input modalities offers an alternative way to mitigate the wearable data scarcity problem. As a third contribution, we design CROMOSim, a cross-modality sensor simulator that synthesizes high fidelity virtual IMU data from data collected with motion capture systems or monocular RGB cameras. It utilizes a skinned multi-person linear model for 3D body pose and shape representations to enable simulating motions from arbitrary on-body positions. A deep learning model is used to learn the functional mapping from imperfect trajectory estimations in a 3D body tri-mesh representation to IMU data. Extensive empirical evidence demonstrates the high fidelity and utility of CROMOSim simulated data in downstream human motion analysis tasks include HAR and human pose estimation.

Acknowledgement

I'd like to express my gratitude to the help of my supervisor, Dr. Rong Zheng. I am extremely grateful to be her student and to know that she had faith in me over the past years. It would not have been possible for me to finish the doctoral program without her guidance and support.

I would like to thank Prof. Boyu Wang, his encouraging words and thoughtful, detailed feedback enlightened my way to research. I also would like to thank my committee members, Dr. Hassan Ashtiani and Dr. Wenbo He, their inspiring questions and suggestions have been very important to me. Thanks to the ABLE and MobilityAI research teams, who are excellent collaborators and whose practical research initiatives motivated me to conduct my study.

I want to thank my parents, for their endless supports. Pursuing a doctoral degree after years of working in the industry is crazy, I really appreciate that they always standing behind me without any exception.

I'd also like to thank everyone in the WiSeR group, for the technical and emotional support they gave. Even though a researcher is meant to be alone, I did not feel lonely in the past four years with their companionship.

Contents

List of Figures

List of Tables

List of Abbreviations

BMI Body Mass Index

CNN Convolutional Neural Network

CV Computer Vision

DA Domain Adaptation

DNN Deep Neural Network

DoF Degree of Freedom

GAN Generative Adversarial Network

GMM Gaussian Mixture Model

HAR Human Activity Recognition

HPE Human Pose Estimation

IMU Inertial Measurement Units

LNL Learning with Noisy Labels

LSTM Long Short-term Memory

MEMS Micro-Electro-Mechanical System

MoCap Motion Capture

RGB Red Green Blue

SMPL 3D Skinned Multi-Person Linear Model

UDA Unsupervised Domain Adaptation

Declaration of Academic Achievement

The work presented here is the result of research performed by myself during the years 2018-2022. Results which have substantial contributions from other authors are clearly prefaced and the contributions of those authors are indicated.

Chapter 1

Introduction

1.1 Motivation

Human motion analysis has been widely studied and employed in numerous real-world applications with wearable sensors. There are two categories of research questions in human motion analysis. The first category of problems, called *human activity recognition* (HAR), are discriminative in spirit and aim at detecting events associated with one or more types of sensor inputs. The second category of problems are more challenging and require the reconstruction or tracking of poses for a full body or a single body part, known as *human pose estimation* (HPE). The wide availability of wearable devices and significant advances machine learning technologies paves the way for automatic yet accurate human motion analysis in daily life.

Training supervised deep learning models generally require a large amount of well-curated labeled sensory data [2]. Existing public inertial measurement unit (IMU) sensor datasets are typically collected under controlled settings where subjects are asked to perform scripted activities in a lab environment. Such datasets tend to have a small collection of subjects and activities, and exhibit very different characteristics from those collected in the wild [3]. Collecting wearable sensor data for human motion analysis in the wild faces its own set of challenges. One main difficulty is to label such data accurately [4]. Recalls from one's memory are known to be notoriously unreliable [5] while labeling wearable data by observing signal patterns requires extensive domain knowledge and is error prone. Yet another issue arises in collecting data for certain activities (e.g., falls) from specific population groups, such as frail older adults with declined physical abilities. With the total number of older adults projected to reach 1.2 billion by 2025 and 2 billion by 2050 [6], the importance of mobility data form older adults cannot be under-estimated by anyone who designs health and fitness applications using wearable devices. But the decrease in mobility leads to less physical activity. Strenuous activities like walking upstairs or jogging are difficult for older people to perform, resulting in under-representation of older adult population in public HAR and HPE datasets. In addition, diversity in sensor data distributions is more pronounced among older adults due to their diverse mobility statuses. Such subject diversity hurts the performance of deep learning methods as neural networks generally have poor generalization ability when the distribution of test data (target domain) differs from that of training data (source domain).

Existing solutions to data scarcity in wearable sensor human motion analysis mainly fall into two categories: 1) reducing the amount of data required from a target domain, and 2) generating more IMU data for model training. This dissertation explores both directions.

In reducing the amount of data required from the target domain, we further consider two problem setups: data from source domains are collected under controlled environments with clean labels and in the wild with noisy labels. Source domains may differ from the target domain due to subject differences, device diversity and different sensor placements. Such domain gaps hinder the direct application of a machine learning model trained on the source domain to the target domain. Various unsupervised domain adaptation (UDA) methods are proposed to tackle IMU-based HAR [7, 8], which requires abundant unlabeled data from the target domain. Existing UDA methods are usually limited to transferring knowledge between a single pair of source and target domains. Meta-learning (a.k.a learning to learn [9, 10]) achieves invariant feature learning across the source and target domains. It assumes the existence of common features despite domain gaps, and requires all domains to share the same activity set [11, 12, 13]. But in practice, missing classes are common in sensory motion datasets (see Table 2.1 for details). In addition, it requires updating the whole model with data from the target domain, which is inefficient. To address the limitations of meta-learning, we propose to separate invariant features across domains from domain-specific ones with a multi-task learning strategy. Target domain data is only used to update domain specific part of the resulting model making our approach more data efficient and able to handle varying number of classes across domains.

Learning from crowdsourced IMU data or data collected in the wild is more challenging. Besides the aforementioned domain shifts, labeling errors are abundant in such data. Learning with noisy labels (LNL) has long been studied in the machine learning community for computer vision tasks [14], but received little attention in body-worn HAR. We find that state-of-the-art LNL methods [15, 16] fail to handle labeling noise in HAR datasets due to inherent domain gaps and the violation of their fundamental assumption that a neural network tends to fit simpler and thus clean data in early training epochs. To address this issue, we present a one-step solution to jointly tackle noisy labels and domain shift.

Among approaches in the second category, various transformations are proposed to augment either raw sensor readings [17, 18, 19] or the extracted features [20]. For example, the works in [21, 22] generate IMU data with a data-driven approach such as generative adversarial networks (GAN). However, these approaches either require the availability of sufficiently real sensor data as their source or fail to generate meaningful IMU data. Instead, we consider another line of solution: transforming action data from other modalities to IMU data, a process called cross-modality simulation. This approach is motivated by the scarcity of IMU data for human motion analysis in contrast to the richness of other data sources. PAMAP2 [23], a benchmark dataset for HAR, consists of 8 subjects with only 59.67[1] minutes of samples per person. In contrast, AMASS [24], a motion capture (MoCap) dataset, includes 2420.86 minutes data and is still growing; not to mention online video repositories such as YouTube, Tiktok, which offer a practically infinite amount of action data.

1.2 Contributions

The overachy goal of this dissertation is to tackle the data scarcity problem in wearable sensor-based human motion analysis. Towards this goal, we make the following contributions as summarized in Fig. 1.1.

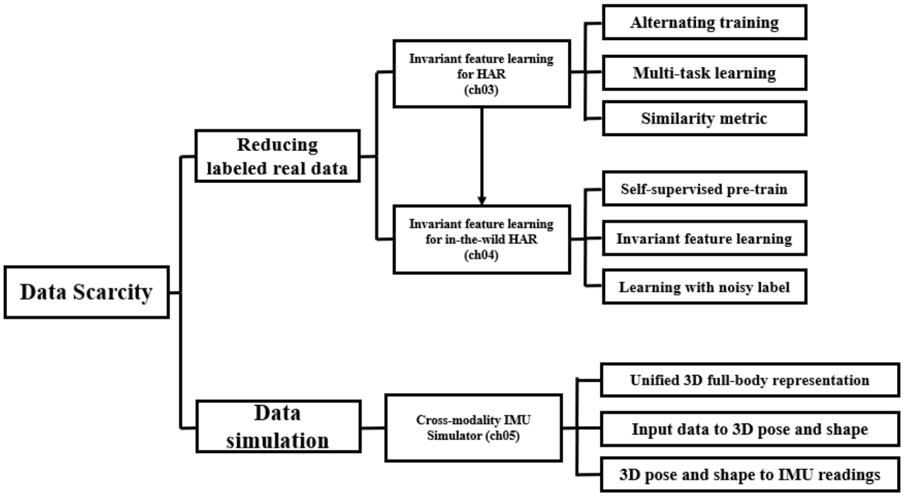

Data Scarcity
Reducing labeled real data
Data simulation
Invariant feature learning for HAR (ch03)
Invariant feature learning for in-the-wild HAR (ch04)
Cross-modality IMU Simulator (ch05)
Alternating training
Multi-task learning
Similarity metric
Self-supervised pre-train
Invariant feature learning
Learning with noisy label
Unified 3D full-body representation
Input data to 3D pose and shape
3D pose and shape to IMU readings

Figure 1.1: Main Contributions and Highlights.

We propose IFLF, an invariant feature learning framework for HAR. It handles various sources of domain shifts by extracting common features across multiple source

[1] This number will further shrink to 31.71 if we consider locomotion related activities only

domains. IFLF alleviates data shortage through shared feature learning from multiple source domains, and introduces a similarity metric to further reduce the amount of labeled data required from a target domain in model adaptation. The proposed method has its superior performance over multiple datasets when compared to a state-of-the-art meta-learning approach in sensor-based HAR tasks.

We develop VALERIAN, an invariant feature learning for in-the-wild domain adaptation method for wearable sensor-based HAR. To take advantage of existing in-the-wild IMU datasets with noisy labels, VALERIAN tackles label noises and learns the shared feature representation among multiple subjects in a one-step fashion. It uses self-supervised pretraining to learn good representations from abundant unlabeled data. Then IFLF is employed to extract common features among multiple training subjects. To combat noisy labels, early-learning regularization is introduced as a loss term reflecting the temporal ensemble of past inference results. We demonstrate that VALERIAN can significantly improve the performance of HAR tasks on synthetic and real-world noisy sensor datasets.

We design CROMOSIM, a multi-modality sensor simulator that synthesizes high fidelity virtual IMU sensor data using data from motion capture systems or monocular RGB cameras. It is the first work that utilizes the SMPL full-body tri-mesh as an intermediate representation for 3D human modelling, and thus enables IMU data simulation at arbitrary on-body positions. CROMOSim mitigates imperfection in intermediate body pose and shape estimations through a supervised learning approach, and achieves higher fidelity and superior performance in HAR tasks compared to SOTA IMU simulators. In addition, we are the first to empirically show the utility of simulated IMU data in HPE tasks using deep learning models.

In summary, IFLF addresses the problem when a body-worn HAR model is trained with a clean labelled and relatively small training set, and new unseen subjects or devices may be frequently added during inference. VALERIAN further considers the problem when the training set is noisy labelled. CROMOSim tackles the data scarcity more directly by taking advantage of MoCap or full-body video motion data.

1.3 Organization

This dissertation is organized into six chapters:

- **Chapter 1:** An overview of the motivation and key research problem is presented, followed by a brief introduction to the contributions and the thesis organization.

- **Chapter 2:** It introduces the background materials necessary for the understanding and reproduction of the dissertation. We discuss the principle of the inertial sensors and two important categories of applications in human motion analysis.

- **Chapter 3:** In this chapter, the learning of invariant features in wearable sensor-based HAR is proposed and evaluated using open and in-house datasets.

- **Chapter 4:** Technique on learning with noisy crowdsourcing datasets in HAR is presented and evaluated using controlled and crowdsourced datasets.

- **Chapter 5:** A deep learning-based cross-modality sensor simulator is proposed. Specifically, it is capable of simulating IMU data from either motion capture or video data.

- **Chapter 6:** The conclusion and future work, are provided in this final chapter.

Chapter 2

Background

Before presenting our solutions to address the data scarcity problem in wearable sensor-based human motion analysis, background and preliminaries will be introduced in this chapter. Specifically, we will explain how IMU sensors work and their data characteristics, the real-world applications of sensor-based human motion analysis, and popular public datasets in this area.

2.1 Inertial Measurement Units

2.1.1 Operational Principles

A modern micro-electro-mechanical system (MEMS) IMU is usually composed of three components with 9-DOF: an accelerometer to measure 3-axis acceleration, a gyroscope to measure 3-axis angular velocity, and a magnetometer to determine global orientations by measuring the 3D earth magnetic field [25]. An accelerometer has a mass attached to a spring which is confined to move along one direction and fixed outer plates. When an acceleration along the particular axis happens, the mass will move and the capacitance between the plates and the mass will change. This change in capacitance will in turn be measured, processed, and translated into a particular acceleration value. A gyroscope measures angular rates based on the Coriolis Effect [26]. The Coriolis acceleration, proportional to the angular velocity, is an apparent acceleration that is observed in a rotating frame of reference. Similar to the accelerometer, the displacement introduced by the Coriolis acceleration causes a change in capacitance which is measured, processed and translated into a particular angular velocity. Most MEMS magnetometers work with the Hall Effect [27]. If we have a conductive plate and set current to flow through it, the electrons would flow straight from one side to another. But the existence of a magnetic field will disturb the straight flow and thus the electrons would deflect to one side of the plate while the positive poles to the other side. Therefore, the voltage between these two sides depends on the magnetic field strength and its direction. Magnetometers are sensitive to magnetic fields generated by the appliances and metal objects in an indoor environment [28], which make them unsuitable for most HAR and HPE tasks. In this dissertation, an IMU sensor refers to accelerometer and gyroscope sensor unless specified otherwise.

2.1.2 Characteristics of IMU data

Generally speaking, inertial sensors can provide information on the pose of any object that they are rigidly attached to. It is also possible to combine multiple IMUs to obtain information about the poses of separate connected objects (as in HPE). However, low-cost MEMS sensors available of commercial-off-the-shelf devices are known for their noisy readings. When calculating positions by taking the double integration of accelerations, or orientations by integrating angular velocities, errors will be amplified and accumulated over time [29]. To handle noisy IMU measurements, existing solutions utilize handcrafted features in time and frequency domains, optimization-based smoothing and filtering, sensor fusion with other data modalities, or more recently deep learning-based data-driven models in human motion analysis. Before being input to a neural network model, multi-channel IMU data typically need to be temporally aligned, up or down-sampled, filtered and segmented.

2.2 Human Activity Recognition

Human activity recognition has been widely studied to enable health and fitness applications, such as mobility assessment, sports performance evaluation, rehabilitation monitoring and so forth. Activity or gesture recognition is also a crucial component of human-computer interaction interfaces to enhance user experiences [30]. One application scenario we consider in this dissertation is the assessment of older adult mobility. It is widely recognized that the life quality of older adults is closely related to their functional mobility status [31]. The fear of falls, and declined mental or physical health, together with visual or hearing impairments, are common causes that lead to degraded mobility. To facilitate continuous mobility monitoring of older adults, IMU sensor can be placed on one's torso or limbs. Through the recognition of locomotion-related activities and statistical analysis of their duration and timing, a healthcare provider can gain insights into the trajectory of mobility declines and determine suitable forms of intervention. A case study on mobility analysis for in-hospital patients is provided in Section 3.6.

From the technical point of view, HAR is a multi-class classification problem. From IMU data, statistical features calculated in individual data windows in the time domain exhibit distinct patterns for different human activities. Examples of such features as mean, standard deviation, variance, interquartile ranges, mean absolute deviation, correlation between axes, entropy, and kurtosis [32, 33]. Frequency domain features such as Fourier Transform and discrete cosine transform are also meaningful in distinguishing different human activities [34]. These handcrafted features in time and frequency domains combined with a shallow classifier were the mainstream strategies in HAR before the deep learning era [30]. The performance of such machine learning models relies heavily on the quality of extracted feature sets. Thus, the main drawback of such solutions is the need of domain expertise in designing feature sets for a specific HAR problem. In recent years, main stream approaches to HAR has gradually shifted to deep learning models due to their ability to extract semantic features from raw sensor signals and the impressive results achieved [35]. However, these (supervised) deep models require a large quantity of clean labeled data to generalize well in presence of subject, device and placement diversity in IMU data.

2.3 Human Pose Estimation

HPE provides geometric and motion information of the human body. It has found a wide range of applications in sports performance evaluation, motion analysis, augmented reality, virtual reality, entertainment and healthcare, etc. A pose is usually defined as body joint coordinates or joint angles between connected limbs. According to [36], HPE problems can be divided into two categories based on the representation of poses: 2D HPE and 3D HPE. The gold standard for 3D HPE is through MoCap systems. In recent years, markless 2D and 3D HPE have gained significant progress from images, video sequence and wearable sensor data [37, 38, 39].

HPE can be formulated as a regression problem, which learns a mapping from the input sensor data to joint angles or parameters of human body models. Early works for sensor-based HPE mainly focus on the signal processing aspects of position and orientation estimation, which usually involves a complementary filter or a variant of the Kalman filter to fuse kinematic constraints with the estimations from individual sensors [40, 41]. Similar to HAR, in recent years, neural networks were extensively used in HPE tasks. Researchers have succeeded in using a small number of IMUs to accurately estimate full-body poses. TransPose [42] achieves a $\sim$ 50 mm mean per joint position error (MPJPE) on TotalCapture dataset [43]. To achieve a MoCap level estimation accuracy on HPE tasks, IMU data is also fused with other data modalities such as single or multi-view camera videos and images [44, 45, 46]. But there is no free lunch indeed. Beside the large amount of data needed by deep learning, HPE tasks require accurate ground-truth pose information, which is typically attained through MoCap systems in lab environments. It is a laborious process to set up the data acquisition system, instrumenting human subjects, and synchronizing different data streams. Collecting data for HPE tasks in-the-wild with accurate ground truth data remains a research challenge.

2.4 Datasets

In this section, we summarize the characteristics of existing datasets for HAR and HPE tasks.

2.4.1 HAR Datasets

Based on how the data was collected, there are two types of IMU-based HAR datasets: The pre-scripted activity data collected under a controlled environment, and the freestyle motion data collected in real-world environments during activities of daily living. A brief summary of locomotion-related HAR datasets is in Table 2.1, with controlled datasets in the top rows and in the wild data at the bottom. In particular, MobilityAI datasets were collected from in-patient older adults from the Juravinski hospital. Its Phase I data were collected under the instructions of a physiotherapist while the Phase II data were mostly collected overnights during patients' hospital stay.

From Table 2.1, it is clear that there is no uniform data collection protocol in IMU-based HAR. The activity set, sensor placement and test subjects differ from one dataset to another. A noticeable trend is that in-the-wild HAR datasets have more subjects and longer average trial length in comparison to those controlled ones. In addition, the type and amount of data of different activities tend to be highly imbalanced in those in-the-wild datasets. This is in part because activities are not pre-scripted by researchers during data collection. Despite the relatively larger volume of in-the-wild datasets, controlled datasets are predominately utilized in evaluating HAR models in the research community as they provide more reliable ground-truth labels.

2.4.2 HPE Datasets

A brief summary of HPE benchmark datasets with IMU data is in Table 2.2. With the exception of 3DPW, all datasets are collected in indoor environments.

In Table 2.2, it is clear that the data scarcity problem is prevalent in IMU-based HPE as well. In addition, the data collection protocols are even more diverse in 3D HPE tasks, as sensor placement, activity types, and the availability of other sensing modalities differed from dataset to dataset.

Table 2.1: A summary of sensory HAR datasets. These datasets are selected based on the diversity of participants and popularity in the HAR research field. The number of activities listed here are locomotion-related only. "missing classes" indicates that the number of activity classes may vary from subject to subject. Length of trials is reported as an average of all subject.

Dataset	Sampling rate	#Sensors	Placement	#Activities	#Subjects	Missing classes	Balanced	Length (in mins)
PAMAP2 [23]	100 Hz	1	dominant side's ankle	8	8	Yes	No	31.71
USCHAD [47]	100 Hz	1	right hip	10	14	No	Yes	29.54
WISDM [48]	20 Hz	1	pant pocket	7	51	Yes	Yes	20.80
MobilityAI-PhaseI	50 Hz	4	waist	4	25	Yes	No	15.29
UTD-MHAD [49]	50 Hz	2	right wrist, right thigh	6	9	No	Yes	1.49
WHARF [50]	32 Hz	1	right wrist	7	17	Yes	No	7.78
MHEALTH [51]	50 Hz	2	chest, left ankle	10	10	No	Yes	9.51
RealWorld [52]	50 Hz	7	chest, forearm, head, shin thigh, upperarm, waist	8	15	No	No	71.75
MobilityAI-PhaseII	30 Hz	2	wrist+thigh or, wrist+ankle	4	30	No	No	1455.29
ExtraSensory [53]	40 Hz	2	wrist, not fix	6	60	Yes	No	6289.1

Table 2.2: A summary of HPE datasets with IMU data. Theses datasets are selected based on the diversity of motions and popularity in the HPE field.

Dataset	Sampling rate	#Sensors	Placement	#View	#Subject	GroundTruth	Length (in mins)
TotalCapture [43]	60 Hz	13	head, upper/lower back, upper/lower arms, legs and feet	8	5	MoCap	9.95
CMU-MMAC [54]	60 Hz	5	back, legs, arms	5	25	MoCap	23.32
MoVi [55]	120 Hz	17	head, shoulders, chest,upper/lower arms, hands, abdomen, thighs, shanks, feet	4	90	MoCap	4.40
TNT15 [56]	50 Hz	10	shanks, thighs, forearms, upper arms, chest and waist	8	4	Xsens [57]	1.12
3DPW [38]	30 Hz	6 to 17	not fix	1	7	algorithmic	4.05

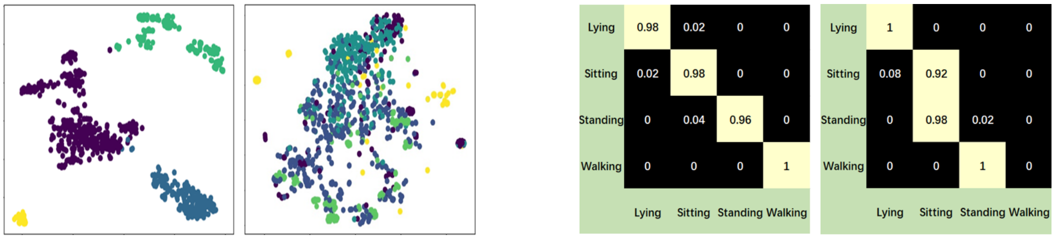
Chapter 3

Invariant Feature Learning Framework for Sensor-based Human Activity Recognition

3.1 Introduction

Human activity recognition (HAR) is the foundation to realize remote health services and in-home mobility monitoring. Although deep learning has seen many successes in this field, training deep models often requires a large amount of sensory data that is not always available [1]. For research ethics compliance, it often takes months to design study protocols, recruit volunteers and collect customized sensory datasets. At the same time, public inertial measurement unit (IMU) sensor datasets on HAR are typically collected by different groups of researchers following different experiment protocols, making them difficult to be used by others. The significant variability among human subjects and device types in data collection limits the direct reuse of data as well. Deep learning methods have poor generalization ability when testing data (target domain) differs from training data (source domain) due to device and subject heterogeneities (generally known as the *domain shift* problem). Fig. 3.1 shows the effects of cross-domain data variability. Fig. 1(a) visualizes features from subjects that are seen to the deep model for HAR (left) and as held-out data to the model (right), respectively. The features are well clustered when subjects are seen to the model and are inseparable for the unseen subject even though she performs the same group of activities wearing the same sensor at the same location. Fig. 1(b) demonstrates the effect of device diversity. The data is collected when a person performs several activities with devices A and B attached to the same on-body locations. A deep learning model is trained with device A's data. We find that despite its high inference accuracy on the testing data from the same device, the accuracy drops drastically on device B's data.

In addressing the aforementioned domain shift problems, a pooling task model



(a) (b)

Figure 3.1: Typical data variance problem caused by subject difference and device diversity. (a) depicts the impact of subject difference, where different colors in the t-SNE plots denote different activities. (b) shows the impact of device diversity. Left confusion matrix shows predictions on device seen to the model while right confusion matrix from a new unseen one. The prediction on unseen device's data is totally confused except for the 'lying' activity.

(PTM) that mixes data from different domains (e.g., subjects, devices) will have low discriminative power as it ignores the dissimilarity among the domains. On the other hand, a model trained solely on data from a specific domain requires a lot of training data as it fails to take the advantage of the similarity among different sources. Since collecting and labeling sensory data with sufficient diversity is time-consuming, it is impractical to train a task-specific model for each new subject or device encountered from scratch. A few previous works have investigated domain shifts caused by device and subject diversity in HAR. In [7] and [58], the problem is formulated as domain adaptation between a pair of participants or devices. However, in practice, HAR is rarely limited to transferring knowledge between a pair of domains, rather from a group of source domains (e.g., subjects, placements, or devices) to a target domain. Furthermore, unsupervised domain adaptation approaches trade-off their performance with data labeling efforts. For example, in [7], the authors report an F1-score ≤ 0.8 in testing compared to 0.92 from [59] with supervised learning on the Opportunity dataset[60].

It is expected that despite their differences, sensory data of the same activity from different subjects or devices intrinsically share common characteristics. A common assumption is the existence of shared representations among source and target domains. When tasks sampled from source domains can 'cover' the representation space, linear predictors built upon feature extractor for each task is sufficient for good prediction[61, 62]. Based on this assumption, we formulate the domain shift problem as a meta-learning problem with the aim to learn invariant features. By extracting features shared across domains and build task-specific layers for different source domains, the trained meta-model has better generalizability and can be adapted to a new target domain with few labeled data (a.k.a *fast adaptation*). To further reduce the amount of labeled data, we devise a metric to qualify the similarity of activities from different domains. Such a metric allows us to selectively collect new labeled data for activities exhibit high domain shifts. We have evaluated IFLF using multiple public HAR datasets and one in-house datasets collected from older adults. Extensive experiments demonstrate that a test accuracy $\geq 90\%$ can be achieved when only 10 seconds of sensory data per activity class is available from a target domain.

The main contributions of this work are:

- We present a deep learning framework that can handle various sources of domain shifts by extracting domain invariant features across multiple source domains.
- IFLF alleviates data shortage through feature sharing from multiple source domains.
- The proposed method achieves superior performance in extensive experiments over multiple datasets than the state-of-the-art meta-learning approach.

- A similarity metric is proposed to further reduce the amount of labeled data needed from a target domain for model adaptation.

The rest of this chapter is organized as follows. In Section 3.2 we discuss related work. Section 3.3 introduces an overview of the proposed invariant feature learning framework for HAR and a detailed description of methodology is in Section 3.4. We present the evaluation results on publicly available datasets and our own dataset in Section 3.5. A case study on the mobility assessment of real in-patient older adults is presented in Section 3.6. Finally, section 3.7 concludes the chapter and lists future directions of study.

3.2 Related work

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

3.2.1 Human Activity Recognition Models

Before the tide of deep learning, one popular approach to solve HAR problems is extracting a set of handcrafted features based on domain knowledge and training a shallow machine learning model[63, 64, 65]. In [65], two types of features are utilized, namely, time domain features (Mean, variance or standard deviation, energy, entropy, correlation between axes, signal magnitude area, tilt angle, and autoregressive coefficients) and frequency domain features (fast Fourier transform and discrete cosine transform coefficients). Accuracy of 99% and 92% is reported with a support vector machine model and a one-layer neural network model, respectively, built upon the features when classifying 16 activities. However, the effectiveness of handcrafted features can be highly activity specific.

With deep learning, features can be learned from data automatically. A convolutional neural network is usually incorporated as part of the feature extractor. Deep models are reported to achieve state-of-the-art results on many popular open datasets [59, 66, 67, 68]. However, deep learning has its own limitations. It requires a large amount of data to train and is sensitive to domain shifts. For example, the t-distributed stochastic neighbor embedding (t-SNE) visualization [69] of 2D features, a form of non-linear embedding for high-dimension data, in Fig.1(a) was originally 128 dimensions extracted by DeepConvLSTM [59]. Its predictive accuracy drops dramatically when applied to unseen subjects and devices. DeepSense[68] is a neural network architecture that is robust to domain shifts by merging local interactions of different sensory modalities into global interactions. However, it requires multi-sensor modalities, long data windows to achieve good performance, and is sensitive to class imbalance, making it unsuitable for transient or highly dynamic activities.

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

3.2.2 Domain Adaptation

As a sub-category of transfer learning approaches, domain adaptation mitigates the problem when the training data used to learn a model has a different distribution from the data on which the model is applied[70]. Differed by the information available for the target task, domain adaptation approaches can be further divided into supervised[71, 72, 73], semi-supervised[74] and unsupervised domain adaptation[7, 75, 8, 76].

Previous work on sensor-based HAR mostly falls in the category of unsupervised domain adaptation. Three types of domain shifts have been considered, namely, subject difference, device diversity and sensor location divergence. In [7], Soleimani and Nazerfard focus on mitigating subject differences when abundant unlabeled data is available in the target domain. A generative adversarial neural network (GAN) based solution is proposed to generate shared feature representation across a pair of source and target domains. Though not targeting HAR, the work [75] is relevant to mitigate domain shifts due to device diversity. It utilizes a cycle-consistent generative adversarial network (CycleGAN) to transform target domain data to a source domain, and then apply a classifier trained on the source domain. In [8], Akbari and Jafari propose a deep generative model to transfer knowledge between a labeled source sensor and an unlabeled target. A mechanism to annotate pseudo labels for a target sensor was proposed by majority voting and intra-class correlation in [76]. It is based on handcrafted features and a SVM model. Despite the popularity of unsupervised approaches, they trade-off the ability of learning invariant features that are robust to new domain with data labeling efforts. Thus, the performance tends to be noticeably inferior to supervised approaches. Also, the problem setup is limited to transfer knowledge between a pair of source and target domains which is limiting in practice.

3.2.3 Domain-Invariant Feature Learning

Learning invariant features across different domains can facilitate a better generalization of a deep learning model. Specifically, meta-learning (a.k.a learning to learn [9, 10]) is one approach to achieve this goal. Model-agnostic meta-learning (MAML) [77] introduces an episodic training paradigm with gradient-based parameter updating. It inspired meta-learning based HAR approaches in [11, 12, 13]. The first two target computer vision tasks, while MetaSense[13] is designed for sensor-based HAR and thus is the most relevant to our work. In MetaSense, Gong et al. proposed to sample tasks both randomly within each source domain and across source domains. It achieves good performance with few-shot learning tests, but one limitation is the task sampling method requires each source domain to have the same number of classes. This assumption does not always hold especially when the activity set involves difficult or intensive motions. Both IFLF and MetaSense are meta-learning approaches but operate under distinctive assumptions. IFLF assumes that the source domains and the target domain share invariant features. In contrast, MAML and its variants such as MetaSense assume the existence of weights that are only a few gradient steps away from the optimal ones in every domain. These assumptions lead to different ways of updating models with data from the target domain: MAML and its variants update parameters of the whole model while IFLF only updates the task-specific layers.

Multi-task learning also helps to extract invariant features by learning the knowledge shared by different tasks (or domains). In [78], the authors propose a personalized shallow model for HAR, with a test accuracy between 63.9% and 72.8% on different experiment settings. It also considers a subject-level similarity as transfer factor that controls model parameter update in a gradient step. The work in [67, 79] adopt deep learning methods. In [67], Peng et al. handle simple and complex activities with different task-specific layers on top of invariant features across them; whereas self-supervised learning is utilized through training an invariant feature extractor that is capable of extracting features behind various signal distortions in [79]. There are 8 types of manually added signal distortions (random noise, scaled, rotated, negated, horizontally flipped, permuted, time-warped, and channel-shuffled) involved, but limited by the types of predefined signal distortions, it reaches an overall accuracy between 75.55% and 88.55% on 6 open datasets.

To the best of our knowledge, ours is the first work to comprehensively deal with the domain shifts arising from multiple sources and data shortage problem. It differs from previous meta-learning approaches in that instead of updating all parameters of a meta-model in a gradient step, it trains a model in an alternating optimization manner [80] to separate the task-specific and domain-invariant knowledge. In IFLF, a small amount of labeled data is required in meta-test step. Comparing to unsupervised domain adaptation approaches, doing so results in better classification accuracy at low costs and the ability to handle missing classes in source or target domains.

3.3 Overview

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

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

There are two key components in the proposed learning framework: 1) a feature extractor $L_\theta : \mathcal{X} \rightarrow \mathcal{Z}$, where $\mathcal{Z}$ is the feature space and θ denotes the parameters of L, and 2) a group of task-specific networks: $S_{\phi^k} : \mathcal{Z} \rightarrow \mathbb{R}^C$, where k denotes the k-th task or domain, ϕ^k are the parameters of k-th task-specific layer S^k, and C is the number of classes in $\mathcal{Y}$. Accordingly, the loss function is also composed of a feature extraction objective ℓ_L and a task-specific objective ℓ_{S^k}, which will be detailed in Section 3.4. The output of a task-specific network is given by $\hat{y} = softmax(S_{\phi^k}(L_\theta(x)))$, where $softmax(z)_j = e^{z_j} / \sum_{k=1}^{C} e^{z_k}$, for $j = 1, ..., C$. IFLF is model-agnostic as L_θ and S_{ϕ^k} can be any reasonable neural network. An example neural network architecture of IFLF is shown in Fig.3.2.

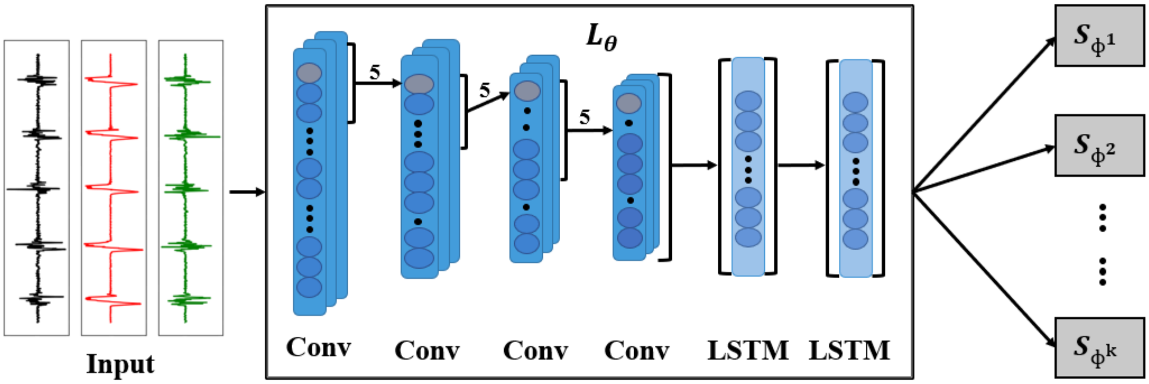


Figure 3.2: An example of IFLF model which employs 4 convolutional neural network (CNN) layers and 2 long short-term memory (LSTM) layers as L_θ, and K softmax layers as S_{ϕ^k} each corresponding to a domain in $\mathcal{D}_{src}$.

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

Algorithm 1 Invariant feature learning for domain adaptation

```
Require: Source domains $\mathcal{D}_{src} = \{D_k\}_{k=1}^{K}$, hyperparameters $\alpha, \beta$
Ensure: IFLF model with parameter $\theta$ and $\phi$
Random initialize $\theta, \phi = \{\phi^1, \phi^2, ..., \phi^K\}$
repeat
    Sample tasks $T = \{T_1, T_2, ..., T_K\}$ over $\mathcal{D}_{src}$;
    //Update $\phi^k$ with fixed $\theta$:
    for $k$ is 1 to $K$ do
        Freeze parameters of $\phi$ except $\phi^k$;
        $\phi^k \leftarrow \phi^k - \beta \nabla_{\phi^k} \ell_{S^k}(T_k, \theta; \phi^k)$;
    end for
    //Update $\theta$ with fixed $\phi$;
    $\theta \leftarrow \theta - \alpha \nabla_{\theta} \ell_L(T, \phi; \theta)$;
until convergence
```

alternating way.

3.4 Invariant Feature Learning Framework for Domain Adaptation

3.4.1 Invariant Feature Learning

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

Alternating Training If an IFLF model is trained by simply iterating among tasks sampled from D_1 to D_K, catastrophic forgetting may occur[81], namely, a model forgets previously learned tasks, and can only works properly on newly learned tasks. To avoid catastrophic forgetting, we adopt the alternating training strategy from [80], to update L_θ and S_{ϕ^k} separately. In each training epoch, we first freeze the parameters of the feature extractor layers, and update parameters of each task-specific layer with its respective data; then, we freeze parameters of the task-specific layers, and update the invariant feature extractor using all data from the previous step.

Feature Extractor By the merit of multi-task learning, L_θ is designed to generalize well across domains through the sharing of representations between related tasks [82]. Despite its model-agnostic nature, we adopt a simple architecture described in [59] that is shown to be effective for HAR (See Fig. 3.2). The network includes four convolutional neural network (CNN) layers and two long short-term memory (LSTM) layers. Because the application of convolution operator depends on the input dimension, we use a 1D kernel to convolve with one-dimensional temporal sequence (a.k.a the sensor signal) [83]. 1D temporal CNNs are widely used in the area of sensor-based HAR (see [35] for a detailed survey), the combination of CNN and LSTM is beneficial for acquiring contextural knowledge and extracting meaningful features for time series data.

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

$$\ell_L = \sum_{k=1}^{K} loss_L(T_k, \phi; \theta), \tag{3.4.1}$$

where $loss_L$ is a loss function calculated on each T_k with given θ and ϕ. Two types of loss functions are employed in this work. The first one is simply a categorical cross-entropy loss, defined as $loss_L = -\sum_{i=1}^{C} y_i log(\hat{y}_i)$ on data from each task k. An IFLF model that uses cross-entropy in the loss term in equation (3.4.1) is called a *basic multi-task learning model* (BMTL).

In light of encouraging features to be locally clustered according to class regardless of the domain, we introduce the second type of loss function which utilizes triplet loss [84]. To calculate the triplet loss, one needs to sample m triplets from raw data in T_k, and a triplet is $t = (x_a, x_p, x_n)$. The corresponding output of a triplet in the feature space $\mathcal{Z}$ is $L_\theta(t) = (z_a, z_p, z_n)$, where x_a denotes the anchor sample, x_p is the positive sample from the same class as x_a, and x_n is a negative sample from class other than x_a. Here, the objective is to maximize the distance between (z_a, z_n) and minimize the distance between (z_a, z_p). Since it is difficult to determine a fixed threshold in a high dimensional space that separate data points into groups that are sufficiently close (and thus belong to the same class), a triplet loss is suitable for learning features that maximizes inter-class distances while minimizing intra-class distances. We compute the triplet loss as:

$$loss_L(T_k, \phi; \theta) = \sum_{i=1}^{m} \max\{0, \left\| z_a^i - z_p^i \right\|^2 - \left\| z_a^i - z_n^i \right\|^2 + \epsilon\}, \tag{3.4.2}$$

where ϵ is a margin enforced between positive and negative pairs [85]. An IFLF model with a triplet loss is called *triplet multi-task learning model* (TMTL). Similar to BMTL, the loss function of TMTL is also calculated on each individual task. We then take the summation of losses over all source domains as the final objective function (3.4.1).

Task-specific Networks Under the assumption that if the shared feature generalizes well across all source domains, it will work on the target domain as well, L_θ

should be capable of exploring the entire latent space $\mathcal{Z}$ and extracting domain invariant feature. At the same time, a task-specific network S_ϕ^k should be simple to save the labor for fast adaptation, and be sparse to take only a subset (selected feature columns) from $\mathcal{Z}$ as its inputs. A lightweight architecture of a task-specific layer S_{ϕ^k} includes a fully connected layer with a softmax activation function. The task-specific objective function is defined as the sum of a categorical cross-entropy loss and an ℓ_1-norm regularization term as follows,

$$\ell_{S^k} = -\sum_{i=1}^{C} y_i^{(k)} log(\hat{y}_i^{(k)}) + \mu|\phi^k|_1, \tag{3.4.3}$$

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

3.4.2 Similarity-based Fast Adaptation

Aiming at further reducing the amount of labeled data required from the target domain for fast adaptation, a metric helps to identify the similarity or dissimilarity of motion patterns is required. We assume that if similar patterns are observed on an activity among all source domains, it is highly likely that the same activity in the target domain follows the same pattern as well. To quantify the similarity of two sensor signals, we propose a similarity metric in equation (3.4.4), which is calculated between data from the same activity across all source domains.

$$similarity_{i,j}^{c} = \sum_{i=1}^{K}\sum_{j=1}^{K}\frac{cov(p^i, p^j)}{\sigma_{p^i}\sigma_{p^j}}, \tag{3.4.4}$$

where σ is standard deviation, (p^i, p^j) =DTW(x_i, x_j) is the pair of warped signals from sensor readings x_i and x_j from subject i and j, c is the activity class and $cov(\cdot, \cdot)$ is the covariance. Dynamic time warping (DTW) [86] calculates the best match between two temporal sequences, which may vary in speed. Here we use it to align raw sensory readings to mitigate time shifts and speed divergence. The Pearson correlation coefficient calculated on a pair of warped signals in equation (3.4.4) provides a normalized similarity score that measures if activity c is performed similarly between a pair of participants. Data needs to be pre-processed (e.g., interpolated, noise filtered and normalized) before feeding to DTW. To eliminate the impact of misaligned sensor axis, we use the magnitude per sensor (e.g., an accelerometer or a gyroscope) as input to the similarity calculation.

Consider measurements from two sensors attached to two subjects (Subject 1 and Subject 2) performing the same activity. If the warped distance of the resulting measurements is small, this implies that the movement patterns are similar between the two subjects for the activity (despite possible differences in pace). Therefore, we can safely substitute Subject 1's data with that of Subject 2 and vice versa. After an IFLF model is trained, we no longer need to obtain labeled data from every class in $\mathcal{D}_{tgt}$. For activities that are considered similar across all source domains, we simply sample from the corresponding activity data in the source domains. These samples together with labeled data for remaining classes from the target domain are then used in updating the parameters in the task-specific layers while keeping the feature extraction layers unchanged during gradient descent. In the experiments, we find that a threshold ≥ 0.8 is suitable for distinguishing whether an activity is performed similarly among different subjects using the similarity measure defined in equation (3.4.4).

3.5 Evaluation and Results

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

3.5.1 Datasets

We consider three publicly available datasets to cover a wide variety of device types, data collection protocols, and activity classes to be recognized, and one in-house dataset that contains measurement data from multiple IMU sensors of different vendors on older patients. Important aspects of the datasets are summarized in Table 2.1 with brief descriptions listed below.

(i) **PAMAP2.** The Physical Activity Monitoring version 2 (PAMAP2) [23] is a dataset collected from one dominant ankle sensor (accelerometer and gyroscope) for 8 different activities, i.e., lying, sitting, walking, running, cycling, nordic walking, ascending stairs and descending stairs. Eight participants performed these activities freely without time constraints and have the option to skip some activities. Thus, there exist missing classes in some participants' data as well as unbalanced data samples across the classes. During data collection, sensors of the same model are instrumented on different subjects running at a sampling rate of 100Hz.

(ii) **USCHAD.** This dataset [47] is collected from 14 participants performing 10 types of locomotions (i.e., walking forward, walking left, walking right, walking upstairs, walking downstair, running forward, jumping up, sitting, standing and sleeping). All activities are performed by each subject in a controlled environment. A sensor (acclerometer and gyroscope) with a sampling rate of 100Hz is attached to the right hip of each participant. 5 data trials per activity were collected per participant, and the duration of each data trial varies.

(iii) **WISDM.** This dataset [48] contains a large number of subjects. Raw accelerometer and gyroscope data have been collected from a smartphone in each participant's pant pocket at a rate of 20Hz. There are a total of 51 test subjects performing 7 locomotion activities (i.e., walking, jogging, stairs, sitting, standing, kicking soccer ball, playing tennis) for 3 minutes apiece to get equal class distribution.

(iv) **MobilityAI-PhaseI.** The mobility analysis by artificial intelligence (MobilityAI) dataset phase one dataset is collected from 25 in-hospital patients whose ages are ≥ 65. The objective of collecting such a dataset is to monitor patients' mobility status during their hospital stay, and to quantify the effectiveness of an early mobilization protocol. To identify the most suitable device, four IMU sensors from different vendors are attached to patients' waists using elastic bands as shown in Fig. 3.3. Each subject performs four activities (lying for 5 minutes, sitting for 5 minutes, standing for 5 minutes and 20 meters walking) for mobility status assessment. The four devices utilized are MetaMotionR [87], Fitbit Versa [88], Mox One [89] and Actigraph [90]. All sensors are set to have a 50Hz sampling rate, and only accelerometer readings are captured. Due to data outage and the limited functional mobility of some participants, there exist missing classes in the dataset.

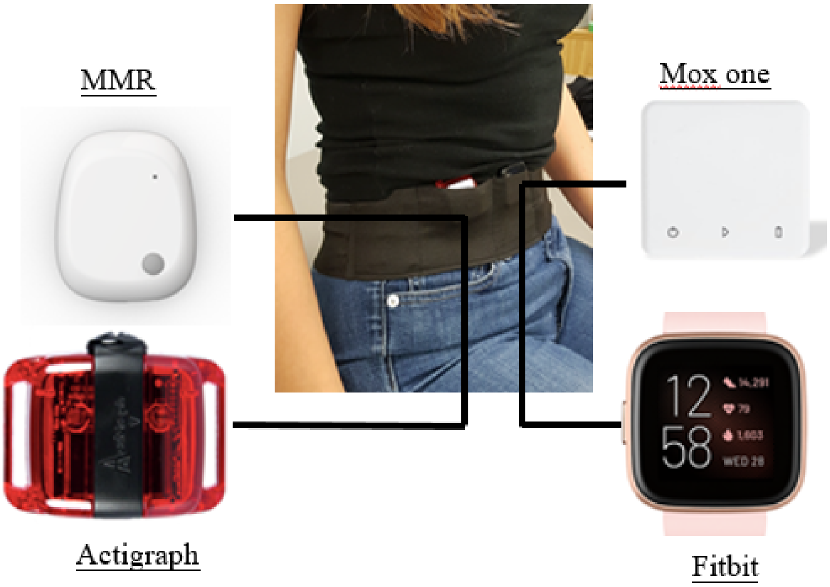


Figure 3.3: The sensors and their placement when collecting data for the MobilityAI-PhaseI dataset.

3.5.2 Implementation and Evaluation Process

In addition to BMTL and TMTL discussed in the previous section, we have also implemented four baseline models: a single-task learning model (STL), a pooling task model (PTM) and MetaSense [13]. STL is trained solely on the target domain for comparison with domain adaptation, whereas PTM is trained with mixed training data from all source domains, to highlight the domain shift problem.

Data Preparation Although deep neural networks can directly learn useful features from raw data [66], data preprocessing such as interpolation, noise filtering, normalization, and the division of sliding windows are still needed. A Butterworth low-pass filter [91] with a cut-off frequency of 10Hz is employed to remove high frequency noise from interpolated data. After low-pass filtering, we normalize the data, calculate similarity metrics, and then segment it into sliding windows with a fixed length of 2 seconds with 80% overlapping for all datasets. To eliminate the impact of different orientations of sensors in MobilityAI-PhaseI, we rotate the orientations of Actigraph, Mox one and Fitbit 3-axis accelerometers to be aligned with that of MetaMotionR.

Implementation The implementation of feature extractor follows DeepConvLSTM [59] for IFLF, STL and PTM models. It includes four layers of 1D CNN and two LSTM layers with 128 hidden units and a drop-out rate of 0.25 to prevent overfitting [92]. The CNN layers have 64 channels with kernel size 5 and stride 1. For a fair comparison with MetaSense, we also implement TMTL use the same network architecture as in [13] based on an open source implementation of MAML [93]. The feature extractor network has five CNN layers and two fully-connected layers, including 128 and 64 hidden units respectively. The reason for not including LSTM layers to MetaSense is two-fold: 1) MAML based approaches require a 2-steps update on layer parameters, keeping intermediate variable for calculated gradients without actual updating. But existing deep learning libraries combines gradient calculation with backpropogation for recurrent neural network parameters, leaving no API for gradient calculation only, and 2) as a model agnostic approach, it is interesting to investigate the performance of IFLF without LSTM as well. Tasks in MetaSense are sampled both within and cross different source domains, keeping activity labels consistent across all tasks.

For STL and PTM, the output layer corresponds to a fully-connected layer with a softmax activation function. Both models are trained with a RMSProp optimizor[94] at a learning rate of 10^{-3} and a decay factor of $p = 0.9$. The maximum iteration number is set to be 100. IFLF models utilize the aforementioned network structure as L_θ, the number of S_{ϕ^k} branches is determined by the number of source domains, and each S_{ϕ^k} may have a different output shape depending on the number of classes. IFLF models are trained with an Adam[95] optimizer at a learning rate of $10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$, and hyper-parameters $\mu = 0.8$. The batch size is set to 100 and the maximum number of training epochs is 30 with early-stopping. In each epoch, TMTL samples m pairs of (x_a, x_p) and m pairs of (x_a, x_n) to form m^2 (x_a, x_p, x_n) triplets from each source domain as task T_k. Similarly, n pairs of (x_a, x_p) and n pairs of (x_a, x_n) are sampled to form n^2 (x_a, x_p, x_n) triplets as validation set$(m > n)$. In most experiments, we set $m = 100$, $n = 10$ and hyperparameter $\epsilon = 0.4$. The hyper-parameters and optimizer of each model are the same across all datasets.

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

After the IFLF model is trained, we randomly sample a fixed test set from $\mathcal{D}_{tgt}$, and randomly select i data windows (of 2s length) per class from the remaining data as the training set to update a trained MetaSense model and an arbitrary S_{ϕ^k} layer of the IFLF model. This process is also called i-shot learning. An STL model is also trained on this training set and test accuracy is recorded for each model by gradually increasing i from 1 to 100. Each evaluation is repeated 5 times and the performance is reported with its mean and standard error if not specified.

3.5.3 Results

Evaluate with various domain shifts

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

Subject Difference Fig. 3.4 shows the averaged test accuracy with standard errors across all datasets. The average is computed over all subjects of each dataset. As demonstrated in Fig. 3.4, in terms of the overall test accuracy among the four models follows, BMTL is better than STL, and TMTL is better than BMTL when few data samples are available from $\mathcal{D}_{tgt}$. However, since STL is solely trained on the target

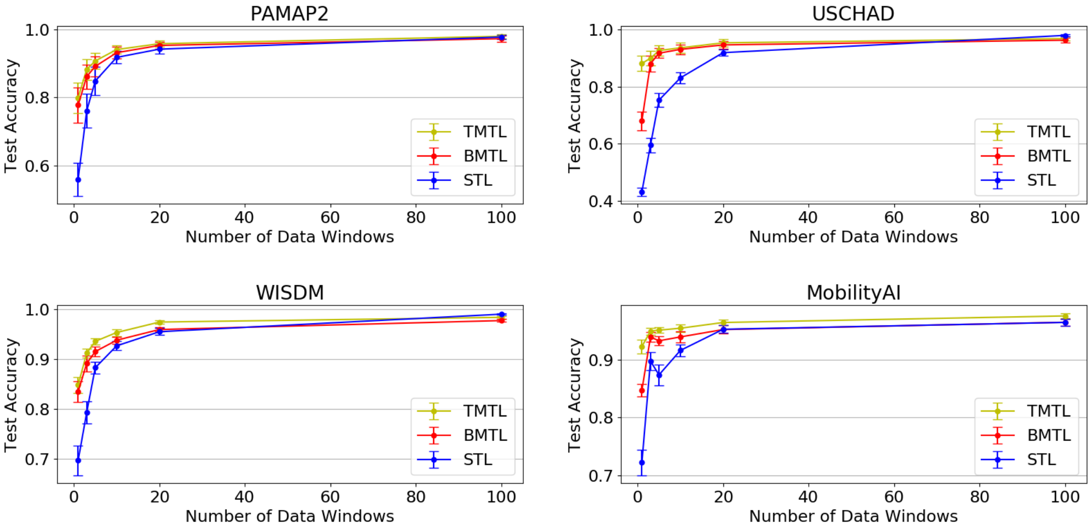
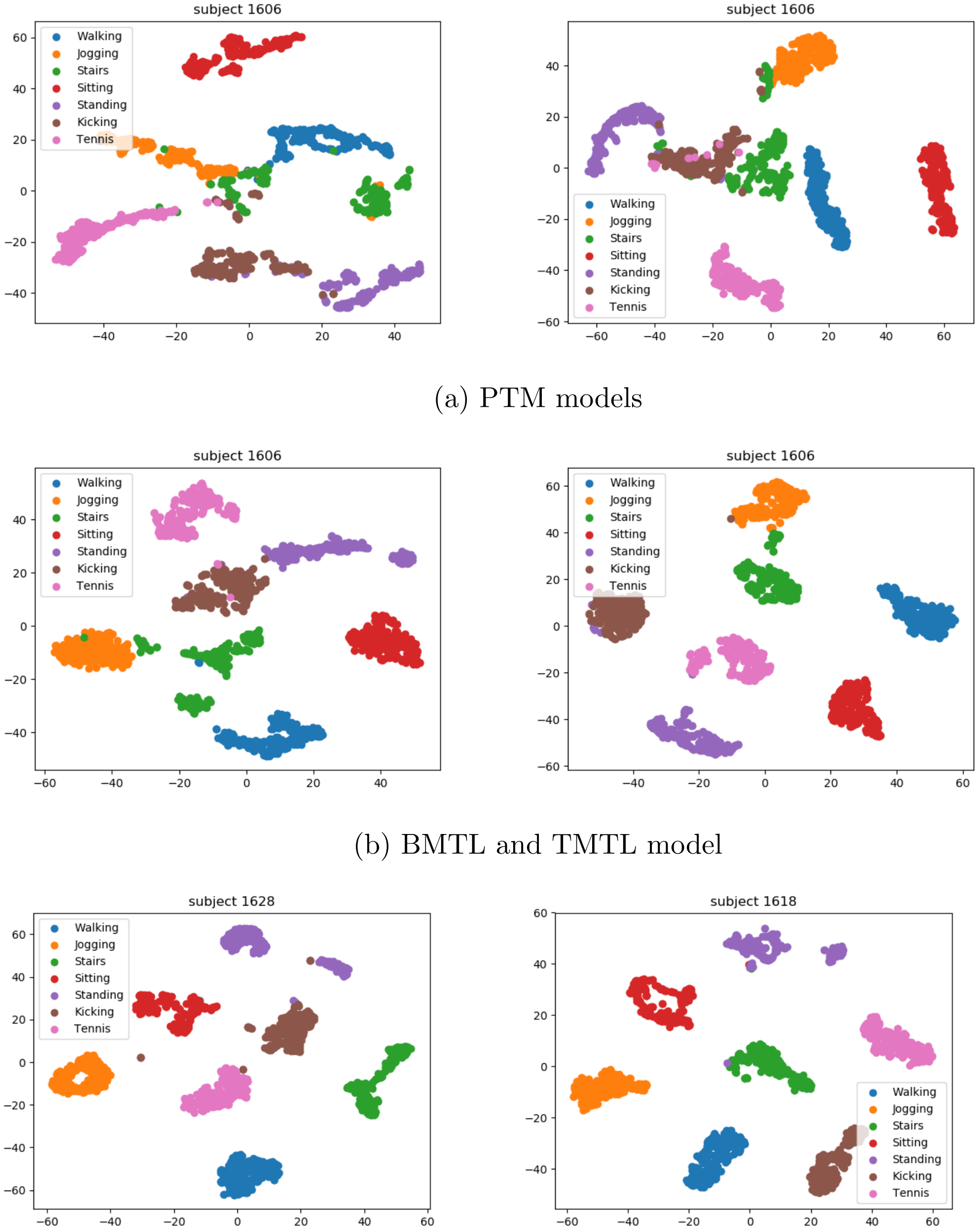


Figure 3.4: Evaluation on subject difference across all datasets when gradually increase the number of data windows per activity class from $\mathcal{D}_{tgt}$. The test accuracy and standard error are averaged across different subjects in leave-one-out experiments.

domain, when i is sufficiently large, its accuracy approaches 100% and tends to be better than both IFLF models. We also observe that with different subjects as the target domain, the converging rate of STL is dramatically different, an indication of subject differences.

To visualize the features produced by the methods, Fig. 3.5 shows the t-SNE of unseen subject's features produced by each model [69]. When generating these plots, we randomly pick a set of subjects as source domains (e.g., subject 1630 to 1646) and one subject as the target domain. The comparison is made among a PTM model, leaving this subject out in the BMTL model and TMTL model. To demonstrate the generality of the results, plots from two PTM models are presented with different random splits of the training set for each source domain. Although the figures are generated from the WISDM dataset, similar observations can be made for



(a) PTM models

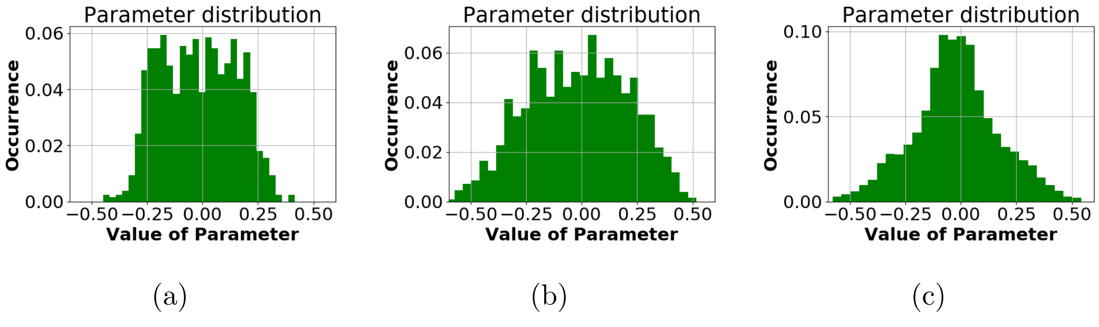
(b) BMTL and TMTL model

(c) Features extracted by TMTL model

Figure 3.5: t-SNE visualization of the learned representations. We visualize the features from each model with the output of L_θ by projection them on 2D space. This example is generated by different models with subject 1630 to 1646 as source domains. (a) are PTM models; (b) left is BMTL and (b) right is TMTL; (c) are from randomly selected subjects as target domains other than the one in (a) and (b).

other datasets. Also, we examine the existence of invariant features across domains by extracting features from unseen target domains.

From Fig. 3.5(a) and (b), it is clear that the features for different activities generated by PTMs are entangled regardless of the splits between training and validation set. BMTL improves the separation among activities to some extent, whereas TMTL generates a set of features with clear clustered structures and large margins. Fig. 3.5(c) further demonstrates the existence of invariant features across domains as features extracted from unseen target domains are well-clustered and linearly separable. Due to space limit, only 3 subjects are shown in Fig. 3.5, but other subjects exhibit similar characteristics. This observation also explained why the fast adaptation can be made on any trained S_{ϕ^k} layer of BMTL and TMTL. As the features are well separated, different choices of the task-specific layer for parameter update have little impact on the performance. However, as the amount of data is quite limited in the fast adaptation step, initializing L_θ randomly will impair the performance of IFLF. But even in this case, we find the IFLF models still work better than STL. To further illustrate the advantage of TMTL over the other two models, the sparsity of



(a) (b) (c)

Figure 3.6: Comparison of the parameter ϕ's distribution for different models. From (a) to (c) are: STL, BMTL, TMTL. X-axis is the value of parameter, y-axis is the normalized occurrence.

the parameters (ϕ) of the learned task-specific layer is compared. In Fig. 3.6, we plot the distribution of ϕ in the three models trained on WISDM dataset.

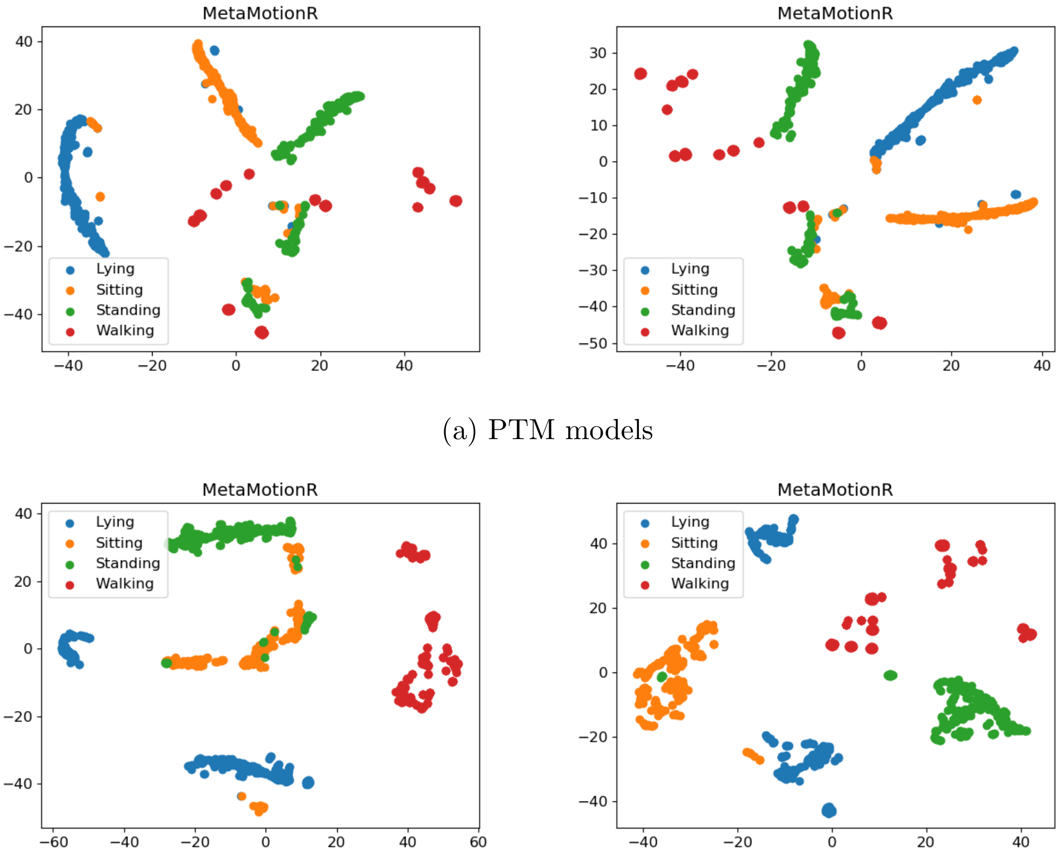
As shown in Fig. 3.6, the parameters in STL are roughly a uniformly distributed between -0.4 and 0.4. In comparison, the parameters of BMTL follow a zero mean Gaussian distribution but with a large variance. Lastly, the majority of TMTL parameters are centered around 0 with a noticeably smaller variance (than that of BMTL). The sparsity of task-specific layer's parameters indicates the easiness of separating the generated feature representations.

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

In following experiments, data from Actigraph, Fitbit, Mox One constitute the source domains, while MMR device's data is selected as D_{tgt}. One data window per activity sampled from D_{tgt} is utilized. To demonstrate the presence of domain shifts between different devices, besides the STL, BMTL and TMTL models, we further present the confusion matrix from PTM trained with data from Actigraph, Fitbit and Mox one but tested on MMR.

From the confusion matrices in Table 3.1, it is clear that BMTL benefiting from invariant feature learning outperformed STL, while TMTL rarely misclassifies any activity in the dataset. Table 3.1(d) shows that PTM is incapable of learning robust features that generalize well to the target domain.

Fig. 3.7 visualizes features extracted by PTM (the two plots in Fig. 3.7(a) are generated from different random splits of the training data), BMTL and TMTL models. Similar to the case of subject differences, we observe that the PTM models fail to extract separable features from MetaMotionR data while BMTL does a better job, but the resulting features are still not well-clustered. In contrast, TMTL gives rise to features with clear boundaries in the feature space and clustered structures.



(a) PTM models

(b) BMTL and TMTL model

Figure 3.7: t-SNE visualization of the learned representations. This example is generated by different models on MobilityAI-PhaseI dataset. (a) are PTM models; (b) left is BMTL and (b) right is TMTL.

Table 3.1: The experiment on sensor diversity. Confusion matrices are generated with only 1 data window per activity involved.

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

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

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

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

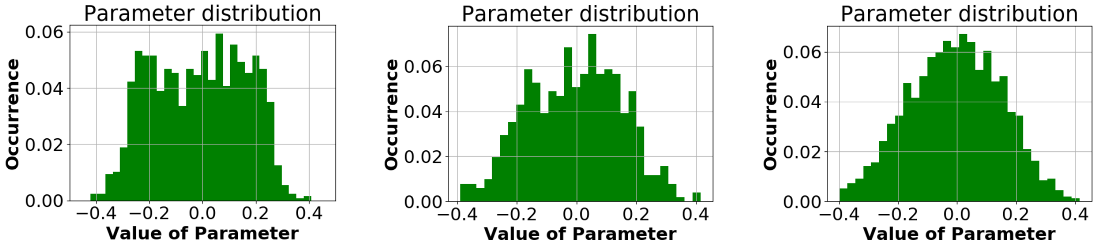
(c) The confusion matrix of the TMTL model fast adapted with MetaMotionR data.

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

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

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

Fig. 3.8 depicts the distribution of parameters ϕ in the task specific layers of each model. Similar to Fig. 3.6, we observe that TMTL has the most sparsity, followed by BMTL, whereas STL leads to the least sparsity. The evaluation on device diversity further demonstrates the IFLF models' capability of learning invariant features across domains.



(a) (b) (c)

Figure 3.8: Comparison of the parameter ϕ's distribution for different models. From (a) to (c) are: STL, BMTL, TMTL.

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

Subject 6, 7 and 8. Fig. 3.9 compares the performance when both the subject

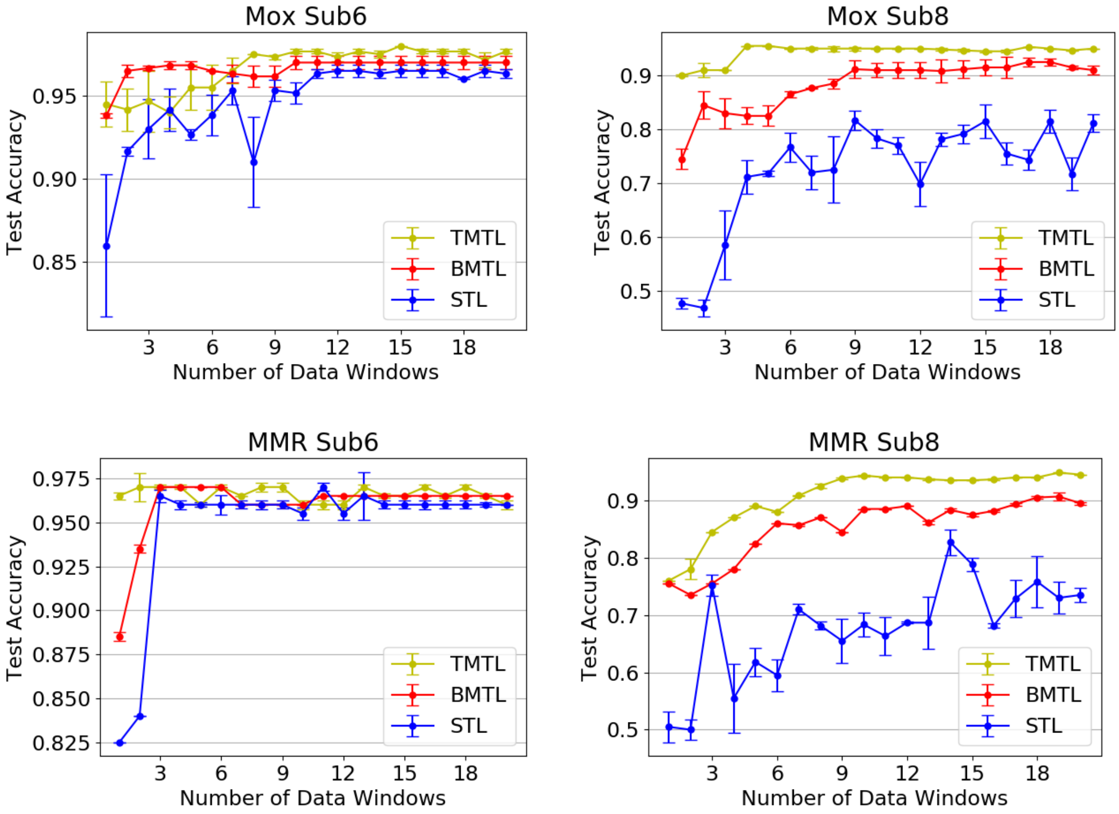


Figure 3.9: Combinations of different sources of diversity (Both the test subject and the test device are not included in the training data for IFLF models). The test accuracy and standard error are averaged across 5 experiments by randomly sampling data windows from the target domain.

and device are unseen to IFLF models. In this case, both BMTL and TMTL have better performance than STL, especially when the amount of labeled data from the target domain is small. Compared to Fig. 3.4, we observe large gaps between TMTL and STL test accuracy. This can be attributed to the significant diversity among different devices. Furthermore, by comparing the results for Subject 6 and 8 for the same device (e.g., Mox One or MMR), we find that Subject 8 appears to have larger differences from subjects in the training set than Subject 6. Despite such differences, TMTL consistently outperforms BMTL and STL. Meanwhile, large standard errors are observed with STL as it is heavily dependent on the training set.

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

Comparison with MetaSense

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

From Fig. 3.10, it is clear that in 19 out of 20 cases tested, TMTL performs better than or comparably as MetaSense. The advantage of TMTL is more prominent when few data samples are available from the target domain. The superior performance of TMTL over MetaSense for very few shot learning is due to the fact that MetaSense needs to update the entire model whereas TMTL only updates a task-specific layer. The latter approach is more data efficient and less likely to over-fit. However, as the number of available data samples from the target domain grows, the performance of MetaSense is comparable to or even slightly better than TMTL with sufficient labeled

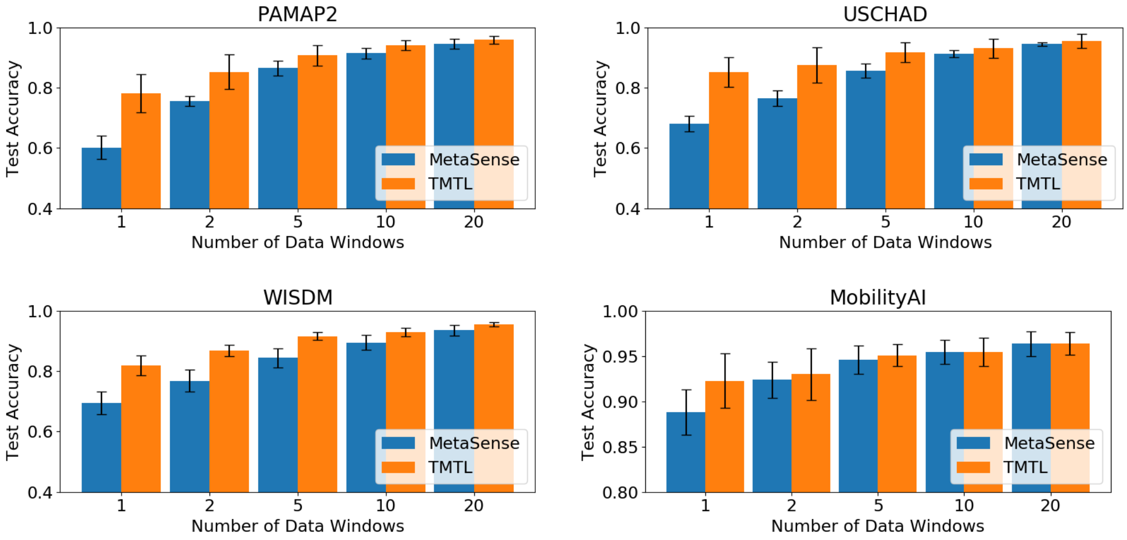


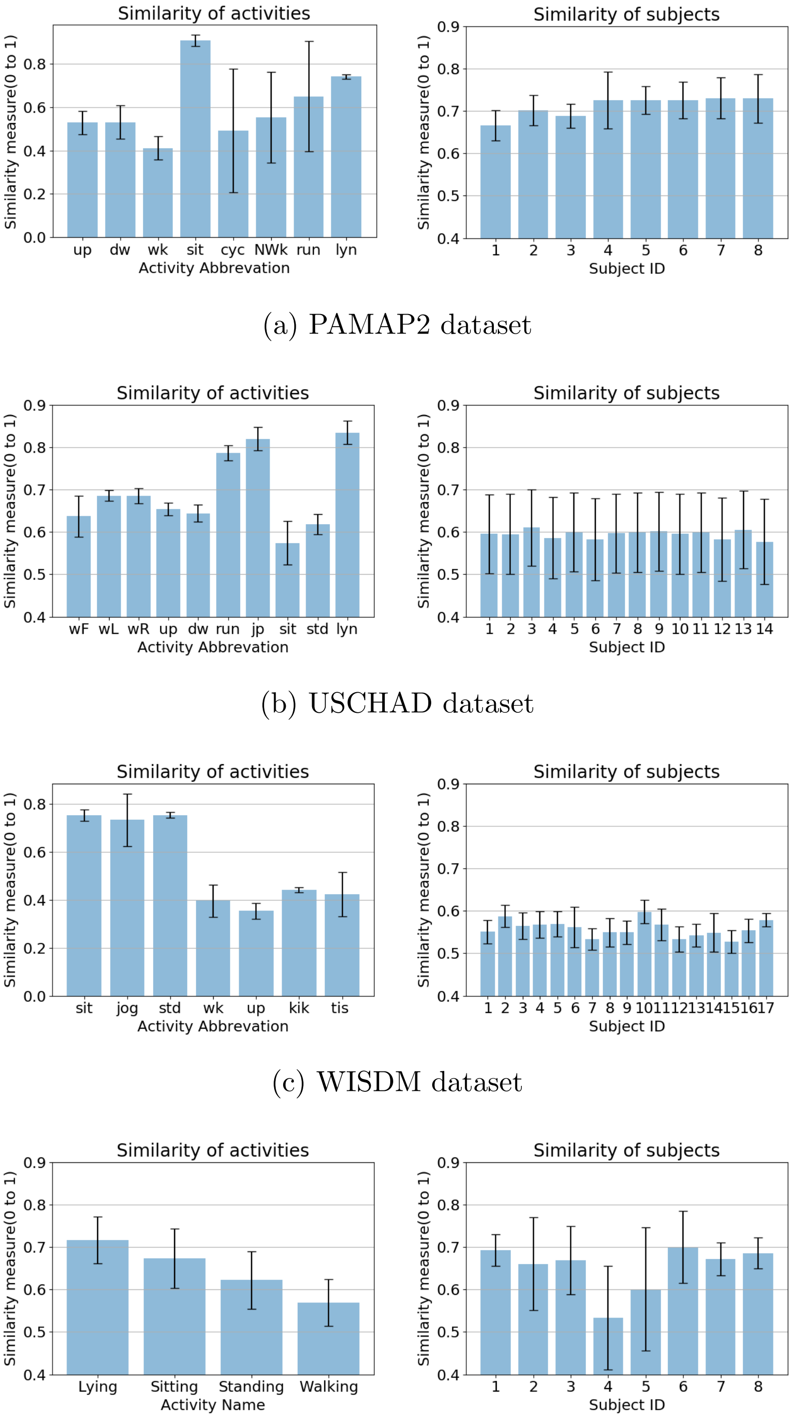
Figure 3.10: Comparison between TMTL and MetaSense on different domain adaptation tasks. Results are reported for 1, 2, 5, 10, 20-shots (data windows) per activity class from $\mathcal{D}_{tgt}$ with average test accuracy and standard error.

data from the target domain.

Compared to the TMTL with LSTM in Section 3.5.3, the performance of TMTL with LSTM drops by an average 2.35% for 1-shot learning across all datasets. However, as the number of shots increases to more than 10, the performance gap is negligible ($< 0.5\%$). This fact further demonstrates that IFLF is a model agnostic method and can be used in conjunction with any suitable network architecture for feature extraction.

Similarity Metric and Fast Adaptation

Using the similarity metric defined in Section 3.4.2, in this section, we first compare inter-subject and intra-activity similarity. The purpose of this study is to understand whether there exist subjects with similar movement patterns in all activities, and whether there exists an activity with little inter-subject variation. Secondly, we



(a) PAMAP2 dataset

(b) USCHAD dataset

(c) WISDM dataset

(d) MobilityAI dataset

Figure 3.11: The evaluated similarity measure on each datasets. Plots on the left are similarity of activities, while plots on the right are similarity of subjects.

evaluate the use of intra-activity similarity in further improving data efficiency in fast adaptation. For activity similarity (left column in Fig. 3.11), we compute the averages and standard deviations of similarities of the same activity across different subjects. The right column in Fig. 3.11 shows the similarity scores between one subject and the others averaged over matching activities.

From Fig. 3.11, it is clear that some activities have higher similarity scores than the others. For example, in the PAMAP2 dataset, 'sitting' appears to be similar across different subjects, while 'walking' has the least similarity. This indicates diverse postures during walking but less variation during sitting in this dataset. On the other hand, when examining subject similarities, we find that some degree of similarity exists across almost all subjects (e.g., with a similarity score > 0.5). However, no two subjects perform activities the same way (e.g., the maximum similarity score is below 0.8). An exception is observed in the MobilityAI dataset, Subject 4 and 5 have noticeably lower similarity from others and larger standard deviations. It is because these two people have mobility issues and have to stand or walk with a rollator. It is expected that larger diversity exists among older adults with different underlying physical conditions. Homogeneity is more pronounced among younger populations as evident from the first three datasets. Another interesting observation is that depending on sensor placements, participants and the protocol of data collection, the same activity may have different cross-subject similarity in different datasets. For example, 'lying on bed', 'sitting' and 'standing' have noticeably different similarity scores across the four datasets.

Next, we investigate whether target domain data can be safely replaced by data from source domains for certain activities in fast adaptation. We first sample 10 labeled data windows from each activity from $\mathcal{D}_{tgt}$ to perform fast adaptation on a trained TMTL with the procedure described earlier. Then, the ten data windows of activities with top 3 similarity scores are replaced with data randomly sampled from $\mathcal{D}_{src}$ to update the TMTL model. The performance metric used is $recall = \frac{tp}{tp + fn}$, as it reveals whether a single activity is wrongly classified. A non-decreasing recall as more and more sampled data from the target domain is replaced by source domain data implies that doing so has little impact on the performance with less data collection efforts. The results for different datasets are shown in Fig. 3.12.

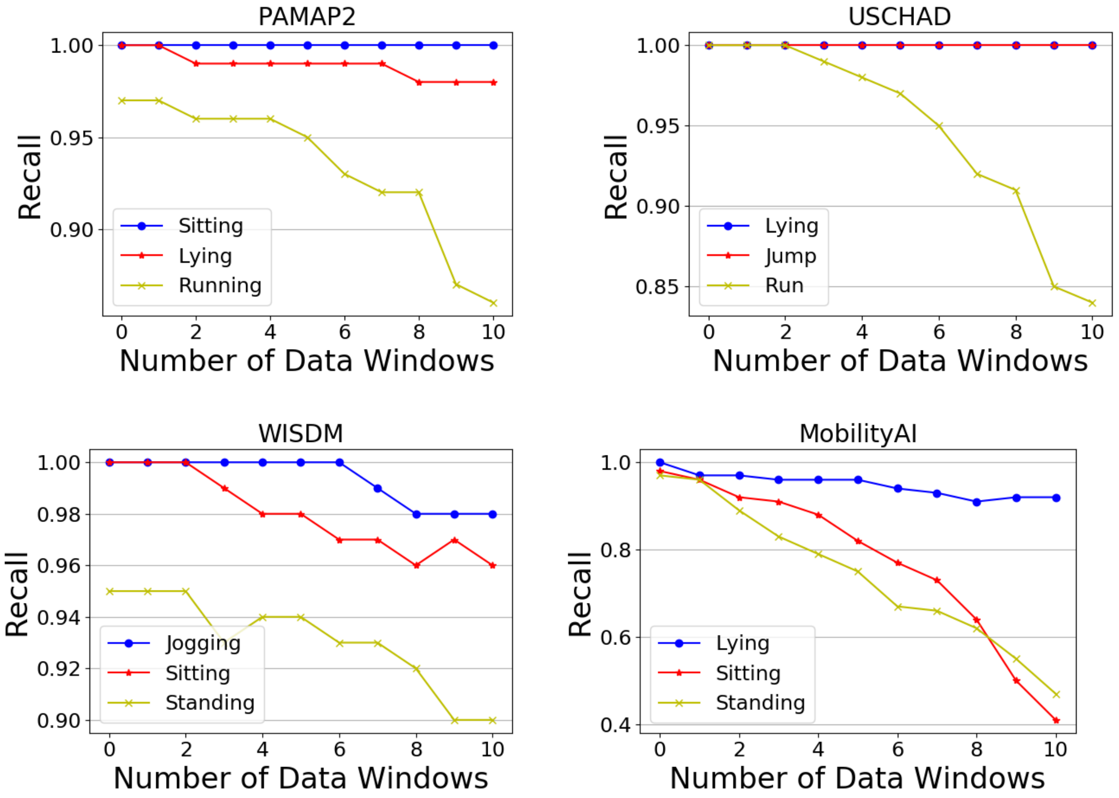


Figure 3.12: The evaluation of applying similarity metric in the fast adaptation step. Activities with top 3 similarity scores are replaced in each dataset.

As evident from Fig. 3.12, ‘sitting’ from PAMAP2, ‘lying’ and ‘jump up’ from

USCHAD can be safely replaced with marginal impacts on the recall. This result is in accordance with the similarity scores presented in Fig. 3.11 as all these three activities have similarity score ≥ 0.8. Although 'lying' appears in three datasets, it can only be safely replaced in the USCHAD. In other words, by sampling activities with high similarity scores and small inter-subject variance from source domains, we can improve data efficiency by 12% to 20% on top of the reduction from fast adaptation.

3.6 A Case Study on Older Adult Mobility

Apart from the evaluations on public datasets collected under controlled settings, we also conduct a case study on in-patient HAR dataset (MobilityAI-Phase I and Phase II) to gain more insights on IMU-based human motion analysis. Part of the work on Phase I is in Section 3.5 and a portion of work on Phase II is reported in [96].

3.6.1 Dataset

A high level summary of the dataset is given in Table 2.1 in Chapter 2. More specifically,

1. Phase I: A detailed description is in Section 3.5.1. The objective of collecting this dataset is a feasibility study as well as picking the best sensor device for motion analysis on older adults.

2. Phase II: Actigraph sensor [90] with a 30 Hz sampling rate accelerometer was utilized. MobilyAI-Phase II includes 30 subjects, 24 were female (80%) and the

average age was 81.4. From each test subject, two mobility assessment data trials were collected under the supervision of a physiotherapist, including five activities: lying, sitting and standing for 5 minutes, a 30-meter walk and time-up and go. Three IMUs were attached to the dominant side thigh, wrist and ankle, and the two trials were separated by a 24 hours time interval. A 24-hour freestyle data trial was also collected from the subjects, they were asked to wear either a combination of wrist+thigh sensors or wrist+ankle sensors during the collection. There is no 3rd party observation or self-reported activity labels for the freestyle trials.

3.6.2 Models and Evaluation Process

As there are no ground-truth labels for the freestyle data trials, quantitative study is mainly conducted on data trials collected in a controlled environment. PTM models were trained on different sensor placements to find the best device combination and investigate the gap between controlled trials and in-the-wild trials.

3.6.3 Results

Evaluation on Phase I dataset

In this evaluation, we compared the model performance on data collected at different placements with Actigraph and MetaMotionR, for picking the best sensor placement and sensor device. The result of MetaMotionR is in Fig. 3.13 and that of Actigraph is in Table 3.2.

The observations from Fig. 3.13 and Table 3.2 are consistent. No apparent performance gap is observed between Actigraph and MetaMotionR devices. Also,

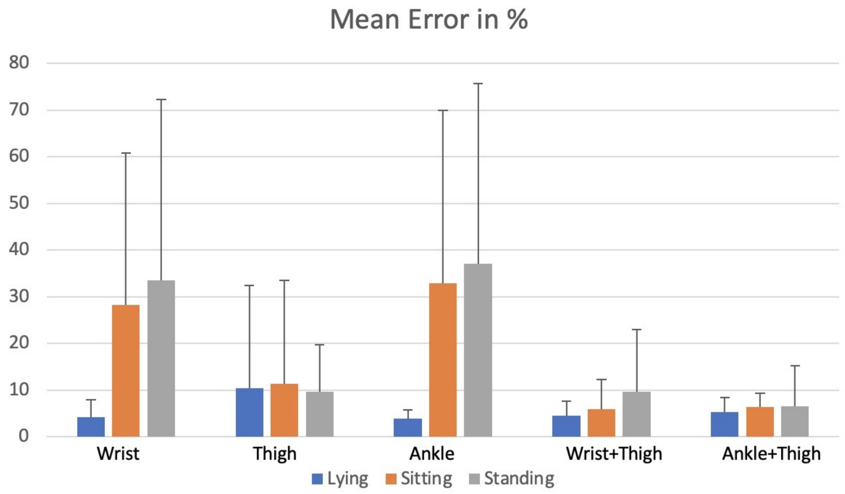


Figure 3.13: Mean error of the MetaMotionR posture detection across all sensor placement

Table 3.2: Test accuracy of each sensor placement.

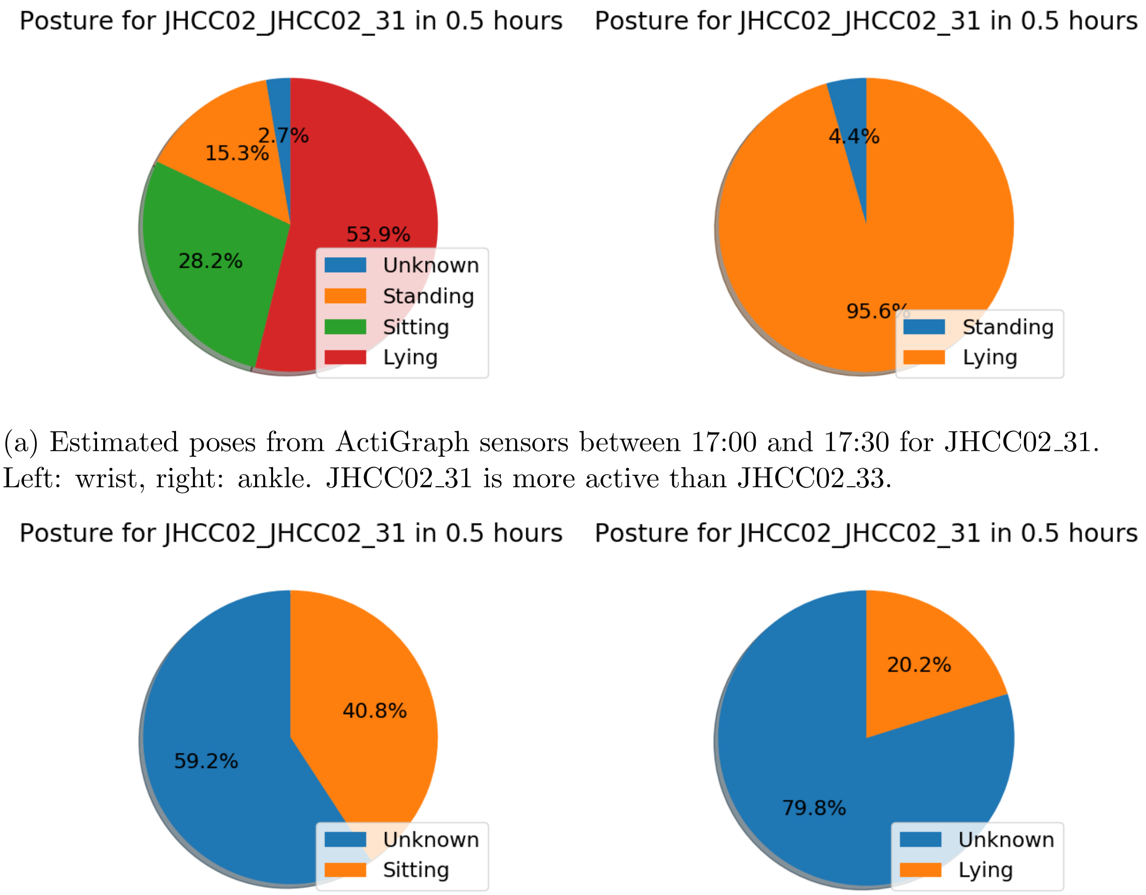
	Ankle	Thigh	Wrist
Test Accuracy	77.95%	85.71%	72.80%
	Ankle+Wrist	Ankle+Thigh	Wrist+Thigh
Test Accuracy	86.14%	87.93%	87.89%
	Ankle+Thigh+Wrist		
Test Accuracy	91.11%		

they follow the fact that the wrist is one of the most active parts of the human body and its motion can be largely irrelevant to body posture. Sensor readings from the wrist sensor are expected to be noisy. But the labeled data trials were collected under a controlled environment, thus make it possible to achieve $> 70\%$ accuracy on posture detection task. However, it is presumably not the case for freestyle data collection. Like the wrist, people may have more free motions at the ankle than that at the thigh. Hence the thigh sensor provides the best accuracy when there is only a single sensor as the data source. The observation from a single sensor that $thigh > ankle > wrist$ still holds for a combination of two sensors. So when we combine wrist and thigh sensors, the accuracy is better than that of wrist and ankle, whereas ankle and thigh accuracy is the best among them. When using all three Actigraph sensors, the best accuracy was achieved, as 91.11%.

Evaluation on Phase II dataset

As the 24-hour data trials have no activity labels, we cannot directly evaluate the trained neural network model on them. But based on an assumption that people are asleep at night and more active in the day, we can obtain some qualitative results of the model performance selected time spans. e.g., 2:00 am to 2:30 am versus 5:00 pm to 5:30 pm. Here we show the results from two subjects JHCC02_33 and JHCC02_31 who have different level of mobility–JHCC02_33 can walk independently with walker while JHCC02_31 requires assistance to walk by one person and a walker. Also included are the results due to sensor placement (wrist+thigh vs.wrist+ankle) during 24-hour data collection. It is expected that JHCC02_31 should be less active than JHCC02_33 during non-sleeping time. The results are shown in Fig. 3.14.

From Fig. 3.14, it is clear that the predicted activities are inaccurate using the wrist sensor only, as there is no lying activity in prediction. Upon a close inspection at the data, we find the IMU data from the wrist sensor is very noisy and do not exhibit clear pattern even during periods when the subject was meant to be still. This can be in part explained by many interfering activities that hands can perform during locomotion. Recall that the model was trained using labeled data trials collected under the supervision of a physiotherapist, and thus a participant was asked to lying or sitting, it is more likely to keep his/her hands still. Interfering activities are one



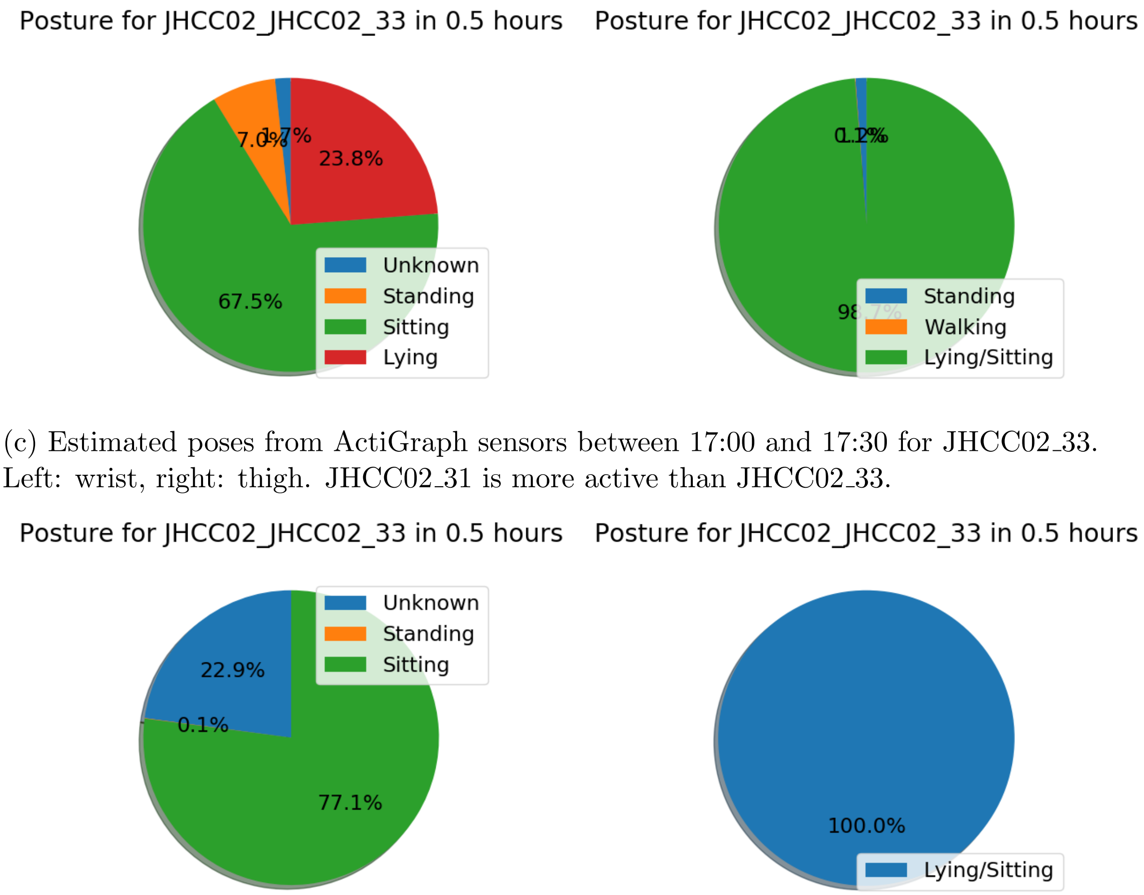
(a) Estimated poses from ActiGraph sensors between 17:00 and 17:30 for JHCC02_31. Left: wrist, right: ankle. JHCC02_31 is more active than JHCC02_33.

(b) Estimated poses from ActiGraph sensors between 02:00 and 02:30 for JHCC02_31. Left: wrist, right: ankle. Predictions from the wrist are totally wrong with no 'Lying' pose detected.

source of domain gap between controlled and in-the-wild datasets that hinders the generalization of the model trained on one dataset.

3.7 Conclusion

In this chapter, we presented an invariant feature learning framework based on meta-training and multi-task learning paradigm to effectively address domain shifts



(c) Estimated poses from ActiGraph sensors between 17:00 and 17:30 for JHCC02_33. Left: wrist, right: thigh. JHCC02_31 is more active than JHCC02_33.

(d) Estimated poses from ActiGraph sensors between 02:00 and 02:30 for JHCC02_33. Left: wrist, right: thigh. Predictions from the wrist are totally wrong with no 'Lying' pose detected. The thigh sensor detected 'Lying' pose but cannot distinguish it from 'Sitting'.

Figure 3.14: ActiGraph predicted postures for unlabeled data during selected time.

and data shortage in HAR. As demonstrated in Section 3.5.3, IFLF has been shown to work efficiently in few-shot learning, especially when the number of shots are few (1 or 2 shots). A $>10\%$ performance margin has been observed when compared to MetaSense under such condition. Also, the proposed TMTL model implicitly handles class imbalance and class missing problems as well. A similarity measure was proposed to further reduce the amount of data required in fast adaptation step. A case study is conducted to better understand the IMU-based human motion analysis on older people.

Chapter 4

Sensor-based HAR with Noisy Crowd-sourced Dataset

4.1 Introduction

Wearable sensor-based human activity recognition (HAR) has gained a lot of interests recently due to the pervasiveness of wearable sensors such as inertial measurement units (IMUs) in smartphone and smartwatch devices and its many applications in fitness and health monitoring [68, 97, 35, 98]. With the increasing adoption of deep neural network models in HAR tasks, there is a need to acquire a large amount of well-curated and labeled sensory data to train such models. Unfortunately, the majority of public HAR datasets are from controlled settings where subjects are asked to perform prescribed activities in lab environments. They typically contain a small collection of subjects and activity types over limited periods of time. For example, PAMAP2 [23], a popular dataset for HAR, only includes eight subjects with 59.67 minutes of measurements per subject. Furthermore, data collected from controlled settings often have very different characteristics from those of freestyle motions in naturalistic environments [3].

Collecting wearable sensor data in the wild faces its own set of challenges. Arguably, the biggest difficulty is to label such data accurately [4]. Recalls from one's memory are known to be notoriously unreliable [5]. Labelling wearable data by observing signal patterns requires extensive domain knowledge and experience since sensor readings are impacted by not only activity types but also subject characteristics, on-body positions and sensor orientations. A mainstream method to label such data is to resort to another human-interpretable modality such as visual or audio recordings and determine the labels manually post hoc. Unfortunately, labels obtained this way are still error-prone due to mis-synchronization across different modalities, human errors or missing data (e.g, occlusion in vision data). As the first contribution of the work, *we examine two datasets collected in naturalistic settings to understand the extent and characteristics of noisy labels.*

Learning with noisy labels (LNL) has long been investigated in the machine learning community with many effective methods being proposed for computer vision tasks [14]. However, through an empirical study, we find that one mainstream LNL method fails to achieve good accuracy and sometimes cannot converge at all. In-depth analysis reveals that the root cause is the violation of a fundamental assumption in this and similar methods that a model trained from noisy data in early training epochs tends to have much higher confidence in correctly labeled data than wrongly labeled data. The reason that the assumption does not hold is in part due to the existence of subject diversity, which makes it difficult to distinguish wrongly labeled data from correct ones from a different subject whose data follows a different distribution (also known as *domain gaps*). Therefore, the second contribution of the work is *to unravel the interplay between subject domain gaps and LNL for HAR tasks.*

The insights from the empirical study motivate our third contribution, namely, to design VALERIAN, an invariant feature learning for In-the-wild domain adaptation method for wearable sensor-based HAR.

Its core component is a one-step domain invariant feature learner that tackles label noises and learns the shared feature representation among multiple subjects. VALERIAN uses self-supervised pretraining to learn good representations to take advantage of abundant unlabeled data (including those mislabeled). The pretrained parameters are used to initialize the shared feature encoder of a multi-task learning model, where each subject in the training data is considered a separate task. The network consists of shared feature representation layers and subject-dependent task-specific layers that are trained iteratively. To combat noisy labels, early-learning regularization (ELR) [16] is adopted by introducing a loss term reflecting the temporal ensemble of past predictions. At inference time, we assume a small number of clean labeled data is available from unseen subjects. The data is used to update the task specific layer to allow fast adaption of the trained model to the target user.

We have evaluated the performance of VALERIAN using two controlled datasets with different degrees and distributions of noisy labels injected and one in-the-wild dataset. We find that VALERIAN consistently outperform baseline approaches almost in all settings. Even with 40% label noise in training data, it achieves $\sim$ 83% test accuracy with only 10 seconds of clean labeled data per class from a new target subject in the controlled datasets. A similar evaluation on a true in-the-wild dataset with auto-corrected labels achieves an over 20% improvement in the F1-Score.

The rest of the chapter is organized as follows. Section 4.2 describes the motivation of this work. In Section 4.4, we introduce the VALERIAN method and the details of each component. In Section 4.5, we present the implementation details and performance evaluation of VALERIAN. Section 4.3 describes the related work and how they differ from ours. Finally, we conclude the chapter in Section 4.6 with a summary and a list of future work.

4.2 Motivation

To understand the characteristics of in-the-wild HAR datasets and to gain insights on why mainstream learning with noisy label (LNL) methods tend to fail on such tasks, we inspect two datasets and the behavior of a state-of-the-art (SOTA)

LNL algorithm in this section.

4.2.1 In-the-wild HAR datasets

In this work, an HAR dataset is considered to be in the wild (or collected in naturalistic settings) if the activities of subjects are not precisely scripted. As a result, experimenters do not know exactly what activities shall be performed at what time. The ExtraSensory dataset is one such example [53], where sensor data were collected from users' smartphone devices as they went about their daily activities. Activity labels were initially self-reported. Further curation was done by researchers who utilized information from other sensing modalities to automatically correct some data labels. For example, if GPS sensor readings indicate a subject is outside, the location label "indoor" submitted by the subject is erroneous and is corrected. A detailed description of the curation procedure used in ExtraSensory can be found in [3]. As another example, the Realworld dataset [52] contains data collected from fifteen subjects performing activities such as climbing stairs down and up, jumping, lying, standing, sitting, running/jogging and walking. Although in most cases, subjects were asked to perform a certain activity, during walking or jogging outside trials, the variations of terrains are not controlled by the experimenters and thus un-prescribed activities may occur.

Fig. 4.1 illustrates the percentage of clean and mislabeled data in both datasets. For RealWorld, we manually inspect the video recording of data trials for climbing up or climbing down activities, and note down the start and end time, and the type of activities. We find that there are periods that the subjects walk on a flat ground (7% of the time) or stand still (3% of the time) during the trials, which were mislabeled as climbing up or down in the dataset. For ExtraSensory, when comparing the self-reported and curated labels, we find that 34.5% are unchanged, 39.2% are corrected in the curation process and 26.3% are marked as invalid since the phones were not on subjects' body at the time. Moreover, upon closer inspection of curated data in ExtraSensory, we find the data labels are still noisy. For example, in Fig. 4.2, the left plot corresponds to accelerometer measurements labeled as standing while the right one is labeled as walking. However, one can easily observe the "signature" periodical pattern associated with walk cycles in the left plot but not in the right plot – an indication of mislabeling even after curation.

From Fig. 4.1, we conclude ExtraSensory is much noisier than RealWorld since the former is crowdsourced data. What also distinguishes the two datasets is the distribution of noisy labels. Specifically, for RealWorld, most mislabeling happens in the climbing up/down trials when the ground labels are "walk on a flat ground" or standing. In contrast, in ExtraSensory, mislabeling exists almost between any two activities. To characterize the distribution of noisy labels, a noise transition matrix T is often used, where element T_{ij} corresponds to the probability of mislabeling a data sample with ground truth label i to label j [99]. When mislabels occur equally likely for all classes other than the true class, the associate noise pattern is called *symmetric noise*. Otherwise, if there is a dominant off-diagonal element in each row in T, the associate noise pattern is called *asymmetric noise*.

Table 4.1 shows the noise transition matrix of data in three locomotion classes and one location class in ExtraSensory by comparing their curated labels (row headings) and the original ones (column headings). As ExtraSensory is a multi-label dataset with many classes, only top-5 mutually exclusive labels are included in the table.

Table 4.1: The noise transition matrix of ExtraSensory, based on its curated labels. For walking and standing, we only show their top-4 mislabeling sources due to space limits.

	walking	strolling	cleaning	cooking	eating
walking	75.28%	3.46%	3.46%	2.35%	1.67%
	running	exercise	go upstairs	go downstairs	
running	79.92%	19.66%	0.21%	0.21%	
	standing	cooking	cleaning	shower	dressing
standing	56.79%	8.47%	7.51%	5.35%	5.34%
	at home	at school	at work	at party	at gym
at home	96.71%	1.49%	1.27%	0.27%	0.26%

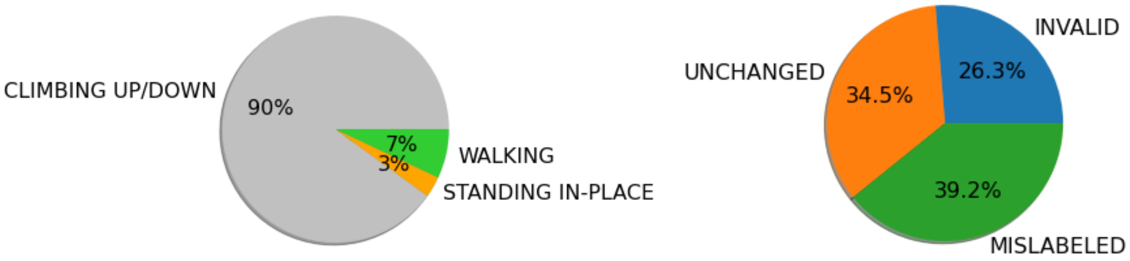


Figure 4.1: The statistics of two in-the-wild IMU-based HAR datasets. Left: Realworld dataset, Right: ExtraSensory dataset. A noticeable portion of the data labels in both datasets are noisy.

We observe that with the exception of "running", noise transition probabilities of all classes are best modeled as symmetric noise.

4.2.2 Failures of mainstream LNL methods

Learning with noisy labels has long attracted attention in the machine learning community with published work on this topic dated as early as 1988 [100]. Recently, many deep learning-based methods have been proposed for LNL that primarily target computer vision tasks. Next, we discuss one SOTA LNL method and investigate its

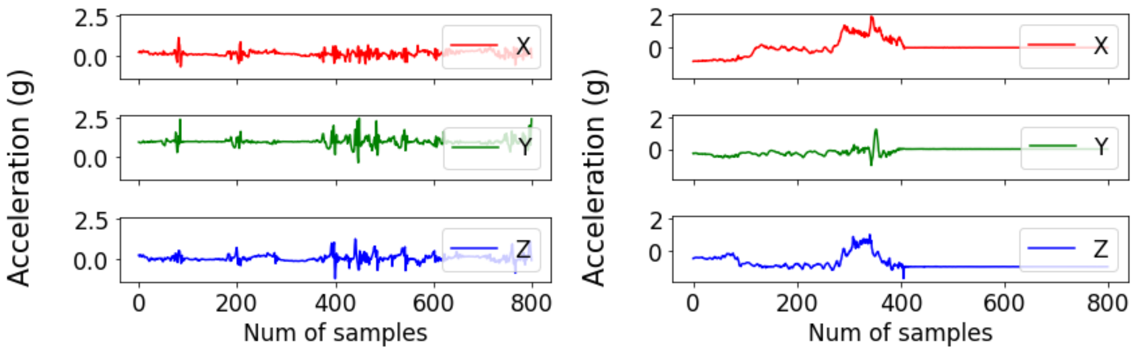


Figure 4.2: Accelerometer measurements from ExtraSensory dataset with curated labels. Left: standing (subject id: FDAA70A1-42A3-4E3F-9AE3-3FDA4 12E03BF, row id: 4339), Right: walking (subject id: 2C32C23E-E30C-498A-8DD2-0EFB9150A02E, row id: 5454).

performance when applied directly to HAR with noisy labels.

DivideMix [15] is a representative co-training based method. The basic idea of DivideMix is to first train a model with all training data for a few epochs (called *warm-up phase*). A Gaussian mixture with two modes is fitted to divide data samples based on their normalized losses into two partitions – those with lower losses (higher confidence) are considered clean labeled data, and those with high losses are treated as unlabeled data. Semi-supervised learning is then applied to the mixed data. Subsequently, co-refinement of labeled data and co-guessing of the labels of unlabeled data is performed by two neural networks working together iteratively, to reduce biases.

To study the behavior of DivideMix for HAR, we utilize the USCHAD dataset [47]. This dataset contains accelerometer and gyroscope measurements collected from fourteen participants performing ten types of locomotions in a controlled environment (i.e., walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing and sleeping). USCHAD is chosen here because the dataset is carefully curated with ground truth labels. To simulate label noise, we consider both symmetric and asymmetric noises as illustrated in Fig. 4.3, where τ is a configurable parameter controlling noise levels.

$$\begin{bmatrix} 1-\tau & \frac{\tau}{n-1} & \dots & \frac{\tau}{n-1} & \frac{\tau}{n-1} \\ \frac{\tau}{n-1} & 1-\tau & \dots & \frac{\tau}{n-1} & \frac{\tau}{n-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\tau}{n-1} & \frac{\tau}{n-1} & \frac{\tau}{n-1} & \dots & 1-\tau \end{bmatrix}$$

(a) Symmetric noise

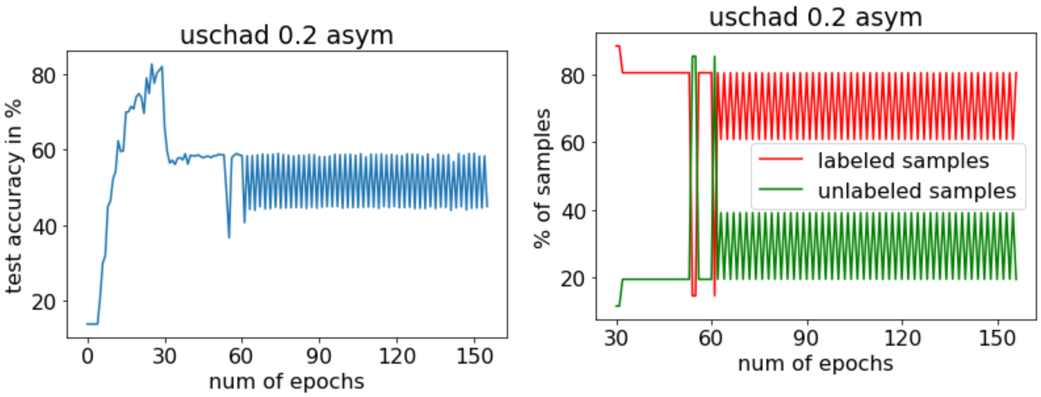
$$\begin{bmatrix} 1-\tau & \tau & \dots & 0 & 0 \\ 0 & 1-\tau & \dots & \tau & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \tau & \dots & 1-\tau \end{bmatrix}$$

(b) Asymmetric noise

Figure 4.3: Noise transition matrix T for symmetric and asymmetric noise distributions

We adopt the DeepConvLSTM model architecture proposed in [97] for HAR tasks and train on the USCHAD dataset. The model contains 4 convolutional neural network (CNN) layers and 2 long short-term memory (LSTM) layers totalling ~296k trainable parameters.

Fig. 4.4 shows the behavior of DivideMix over training epochs using the DeepConvLSTM model in presence of 0.2 asymmetric labelling noise. In the experiments, 13 of 14 subjects are included in the training data and the remaining subject is used in testing. The warm-up phase ends at 30 epochs. As shown in Fig. 4.4a, test accuracy increases quickly during the warm-up phase indicating that the model can learn despite label noises. However, after the warm-up phase, the test accuracy drops drastically and fluctuates between 45% and 60% after 60 epochs. A closer look at the division between labeled and unlabeled data in the training set by DivideMix in Fig. 4.4b reveals that despite only 20% of the data samples are labeled incorrectly, DivideMix gradually converges to split the data approximately 81-19 or 61-39. As a result, some clean labeled data is classified as unlabeled and fail to contribute as much to the training process.



(a) Test accuracy

(b) Division of clean and noisy labeled data

Figure 4.4: The performance of DivideMix on USCHAD in leave-one-subject-out experiments.

To shed the light on why DivideMix fails in these experiments, further analysis is in order. First, we inspect the effect of memorization. Deep neural network models are known to have the propensity for fitting training data including outliers or mislabeled data. However, it has been empirically demonstrated that such a memorization phenomenon tends to happen at a late stage of training [101, 16]. In early training epochs, the model prioritizes learning simple patterns. Based on such observations, many LNL approaches take advantage of *early stopping* to learn a base model from all data. Since noisy labels are in the minority and tend to be "irregular", smaller losses and higher prediction confidence are associated with clean labels in the early training epochs. To test if this hypothesis is true for HAR tasks, we show in Fig. 4.5 the breakdown of training samples among five categories. Specifically, a data sample that is correctly labeled can be either correctly or wrongly predicted by the trained model up to the associated epoch. For a data sample that is wrongly labeled, three situations may arise: i) its prediction is the same as the ground truth label (*correct*),

ii) its prediction is the same as the wrong label (*memorized*) or iii) otherwise, i.e., its prediction is neither the ground truth label nor the wrong label. From Fig. 4.5, even after a few epochs, memorization is non-negligible especially in the case of asymmetric noise. When a noisy label is memorized, the model has high confidence in its *wrong* prediction.

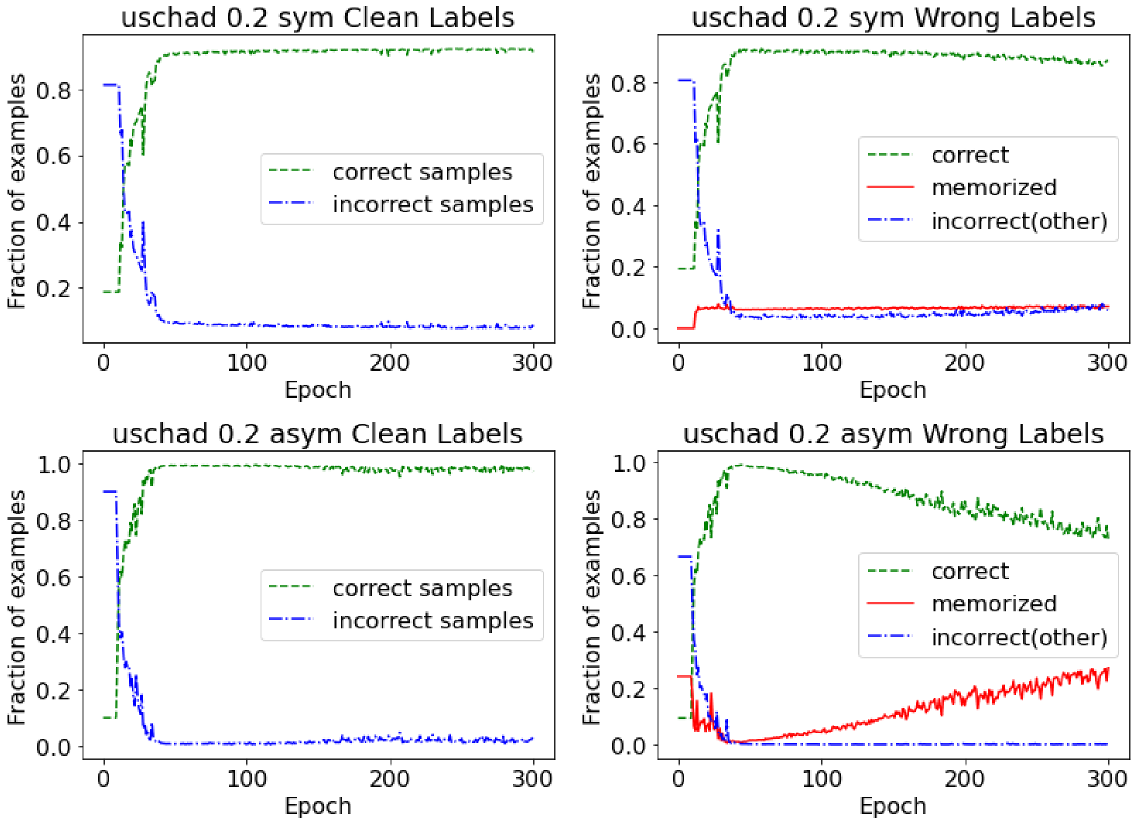
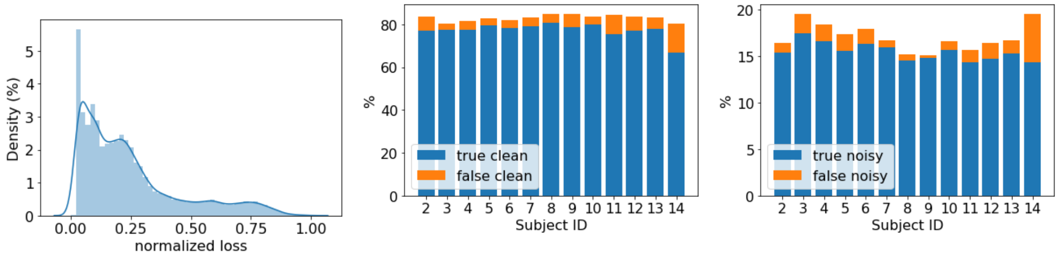


Figure 4.5: Results of the DivideMix model on USCHAD with 0.2 symmetric noise and asymmetric noise. Left: the fraction of clean labeled samples that are predicted correctly (green) and incorrectly (blue). Right: the fraction of samples with wrong labels that are predicted correctly (green), memorized (red), and incorrectly as neither the true nor the labeled class (blue).

We believe the root cause of early memorization and the consequent failure of

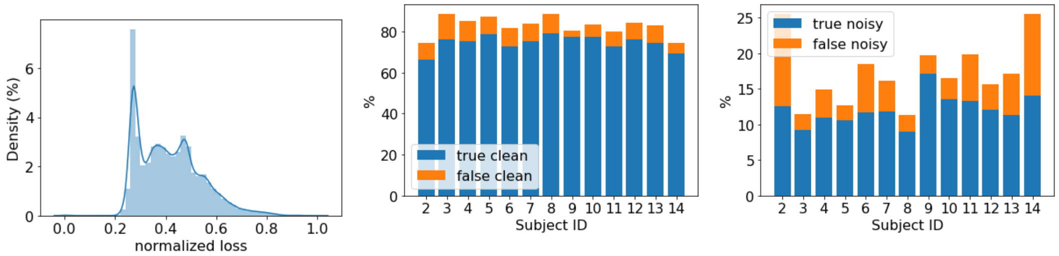


(a) Distribution of normalized losses

(b) Partition of the data predicted as clean by DivideMix

(c) Partition of the data predicted as noisy by DivideMix

Figure 4.6: Effects of subject diversity on early learning. Plots are generated on a model trained on Subject 2 – 14 in USCHAD with 0.2 symmetric noise and after 30 epochs of warm-up training.



(a) Distribution of normalized losses

(b) Partition of the data predicted as clean by DivideMix

(c) Partition of the data predicted as noisy by DivideMix

Figure 4.7: Effects of subject diversity on early learning. Plots are generated on a model trained on Subject 2 – 14 in USCHAD with 0.2 asymmetric noise and after 30 epochs of warm-up training.

DivideMix in HAR tasks is due to the large variability across subjects when performing the same activity. Subject diversity is a well-recognized problem in HAR [102]. However, the problem is exacerbated when noisy labels are present. In Fig. 4.6 and 4.7, we show the normalized cross-entropy losses for Subject 2 – 14 in the training data and the division of clean and noisy labels for each subject in DivideMix after a 30-epoch warm-up period. Clearly, the normalized losses (Fig. 4.6(a) and 4.7(a)) no longer follow a two-component GMM. Instead, they are better modelled by a mixture of three or more components. Moreover, the first and second components in Fig. 4.6(a) and all components in Fig. 4.7(a) have close mean values. Inspecting the division of labelled and unlabeled data for each subject by DivideMix, we find that some presumably clean data is in fact noisy (false clean in Fig. 4.6(b)) while a portion of presumably noisy data is in fact clean for each subject (false noisy in Fig. 4.6(c)). Some subject (e.g., Subject 14) appears to be penalized with a higher percentage of clean data being mislabeled as unlabeled by DivideMix. The wrong division is even more prominent with asymmetric noise in Fig. 4.7(b) and (c), where more than 10% of Subject 14's clean data is wrongly classified as wrong labels (due to high normalized losses).

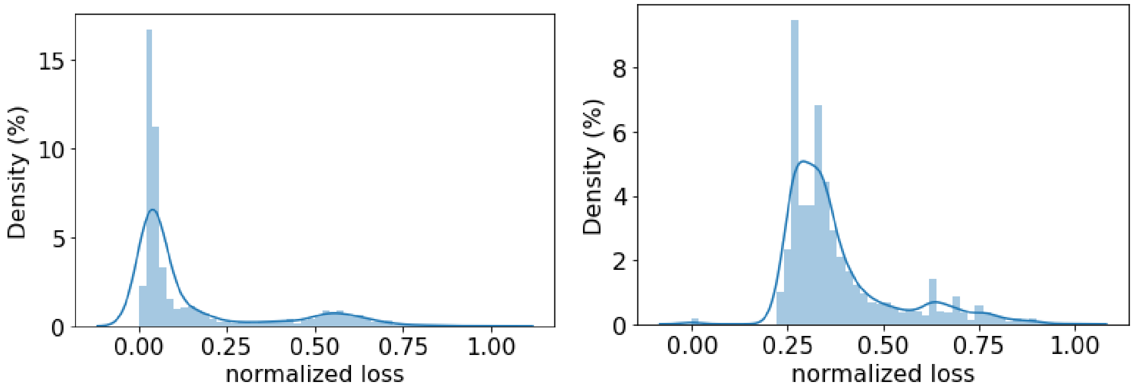


Figure 4.8: Distribution of normalized losses on USCHAD with one subject (subject id: 2). left: 0.2 symmetric (false clean: 5.25%; false noisy: 0%); right: 0.2 asymmetric (false clean: 2.47%; false noisy: 5.58%).

Fig. 4.8 shows the distribution of normalized losses at the end of the warm-up phase when training DivideMix on data from one subject with 0.2 symmetric and asymmetric labeling noise. To avoid overfitting, we reduce the network size of DeepConvLSTM and retain only two CNN layers and one LSTM layer with a total of 56k trainable parameters. The distributions in Fig. 4.8can indeed be modeled as 2-component GMM following the basic assumption of DivideMix and thus can be correctly handled by the method (results omitted for brevity). Comparing the results with Fig.4.6(a) and 4.7(a), it is reasonable to conclude that the discrepancy is due to domain gaps in multi-subject data in the former cases.

Though our analysis focuses on DivideMix, other LNL methods such as ELR [16] make the same assumptions that high-confidence labels in early training stages are more trustworthy. Unfortunately, as evident from the empirical analysis in this section, such assumptions no longer holds in presence of diverse subject data in HAR tasks.

4.3 Related Work

Learning with noisy labels LNL has been investigated in computer vision and audio signal processing for over a decade [100, 14]. Existing methods can be categorized into three groups. First, contrastive learning-based LNL methods [103, 104] add regularization terms to the loss function to obtain a well-clustered feature structure. Second, curriculum learning [105, 106] or teacher-student networks such as MentorNet [107] trains a neural network to guide a student network by assigning weights to samples. Since the pioneer co-teaching work [99], the use of two networks together gains popularity in LNL and has been adopted in several recent papers including DivideMix [15], ELR+ [16], co-regularization [108]). Instead of training a model that works on the noisy labeled samples, another line of work aims to select clean labeled samples out of noisy ones [109, 110]. Despite all the advancements in LNL, none of the afore-mentioned work considers domain gaps between source and target domains (also known as domain shifts).

Weakly-supervised learning in sensor-based HAR There are some works in mobile computing that deal with weakly-supervised learning problems related to sensor-based HAR [111, 112, 113]. Wang *et al.* in [112, 113] define weakly-supervised learning as detecting the start and end of an activity of interest in a given time-series data sequence, similar to the sound event detection problem[114, 115, 116]. Unlikely our problem, the goal is to crop the data of interest from a noisy sequence for training so that a machine learning model can gain a better discriminative power. For instance, consider a collected *climbing up* IMU data trial with two activities: climbing upstairs and walking on the flat ground. Wang *et al.* treat walking as a background activity and try to detect the onset and offset timestamps of climbing upstairs events. In contrast, in this work, we treat the data within such a trial as a mixture of *climbing up* and noisy labeled *walking* activities. Apart from the different ways of treating label noises, existing works still require further steps to handle subject diversity within the training process to generalize well to new unseen subjects.

Joint LNL and domain adaptation A few works consider LNL together with domain shifts. Shu *et al.* in [117] considered noise either in data or label of a single source domain and perform weakly-supervised model training to adapt to a target domain. In [118, 119] researchers propose one-step solutions to LNL and unsupervised domain adaptation. However, these methods have been applied to image classification tasks, where there is only a single source domain. Thus, the authors only consider the domain shift between one source domain and one target domain. In contrast, in our work, we need to take into account domain shifts amongst multiple source domains, namely, different human subjects. As discussed in Section 4.2, subject diversity in training data prevents conventional LNL methods from working effectively since early learning can inadvertently memorize noisy data.

4.4 Invariant Feature Learning for IMU-based HAR in the Wild

Let the input and label spaces be $\mathcal{X}$ and $\mathcal{Y}$, respectively. Due to high subject diversity in HAR tasks, each subject in the training set is treated as a separate source domain in the joint space $\mathcal{X} \times \mathcal{Y}$. In the rest of the chapter, we use "domain" and "subject" interchangeably. Let $\mathcal{D}_k = \{(x_n^k, \tilde{y}_n^k)\}_{n=1}^{N_k}$, where N_k is the number of data samples from subject k and $\tilde{y}$ denotes noisy labels. The source domains are denoted by $\mathcal{D}_s = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$, where K is the number of subjects. We further assume that a small collection of clean labeled samples can be obtained for an unseen subject t denoted by $\mathcal{D}_t = \{(x_n, y_n)\}_{n=1}^{M}$. The goal of *HAR from data in-the-wild* is to learn a model from $\mathcal{D}_s$ that can be easily adapted to a new target domain given $\mathcal{D}_t$.

A naive solution to this problem is to first apply an existing LNL method to data from individual subjects separately to acquire more accurate pseudo labels based on the predictions of the respective subject-dependent model (Step 1). The pseudo-labelled data from multiple subjects are then aggregated to train a subject-independent model (Step 2), which can later be adapted to unseen subjects with clean data. There are two problems with such a two-step approach. First, a subject-dependent model of suitable capacity has to be chosen in Step 1 according to the size of data from each subject. Small-capacity models generally under-fit data while large-capacity models can easily overfit and memorize noisy labels[1]. More importantly,

[1] This is precisely the reason we had to manually prune the DeepConvLSTM model architecture to make DivideMix works for single subject's data in Fig. 4.8.

since LNL and learning domain invariant feature (to train a subject-independent model) is done in two steps, mislabeled data after LNL cannot be corrected in domain adaptation; and LNL cannot benefit from any shared pattern across different subjects.

Motivated by the observations from Section 4.2, we propose VALERIAN, a one-step method that handles noisy labels and distribution gaps across multiple source domains simultaneously. Our solution is based on two key insights: i) unsupervised learning that aims to learn representations invariant to instance-level variations is not affected by noisy labels; and ii) within each source domain, clean data tends to exhibit simpler patterns (than wrongly labelled data), which can be learned in early training epochs. Moreover, we assume that in absence of noisy labels, there exist domain-invariant features across subjects in HAR tasks. This assumption has been empirically verified in prior work [120].

As a one-step solution to tackle noisy labels and domain shift issues, VALERIAN differs from Butterfly [118] in a way that we also consider the variance among source domains, which is crucial for HAR tasks. Empirical results on public datasets show that our approach is superior to Butterfly. (see Section 4.5.4 for details)

4.4.1 Solution overview

VALERIAN takes advantage of known techniques in machine learning but combines them in innovative ways. It has three key building blocks: i) self-supervised pre-training, ii) invariant feature learning from noisy labelled data, and iii) fast adaption to unseen subjects.

Self-supervised pre-training takes unlabeled data and performs data augmentation to pre-train feature extractors that capture structures of underlying distributions. Invariant feature learning in VALERIAN has two objectives: 1) to learn shared feature representations across domains and 2) to combat the memorization effect introduced by noisy labels. To do so, we adopt a multi-task learning model for domain invariant feature learning first proposed in [120] that consists of shared features across multiple source domains and a task-specific output layer.

To counter the effect of noisy labels, we introduce a regularization term similar to ELR in the loss function during training. Finally, for a new subject with a small amount of clean data, fast adaption is performed on the task-specific layers of the multi-task model only.

Algorithm 2 summarizes the training procedure of VALERIAN. Next, we will provide the details of each building block.

4.4.2 Self-supervised pre-training

In [121], the authors find that a ResNet pre-trained on ImageNet datasets appears to work consistently better than random initialized ones as a feature extraction network for LNL image classification tasks. Inspired by this, here, we pre-train a feature extractor network by removing the labels in HAR datasets. In such cases, it is natural to consider feature learners that require no label information, such as contrastive learning [122] or self-supervised learning. Self-supervised learning is a machine learning method that learns semantic features from unlabeled data with customized tasks [123]. As there is no ground truth label, to take advantage of this technique, data augmentation and auxiliary tasks need to be introduced. In [79],

Algorithm 2 Invariant feature learning for in-the-wild domain adaptation

```
Require: Source domains $\mathcal{D}_s = \{D_k\}_{k=1}^{K}$, learning rate $\gamma$, hyperparameters $\alpha, \beta, \lambda, \mu$
Ensure: VALERIAN model with parameter $\theta$ and $\phi$
Initialize $\theta$ with self-supervised pretrain
Random initialize $\phi = \{\phi^1, \phi^2, ..., \phi^K\}$
Initialize ensemble predictions $t \leftarrow 0_{[n \times C]}$
repeat
    Sample tasks $T = \{T_1, T_2, ..., T_K\}$ over $\mathcal{D}_s$
    //Update $\phi^k$ with fixed $\theta$
    for $k$ is 1 to $K$ do
        Freeze parameters of $\phi$ except $\phi^k$
        for each minibatch B in $T_k$ do
            for $(x_i, \tilde{y}_i)$ in B do
                $p_i \leftarrow S_{\phi^k}(L_\theta(x_i))$
                $t_i \leftarrow \beta t_i + (1 - \beta) p_i$
            end for
        end for
        $\mathcal{L}oss \leftarrow \mathcal{L}_{CE}(T_k, \theta; \phi^k) + \mu|\phi^k|_1 + \frac{\lambda}{|B|} \sum \log(1 - \langle p_i, t_i \rangle)$
        $\phi^k \leftarrow \phi^k - \gamma \nabla_{\phi^k} \mathcal{L}oss(T_k, \theta; \phi^k)$
    end for
    //Update $\theta$ with fixed $\phi$
    for each minibatch B in $T$ do
        $B' = Mixup(B, \alpha)$
        for $(x_i, \tilde{y}_i)$ in B' do
            $p_i \leftarrow S_\phi(L_\theta(x_i))$
            $t_i \leftarrow \beta t_i + (1 - \beta) p_i$
        end for
    end for
    $\mathcal{L}oss \leftarrow \mathcal{L}_{CE}(T, \phi; \theta) + \mu|\phi|_1 + \frac{\lambda}{|B|} \sum \log(1 - \langle p_i, t_i \rangle)$
    $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}oss(T, \phi; \theta)$
until convergence
```

Saeed *et al.* introduce several transformations to input data and train a multi-task model to classify the type of transformation applied. We adopt the same idea and apply the following transformations:

1. *Noised*: it adds random Gaussian noise to the original data samples.

2. *Scaled*: this transformation changes the magnitude of data samples within a sliding window by multiplying with a random scalar.

3. *Rotated*: this transformation mimics different sensor orientations by multiplying the original data with a rotation matrix of randomly generated axis-angle.

4. *Negated*: this transformation negates samples within a time window, resulting in a vertical or a horizontal flip of the original input signal.

5. *Reversed*: it reverses the data along the time dimension, resulting in a complete mirror image of the original input.

6. *Permuted*: sensor signals are randomly sliced and swapped within a data window.

7. *Time-Warped*: it mimics the change of motion frequency by locally stretching or warping a time series through a smooth distortion of time intervals.

8. Channel-Shuffled: this transformation randomly shuffles the sensor signals in axial dimensions.

One or several of these transformations (called *pretext tasks*) are applied to each data window to each sensor separately (accelerometer and gyroscope). Each head of the multitask learning model corresponds to a binary classifier. By learning whether a certain type of transformation has been applied to the original data samples, the feature extractor portion of the network captures high-level semantic information that is invariant to these transformations and thus beneficial to downstream tasks.

4.4.3 Domain invariant feature learning

Self-supervised learning alone is insufficient to handle domain gaps among subjects. Moreover, data labels are necessary to fine tune model parameters for downstream tasks. To generalize well to unseen subjects, we utilize the invariant feature learning framework (IFLF) from Chapter 3. It consists of three components:

Alternating training An IFLF model is a multi-task model trained with tasks sampled from all source domains. In each training epoch, we first freeze the parameters of the feature extractor network, and update the parameters of each task-specific layer with its respective data; then, we freeze the parameters of all task-specific layers and update the invariant feature extractor using all data from the previous step.

Feature extractor By the merit of multi-task learning, L_θ generalizes well across domains through the shared representations among related tasks [82]. For HAR tasks, we use DeepConvLSTM [97] as the backbone network. It includes four CNN layers and two LSTM layers. The objective function ℓ_L works on multiple source domains to learn a domain invariant feature representation that clusters the features by their labels. It is defined as follows:

$$\ell_L = \sum_{k=1}^{K} \mathcal{L}_{CE}(T_k, \phi^k; \theta), \tag{4.4.1}$$

where $\mathcal{L}_{CE}$ is the categorical cross-entropy loss function calculated on each T_k with given θ and ϕ, defined as $\mathcal{L}_{CE} = -\sum_{i=1}^{C} \tilde{y}_i log(p_i)$ on data from each task k. We call such a multi-task model *basic multi-task learning model* (BMTL). To further boost the quality of extracted features, we use self-supervised pre-train as described in Section 4.4.2 to initialize the model parameter θ.

Task-specific networks Generally, if the shared feature generalizes well across all source domains, it also works well on the target domain. L_θ needs to have sufficient capacity to explore the entire latent space $\mathcal{Z}$ and extract domain invariant features. In contrast, a task-specific network S_ϕ^k should be as simple as possible with fewer learnable parameters to allow fast adaptation with target domain data. In the implementation, a lightweight task-specific layer S_{ϕ^k} includes a fully connected layer with a softmax activation function. The task-specific objective function is defined as the sum of a categorical cross-entropy loss and an ℓ_1-norm regularization term as follows,

$$\ell_{S^k} = \mathcal{L}_{CE}(T_k, \theta; \phi^k) + \mu|\phi^k|_1, k = 1, 2, \ldots, K, \tag{4.4.2}$$

where μ is a hyper-parameter to control the sparsity of S_ϕ^k. The regularization term imposes sparsity on the task-specific layers and helps mitigate overfitting.

4.4.4 Learning with noisy labels

With the multitask learning model introduced previously, we can get the best of both worlds: shared network parameters for feature extraction for all subjects and subject-dependent output layers. As a result, the underlying assumption of dominant LNL methods is that in early training epochs, each subject-dependent model tends to incur low losses (higher confidence) on clean data and large losses on mislabeled data are likely to hold. To handle noisy labels, in principle, we can incorporate any existing LNL method in the invariant feature learning framework. However, we find that DivideMix has high training costs due to its use of two networks in co-teaching and co-refinement. When combined with invariant feature learning, its complexity grows linearly with the number of source domains. Therefore, in VALERIAN, we employ ELR to counter memorization effects by forcing model predictions to be close to their temporal ensemble. An ELR loss is defined as :

$$\mathcal{L}_{elr} = \frac{1}{|B|} \sum_{i=1}^{|B|} \log\left(1 - \langle p_i, t_i \rangle\right), \tag{4.4.3}$$

where p_i is the model output of input sample x_i, and $t_i = \beta t_i + (1-\beta) p_i$ is the temporal ensemble controlled by hsyper-parameter β. (4.4.3) maximizes the inner product of p_i and t_i, and the logarithm in $\mathcal{L}_{elr}$ inverts the exponential function implicit in the softmax function in p_i.

MixUP [124] is a simple yet effective data augmentation technique in boosting model generalization capabilities [125]. In HAR tasks, we can mix up data samples from the same activity class but different subjects. To apply Mixup data augmentation, each data sample of a mini-batch is interpolated with another sample randomly chosen from a different source domain but belongs to the same class. Specifically, for a pair of samples $(x_1, \tilde{y}) \in \mathcal{D}_i$ and $(x_2, \tilde{y}) \in \mathcal{D}_j$, the mixed data sample $(x', \tilde{y})$ is computed by:

$$a \sim Beta(\alpha, \alpha), \tag{4.4.4}$$

$$a' = max(a, 1-a), \tag{4.4.5}$$

$$x' = a' x_1 + (1-a') x_2 \tag{4.4.6}$$

where a is the MixUp factor sampled from a *Beta* distribution controlled by hyper-parameter α. Finally, the total losses in (4.4.1) and (4.4.2) are updated as:

$$\mathcal{L}oss_L = \ell_L + \mu|\phi|_1 + \lambda\mathcal{L}_{elr}, \tag{4.4.7}$$

$$\mathcal{L}oss_{S^k} = \ell_{S^k} + \lambda\mathcal{L}_{elr}, k = 1, 2, \ldots, K, \tag{4.4.8}$$

where λ is a hyper-parameter to control the importance of ELR. It is worth noting that the loss is calculated differently in the alternating training procedure as L_θ includes all source domains while ϕ^k only concerns the data of the kth subject. MixUp augmentation is only used in updating the feature extraction layers (L_θ).

4.4.5 Fast adaptation to new subjects

Since the network parameters in task-specific layers are already sparse, for a new subject, one can either initiate a new task-specific layer from scratch or randomly select a S_{ϕ^k} to update its trained parameters. A small amount of clean data is taken from $\mathcal{D}_t$ to train the task-specific layer.

4.5 Experimental Evaluation

In this section, we evaluate the performance of VALERIAN in IMU-based HAR tasks under different scenarios.

4.5.1 Datasets

We consider three public datasets to cover a wide variety of device types, data collection protocols, and activity classes to be recognized. Because the evaluation of machine learning models requires the availability of clean ground truth labels, the first two datasets, USCHAD and WISDM [48] were collected under controlled laboratory environments. To simulate labelling errors, symmetric or asymmetric noise is injected into the labels with different noise transition matrices. WISDM contains a large number of subjects. Raw accelerometer and gyroscope data have been collected from a smartphone in each participant's pant pocket at a rate of 20Hz. There are a total of 51 test subjects performing seven locomotion activities (i.e., walking, jogging, stairs, sitting, standing, kicking a soccer ball, playing tennis) for three minutes per trial to get equal class distribution.

The third dataset, ExtraSensory, allows us to gauge VALERIAN's ability to handle real in-the-wild data. In ExtraSensory, crowdsourced mobile phone data are collected from 60 subjects during daily living activities. In the evaluation, we only consider six locomotion-related activities, namely, walking, running, cycling, sitting, standing and lying down. In the absence of ground truth labels, we take instead the curated labels as ground truth. However, as discussed in Section 4.2, the curated data remains to be noisy. Moreover, ExtraSensory also suffers from severe class imbalancing and missing class issues (only nine out of 60 subjects have data from all six classes in the dataset).

4.5.2 Baseline methods

We have implemented five baseline models for comparison purposes.

- *Single-task learning model (STL)*: STL is trained from scratch solely on the clean data from a target domain (a new subject). As the amount of clean data increases, it is expected that STL's performance improves since there is no label noise.

- *Basic multi-task learning model (BMTL)*: Similar to VALERIAN, BMTL is a meta-learning method trained with noisy source domains and adapted to a target domain with a small amount of clean labels. However, unlike VALERIAN, BMTL does not perform self-supervised pre-training, and treats all training data as clean.

- *Subject-independent model with cross-entropy losses (SI)*: This method pools all but test subjects' data to train a single subject-independent model and treats all training data as clean.

- *Subject-independent model with ELR (SI-ELR)*: It is a subject-independent model trained by pooling all but test subjects' data together. Unlike SI, it utilizes ELR to combat noisy labels.

- *Butterfly [118]*: It is a joint LNL and domain adaptation method, which treats all but test subjects' data as a single source domain and takes all unlabeled data samples from a target domain to train a model. Butterfly maintains four deep networks simultaneously, two of which perform adaptations (i.e., noisy-to-clean, labeled-to-unlabeled, and source-to-target domains) and the remaining two perform classification in the target domain.

BMTL and VALERIAN are meta-learning methods, while STL is their natural contrast in that they utilize some clean data from the target domain. From the discussion, it can be seen that SI and SI-ELR do not require any target domain data. However, for a fair comparison, we perform transfer learning to update the parameters of both models using a few samples per class from a target domain. Butterfly on the other hand includes unlabeled target domain data during training and thus no transfer learning using clean target domain data is done at inference time.

4.5.3 Implementation and evaluation procedure

Data preparation A standard IMU data pre-processing procedure is implemented for the experiments, including interpolation, low-pass filtering, normalization, and data segmentation. A Butterworth low-pass filter [91] with a cut-off frequency of 10Hz is employed to remove high-frequency noise from interpolated data. After low-pass filtering, we normalize the data and then segment it into sliding windows with a fixed length of 2 seconds with 80% overlapping between adjacent windows.

Implementation The implementation of the feature extractor follows DeepConvLSTM in all models. It includes four layers of 1D CNN and two LSTM layers with 128 hidden units and a drop-out rate of 0.25 to prevent over-fitting [92]. The CNN layers have 64 channels with kernel size 5 and stride 1.

For STL, the models are trained with a RMSProp optimizer [94] at a learning rate of 10^{-3} and a decay factor of $p = 0.9$. The maximum iteration number is set to be 500. The SI models are trained with 200 epochs only, as the memorization effect will gradually degrade the model performance in latter training epochs. Butterfly and ELR are trained using hyper-parameters as specified in the original papers. VALERIAN utilizes DeepConvLSTM in L_θ while the number of S_{ϕ^k} branches is determined by the number of subjects in the training data. Each S_{ϕ^k} may have a different output shape depending on the number of classes in the dataset for the respective subject. VALERIAN is trained with an Adam [95] optimizer at a learning rate of $10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$, with hyper-parameters $\mu = 0.4$, $\alpha = 0.2$, $\beta = 0.7$, and $\lambda = 3$. The batch size is set to 64 and the number of training epochs is 300 without early stopping. The hyper-parameters and the optimizer used in each model are consistent across all datasets.

Evaluation process Since the noise levels in in-the-wild data are unknown, we evaluate the robustness of the proposed approach by introducing different levels of noise to clean datasets. Two noise patterns with four levels each are considered, namely, symmetric noise with $\tau = \{0.1, 0.2, 0.4, 0.6\}$ and asymmetric noise with $\tau = \{0.1, 0.2, 0.3, 0.4\}$. The noise transition matrices are then defined according to Fig. 4.3. From Section 4.2, we have seen that LNL with asymmetric noises is generally harder than that with symmetric noises. For example, when $\tau = 0.4$ and the number of classes $C = 10$, under asymmetric noise, roughly 60% of data in each class is correctly labeled while the remaining 40% is labeled to another class. As a result, the percentage difference between correctly and wrongly labeled data is only 20%. In contrast, in symmetric noise cases, the percentage gap is $60 - \frac{40}{9} \approx 55.6\%$ (since the percentage of the wrongly labeled class is $\frac{40}{9}$). Therefore, for asymmetric noise, the maximum τ is set to 0.4 but in the case of symmetric noise, the maximum τ is set to to 0.6. In the experiments, to better simulate real-world noise patterns, the noise transition matrices of asymmetric noise are defined by setting the probability of the most similar class of each activity to τ, as shown in Fig. 4.9[2].

[2] The most similar class is determined by the confusion matrix of a model trained on clean data.

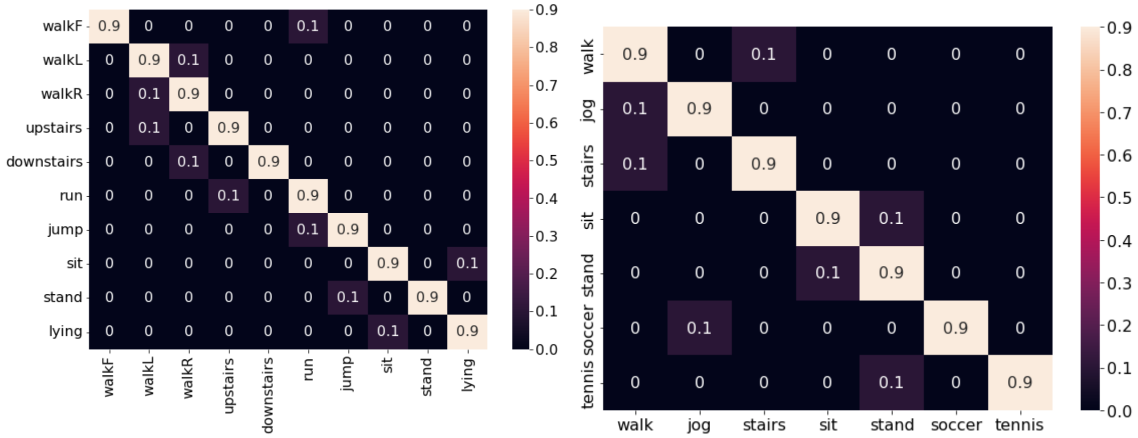


Figure 4.9: Noise transition matrix T with asymmetric noise for USCHAD (Left) and WISDM (Right), $\tau = 0.1$.

Leave-one-subject-out evaluation is conducted on all datasets. In the experiments, we randomly select one subject as the target domain at a time, until all subjects are chosen. In Butterfly, we evaluate it in a way as described in its original paper [118]. For SI-ELR and SI, we update the trained models with five clean data windows per class from the target domain in transfer learning. For VALERIAN and BMTL, we randomly sample a fixed test set from the target domain $\mathcal{D}_t$, and then select n data windows (of length 2s) per class from the remaining data in $\mathcal{D}_t$ to update an arbitrary S_{ϕ^k} layer. This process is also called n-shot learning. Here $n = \{1, 2, 5, 10, 20\}$. An STL model is also trained with the n data windows and test accuracy is recorded for each n. Experiments are repeated five times for each parameter setting, and the mean test accuracy and its standard deviation are reported.

4.5.4 Results

Controlled Datasets

First, we present the evaluation results on controlled datasets with meta-learning methods (e.g., STL, BMTL and VALERIAN). Fig. 4.10 and 4.11 show the results on USCHAD with asymmetric and symmetric noise of different levels respectively. The results on WISDM are given in Fig. 4.12 and 4.13.

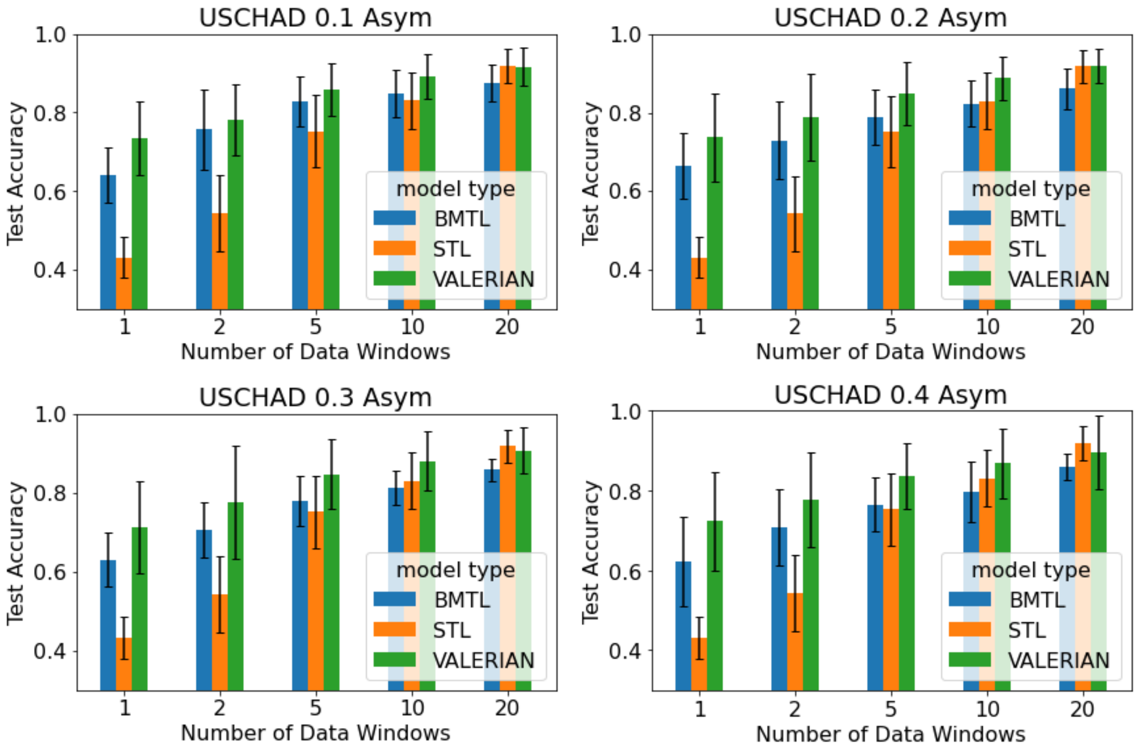


Figure 4.10: Evaluation on USCHAD with different levels of asymmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

Overall performance From these figures, we observe that VALERIAN works well and its performance is quite stable across different noise levels and types of noise in

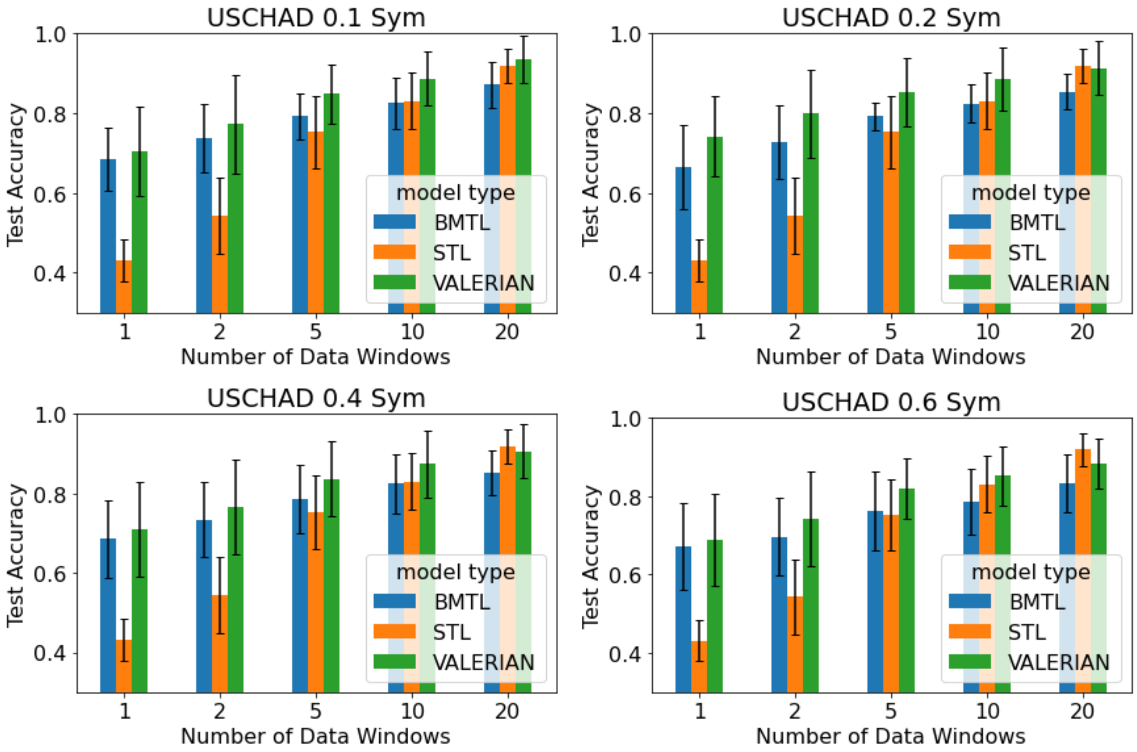


Figure 4.11: Evaluation on USCHAD with different level of symmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

both datasets. It almost always outperforms BMTL, which confirms the necessity to handle noise in meta-training. As STL is trained entirely on clean data from $\mathcal{D}_t$, its performance is not impacted by noise patterns and levels. As more clean data become available, the performance of STL serves as an upper bound of LNL models. From the figures, we see that with 20 shots, VALERIAN has comparable or slightly worse performance than STL. However, with a smaller number of target domain data, VALERIAN learns more efficiently. For example, with five shots, the average accuracy of VALERIAN for UHSCHAD and WISDM across all noise levels and patterns are 84.35 and 83.87, respectively, which are superior than BMTL (78.71 and 78.46) and

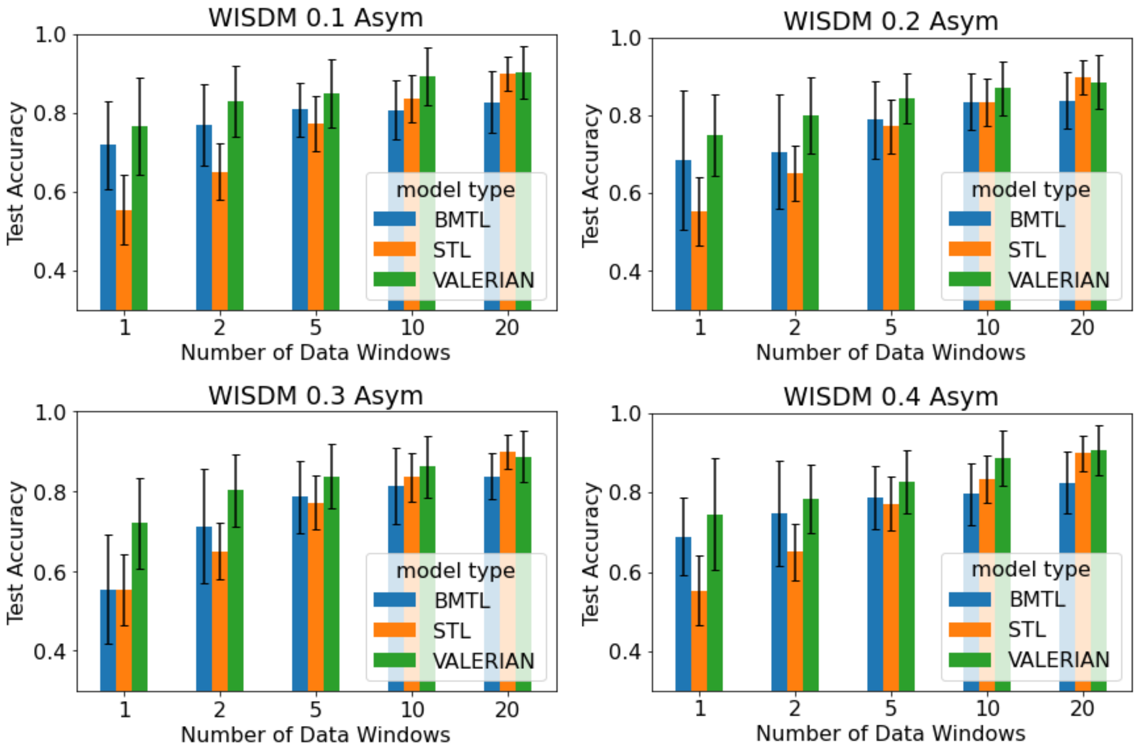


Figure 4.12: Evaluation on WISDM with different level of asymmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

STL (75.26 and 77.20).

Effect of noise levels and noise pattern From Fig. 4.10 – 4.13, as expected, as the noise level increases, the accuracy of VALERIAN degrades slightly. However, even with 60% symmetric noise, it can achieve an average accuracy of 81.99% for USCHAD for 5-shot learning, amounting to less than 3% reduction compared to the case with 10% symmetric noise. Similar observations can be made for asymmetric noise and WISDM.

When comparing the accuracy of VALERIAN in symmetric and asymmetric noise cases, interestingly, not much difference can be observed even for 1-shot case. This is

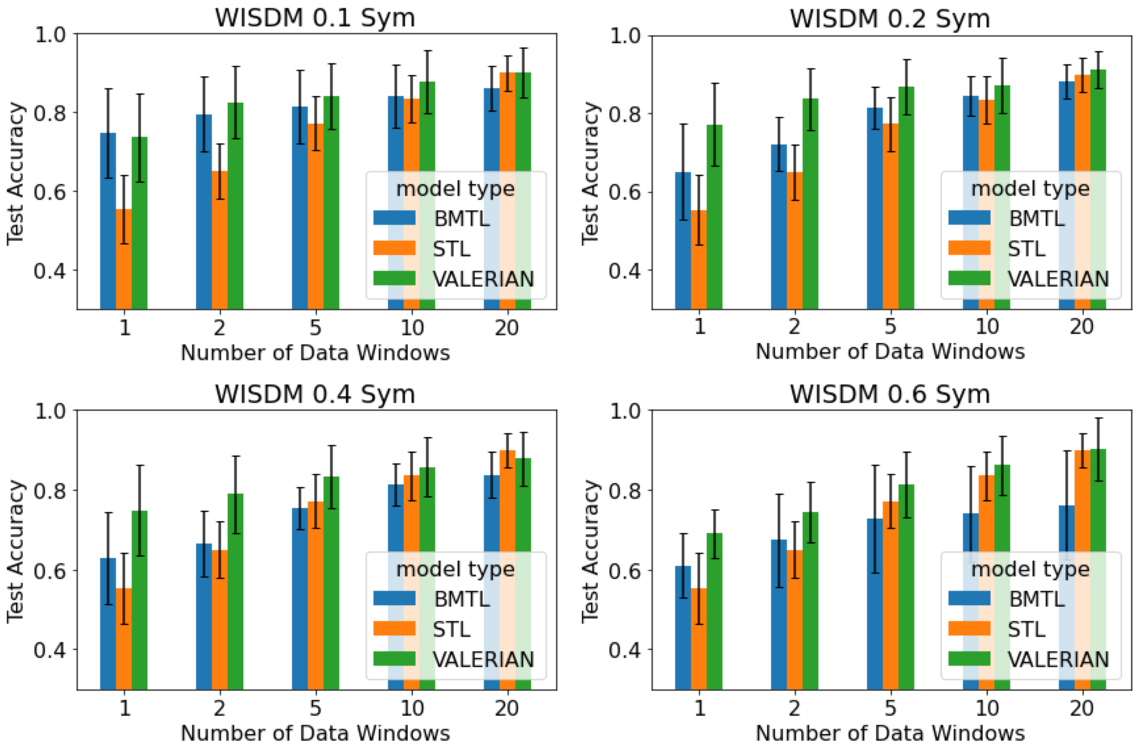


Figure 4.13: Evaluation on WISDM with different level of symmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

in contrast to the more pronounced degradation of BMTL in 1 or 5-shot learning with asymmetric noise. Our observation is consistent with what the authors reported in [16] in that ELR handles asymmetric noise well in image classification tasks and even better than scenarios with the symmetric noise in some cases. Recall that asymmetric noise in our experiments is from similar classes. These scenarios represent realistic and hard cases for LNL.

Results of non-meta-learning methods Table 4.2 compares the results of SI, SI-ELR and Butterfly against VALERIAN with 5-shot learning for the two datasets.

From Table 4.2, it is clear that none of the three methods performs well in

Table 4.2: Results of non-meta learning methods on UHCHAD and WISDM, when 5 clean labeled samples per class are available from the target domain. Report with mean test accuracy in (%) with standard deviation.

(a) Results on USCHAD dataset with artificially added 8 types of noise patterns in data labels.

Dataset	USCHAD							
Noise type	Sym.				Asym			
Method/Noise ratio	10%	20%	40%	60%	10%	20%	30%	40%
SI	74.6 ± 14.9	72.1 ± 16.1	69.6 ± 15.9	65.1 ± 15.1	74.3 ± 16.1	73.4 ± 13.6	68.7 ± 13.6	65.3 ± 14.3
SI-ELR-last	30.8 ± 11.9	26.5 ± 9.6	23.5 ± 13.2	15.1 ± 2.9	59.1 ± 30.4	48.0 ± 22.5	27.0 ± 10.5	18.0 ± 4.5
SI-ELR-best	76.3 ± 15.1	71.1 ± 12.4	65.2 ± 8.8	58.3 ± 13.1	77.8 ± 15.4	70.5 ± 17.4	69.7 ± 20.4	58.5 ± 13.2
Butterfly	67.2 ± 22.5	66.3 ± 21.7	67.1 ± 22.1	21.3 ± 6.3	65.1 ± 15.2	52.4 ± 25.5	42.0 ± 28.1	37.9 ± 15.1
VALERIAN	84.8 ± 7.3	85.3 ± 8.4	83.6 ± 9.4	82.0 ± 7.6	85.7 ± 6.6	84.9 ± 8.1	84.8 ± 8.8	83.7 ± 8.2

(b) Results on WISDM dataset with artificially added 8 types of noise patterns in data labels.

Dataset	WISDM							
Noise type	Sym.				Asym			
Method/Noise ratio	10%	20%	40%	60%	10%	20%	30%	40%
SI	61.1 ± 14.4	59.9 ± 13.8	56.8 ± 12.1	50.9 ± 12.1	58.7 ± 17.4	56.8 ± 15.3	53.1 ± 13.7	48.4 ± 11.4
SI-ELR-last	29.1 ± 11.4	27.0 ± 10.5	16.7 ± 4.4	16.8 ± 4.2	36.8 ± 17.2	24.5 ± 10.4	17.1 ± 3.1	16.9 ± 4.4
SI-ELR-best	68.2 ± 12.8	60.1 ± 12.8	52.5 ± 10.7	46.0 ± 13.2	66.1 ± 11.5	65.2 ± 8.8	58.6 ± 11.9	55.1 ± 9.7
Butterfly	58.4 ± 21.4	46.0 ± 13.2	42.3 ± 12.7	39.1 ± 9.7	58.0 ± 14.7	36.7 ± 13.2	24.5 ± 15.4	14.3 ± 1.4
VALERIAN	84.1 ± 8.3	86.8 ± 7.0	83.3 ± 8.0	81.3 ± 8.3	84.8 ± 8.7	84.4 ± 6.5	83.7 ± 8.0	82.6 ± 8.0

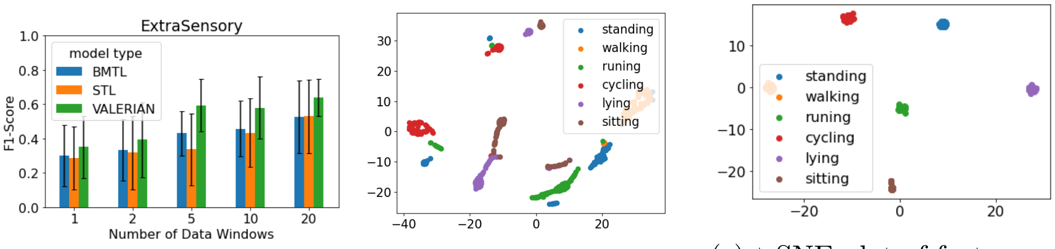
HAR with noisy labels. The vanilla SI model does not explicitly handle subject divergence nor label noises. Its performance degrades as the noise ratio τ increases. In comparison, SI-ELR ignores subject divergence and deals with noisy labels using a regularization term. In the table, we consider the results of two variants: SI-ELR-best and SI-ELR-last, where after the training epochs, the best performing model (based on clean validation data) or the model in the last epoch are saved, respectively. Note, in practice, we cannot decide when to stop training to obtain SI-ELR-best, and thus its results are presented for reference only.

We observe significant differences between results from SI-ELR-best and SI-ELR-last. This is because the high variance of SI-ELR over training epochs. Though designed to handle label noise, SI-ELR-best has worse performance than SI when the symmetric noise level is greater than 10%. The results are consistent with our observations with DivideMix in Section 4.2 and reveal that subject diversity harms ELR's ability to combat label noises. SI-ELR fares somewhat better for asymmetric noise. However, with 40% noise, SI-ELR-best is 7% worse than SI and 25% worse than VALERIAN in USCHAD. On the other hand, at the 40% noise level, ELR-last achieves an average accuracy of 17.97 (16.91) on USCHAD (WISDM), which is only slightly better than random guess with 10% (14.28%) for USCHAD (WISDM) since the number of classes in USCHAD (WISDM) is 10 (7).

Butterfly on average has worse accuracy than SI and SI-ELR-best and performs poorly as the noise level increases in both datasets. This is in part due to the fact that Butterfly uses unlabeled target domain data at training time while SI and SI-ELR-best benefit from transfer learning with a few shots of clean labeled data at inference time. However, the difference in accessing target domain lables does not justify the large variance in Bufferfly's test accuracy on USCHAD as shown in Table 4.2. As an example, with 0.4 symmetric noise, its highest test accuracy is 91.69% when data from subject 5 is in the test set, whereas its lowest accuracy is 17.77% when leaving subject 10 out. We believe that the poor performance of Butterfly is because it treats different subjects in the training set as a single domain.

In-the-wild Dataset

Next, we compare the performance of VALERIAN, BMTL and STL on ExtraSensory, which is a pure crowdsourcing dataset. Considering the data imbalance and class missing issue, we take F1-Score rather than test accuracy as metrics to evaluate model performance here. Note that since the ground truth labels from curated data are noisy, the results need to be taken with a grain of salt. Fig. 4.14(a) shows the F1-Score of the three models with gradually increasing the number of data windows. Compared to results with the two controlled datasets, all methods have their worst accuracy. This can be attributed to the noisy target domain labels in fast adaption or learning STL model. The large standard deviation in STL results at even 20-shots indicates either label noise in target domain data or noise in ground truth or both. In fact, for many subjects, the training and validation set is not i.i.d due to data noise, resulting in a validation accuracy jumping back and forth between training epochs. However, VALERIAN still outperforms the other two methods in all cases. To see if VALERIAN can indeed learn good features from noisy data, we show in Fig. 4.14(c) the t-distributed stochastic neighbor embedding (t-SNE) plot of the outputs of its feature extraction network. Clearly, the classes are well separated. This is in contrast with overlapping among classes in 4.14(b), which shows the t-SNE plot of the outputs from the feature extraction network in BMTL.



(a) F1-Score.

(b) t-SNE plot of features in BMTL.

(c) t-SNE plot of features in VALERIAN.

Figure 4.14: Evaluation on ExtraSensory with different the number of data windows per activity class from $\mathcal{D}_t$. The mean and standard deviation F1-Scores are averages across all subjects in leave-one-out experiment. t-SNE are generated on a random subject (id:4FC32141-E888-4BFF-8804-12559A491D8C) with data from all six classes.

Table 4.3: Ablation study of VALERIAN on USCHAD with 0.4 asymmetric noise.

(a) Taking 1 data window per class from the target domain.

Method	**Test Accuracy**
VALERIAN	72.29 ± 12.42
VALERIAN w/o ELR	62.21 ± 11.09
VALERIAN w/o self-supervised pre-train	63.58 ± 5.25
VALERIAN w/o MixUp	63.35 ± 3.29
VALERIAN w/o IFLF	50.74 ± 10.85

(b) Taking 5 data windows per class from the target domain.

Method	**Test Accuracy**
VALERIAN	83.68 ± 8.18
VALERIAN w/o ELR	76.55 ± 6.69
VALERIAN w/o self-supervised pre-train	79.28 ± 7.37
VALERIAN w/o MixUp	77.69 ± 2.34
VALERIAN w/o IFLF	60.18 ± 10.35

Ablation Study

To see how each component contributes to the final performance of VALERIAN, an ablation study is conducted on the USCHAD dataset with 1-shot and 5-shot learning respectively, with 0.4 asymmetric noise. Similar results can be expected for other noise settings or datasets.

As shown in Table 4.3(a) and (b), the performance gap of VALERIAN and its variant by removing one of its components becomes more prominent. But the overall contribution of each component to VALERIAN almost remain unchanged. In both cases, the domain invariant feature learner plays the most important role in VALERIAN. Without IFLF, VALERIAN degrades to an ELR model and fails to deal with subject divergence. Moreover, in absence of a dedicated meta-learning strategy, it is insufficient to update parameters of the whole model by only a few clean labeled data samples. As a result, a large standard deviation in test accuracy is observed. MixUp contributes a $\sim 9\%$ and $\sim 6\%$ accuracy to the overall solution in Table 4.3(a) and (b), empirically demonstrating its usefulness in improving model generalization in HAR tasks with noisy labels. Inclusion of ELR in VALERIAN leads to a $\sim 10\%$ improvement in 1-shot learning and $\sim 7\%$ in 5-shot ones. Recall the poor performance of ELR alone in Table 4.2. The results speak unequivocally for the need to combine LNL and meta learning to handle subject diversity. Lastly, we find that self-supervised pre-train contributes $\sim 9\%$ and $\sim 4\%$ test accuracy respectively.

4.6 Conclusion

In this chapter, we proposed VALERIAN, a domain invariant feature learning approach for sensor-based HAR in the wild. An extensive experimental study demonstrated its superior performance over baseline methods for different levels of noise and noise patterns. The key takeaway from this work is two-fold: 1) the effects of subject diversity and label noises intertwine in the learning behaviour of LNL models and can lead to catastrophic memorization of wrongly labelled data, and 2) it is important to design domain adaptation strategies to explicitly handle subject diversity in conjunction with LNL for better generalization in HAR.

Chapter 5

A Deep Learning-based Cross-modality Inertial Measurement Unit Simulator

5.1 Introduction

Nowadays, inertial measurement units (IMUs) have become ubiquitously available in wearable and mobile devices. An important category of IMU-enabled applications is monitoring and assessing human mobility, which aims to continuously track people's daily activities, analyze motion patterns and extract digital mobility biomarkers such as gait parameters in the wild. Increasingly, data-driven deep learning models have been developed for human activity recognition (HAR) [35, 126] and human pose estimation (HPE) [36]. Despite their impressive performances, these models generally require a large amount of sensory data for model training. Unfortunately, it is challenging to collect high-quality IMU data in the wild. Moreover, data collected from controlled settings where subjects are asked to perform certain activities often have very different characteristics from those in freestyle motions [3]. On the other hand, annotating IMU data post hoc is challenging as raw IMU signals are hard to interpret even by domain experts.

The scarcity of IMU data for human mobility assessment is evident when compared with the richness of other data sources. PAMAP2 [23], a benchmark dataset for HAR, consists of 8 subjects with only 59.67 minutes of samples per person. In contrast, AMASS [24], a motion capture (MoCap) dataset, includes 2420.86 minutes of data and is still growing; not to mention YouTube videos, which offer a practically infinite amount of action data. Therefore, to mitigate the "small data" problem, one possible solution is to convert data from other modalities to IMU, a process called *cross-modality simulation*.

Though several previous works explored the feasibility of simulating IMU sensor data from other data modalities (see Section 5.2), two fundamental challenges remain. First, sensors are attached to human skin rather than directly to bone joints during data collection. Skeleton models are inadequate in representing human poses and shapes. Second, even with state-of-the-art (SOTA) solutions in computer vision, the extracted 3D human motion trajectories from monocular video clips remain inaccurate. Analytically compute IMU readings on such imperfect input sequences will result in large errors. However, if a deep learning model is adopted to learn the mapping between noisy motion trajectories and measured sensor readings, it is unclear how well such models generalize to arbitrary unseen on-body positions.

To tackle the aforementioned challenges, we design and implement CROMOSim, a cross-modality IMU sensor simulator that simulates high fidelity virtual IMU sensor data from motion capture systems and monocular RGB cameras. It differs from existing work in two important aspects. First, it is based on the 3D skinned multi-person linear (SMPL) model [127], which serves as an intermediate representation of motion sequences and entitles our CROMOSim for an arbitrary on-body simulation. SMPL has been widely used in HPE tasks [128, 39, 129, 130], which is capable of modelling muscle and soft tissue artifacts. In contrast, the 2D or 3D skeleton representations adopted by other works are segment models without volumetric information. Second, we empirically demonstrated that the direct computation of IMU readings from motion trajectories extracted from videos is unreliable (in Section 5.4), even with filtering and interpolation techniques as the case of IMUSim [131]. We instead design and train a neural network to learn the relationship between measured IMU readings and the noisy motion trajectories. Special cares have to be given to ensure the trajectories are represented in a consistent global coordinate frame even if the videos are captured by moving cameras. Compared to existing IMU simulators, experiments show that CROMOSim achieves higher fidelity and superior performance in various HAR tasks. HPE tasks are also evaluated to demonstrate the utility of simulated data in downstream applications.

In summary, we make the following contributions in this chapter:

1. CROMOSim is the first work that utilizes SMPL full-body tri-mesh as an intermediate representation for IMU data simulation.
2. CROMOSim offers a generic pipeline for generating IMU readings at *arbitrary* on-body locations from either MoCap or monocular RGB data. It is readily extensible to other input modalities and configurations.
3. CROMOSim mitigates imperfection in intermediate body pose and shape estimations through a supervised learning approach.
4. Compared to SOTA IMU simulators, CROMOSim achieves higher fidelity and superior performance in HAR tasks.
5. We are the first to empirically show the utility of simulated IMU data in HPE tasks under a deep learning context.

The rest of the chapter is organized as follows. Section 5.2 describes related work. In Chapter 5.3, we introduce the CROMOSim pipeline and details of each component. In Section 5.4, we present the implementation details and performance evaluation of CROMOSim standalone and in downstream tasks, respectively. Finally, we conclude the chapter in Section 5.5 with discussion and future work.

5.2 Related Work

The proposed cross-modality simulation framework is a type of data augmentation technique, which is broadly used in machine learning to compensate for data scarcity, to improve data diversity, and boost the generalization of a trained model. In the context of augmenting IMU data, we categorize existing methods into three groups: transforming real IMU recordings, generative models for IMU data, and cross-modality simulators.

IMU transformations Simple operations such as flipping, rotation, scaling and changes in brightness can be applied to augment image data. Similar ideas are applicable to IMU data as well. In [17, 18], random relative rotations between a sensor and human body were added within a predefined range, to make the trained model robust to subject divergence. In [19], the authors proposed a systematic way to augment the IMU data via rotation, permutation, time-warping, scaling, magnitude-warping and jittering. Eyobu *et al.* went even further in [20] to transform handcrafted features rather than the raw recordings of wearable sensors. Though easy to implement, IMU transformation methods require the availability of sufficient real sensor data as their source.

Generative models for IMU data Generative adversarial networks (GAN) use two neural networks, pitting one against the other in order to generate new, synthetic instances of data that is indistinguishable from real ones [132]. Researchers designed GANs to generate IMU data in [21, 22, 133]. SensoryGANs [21] adopt adversarial learning in generating diverse yet realistic IMU sensor readings for locomotion. However, this method is highly complex: a different neural network architecture is devised for each activity. Moreover, due to large variances in the generated data, it is only suitable for relatively simple HAR tasks with easily separable patterns–both SensoryGANs and ActivityGANs [133] simulate only *stay still, walk, jog* activities in their HAR evaluation.

Cross-modality IMU simulation Given motion trajectories in a global frame, acceleration can be calculated by taking the second derivatives of positions over time. Researchers may take advantage of this simple computation strategy to simulate accelerometer data from MoCap sequences. The resulting data has been used in recent works to pre-train human pose estimation (HPE) [134] and HAR models [135, 136]. One drawback of this method is that none of these researches targets to simulate realistic IMU sensor readings, and gyroscope data is omitted. For a more systematic IMU simulation, IMUSim [131] is among the first open-source tool to simulate IMU data from either MoCap data in the Biovision Hierarchy (BVH) format or a user-provided 3D position and orientation sequence. Though employs data smooth and filtering techniques to tackle outliers, this method is built upon analytically calculation with low data fidelity (see Section 5.4.2).

After that, simulating IMU readings from monocular RGB videos for data augmentation has attracted some attention in recent years. ZeroNet [137] extracted finger motion data from videos, then transformed them into acceleration and orientation information measured by IMU sensors. The authors of [138] and its follow-up work [139, 140] simulated acceleration norms and/or angular velocity norms from human 2D poses for a HAR purpose. In their latest work [140], Rey et al. skipped the video processing steps and directly mapped vision data to IMU readings with placement specific neural networks. These works avoid the video-based global motion tracking by limiting human subjects‘ movement to a fixed camera scene (in-place motion), and thus cannot be applied to handle in-wild video data with moving cameras. Closest to our work are IMUTube [1] and its extension in [141], which aim at simulating full-body IMU data from moving camera videos captured in the wild. But limited by the skeleton body representation adopted, neither work can simulate realistic sensor readings from arbitrary on-body locations. Moreover, in IMUTube, the estimation of view depth and camera ego-motion is in two independent steps though the two are intrinsically coupled [142, 143]. A wrongly predicted camera pose can lead to inaccurate view depth estimation and vice versa [144]. In addition, the lifting of 2D postures to 3D module in IMUTube pipeline is more compute-intensive and error-prone, as it is a simple combination of existing technologies.

5.3 Deep Learning-based Cross-modality IMU Simulator

Before introducing the proposed method, the notations used in this chapter are defined as follows. There are four different coordinate frames involved in this work: F^G for the global tracking frame, which is a fixed coordinate system for representing objects in the world. F^B for a bone coordinate frame defined as originate at its distal joint and take the direction along the bone is x positive while z positive orthogonal to its upper surface and points outside. F^S is the sensor frame that is fixed on the sensor and is determined by its manufacturer. F^C for the camera frame that takes the center of the camera's image plane as its origin and the optic axis as the Z-axis (Fig. 5.1). Rotation matrix R_B^S denotes the rotation from bone frame to sensor frame.

For simplicity, amongst camera intrinsic parameters , we assume the optical centers of the camera in pixels on the x and y axis cx = cy = 0, and only estimate the focus length in the x and y axis f_x and f_y. Camera extrinsic parameters include rotation matrix R and translation vector t, respectively. R and t are fixed for fixed cameras and need to be updated for moving cameras. During movements, both F^B and F^S changes relative to F^G and are placement or device specific. Therefore, it is necessary to transform representations of motions into a unified global coordinate first.

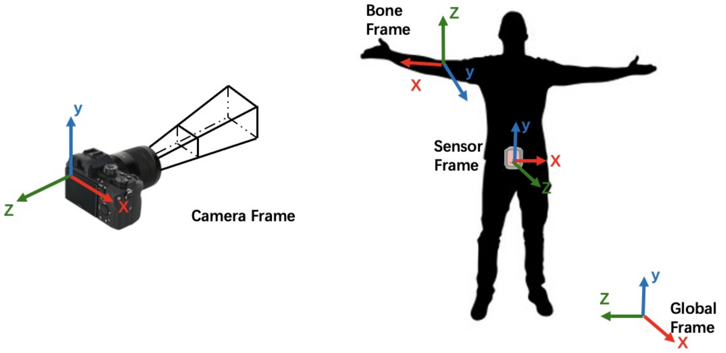


Figure 5.1: An example of different coordinate frames involved in this work.

5.3.1 Overview

CROMOSim is designed with several requirements in mind: i) allowing arbitrary user-specified placement and orientation of target sensors, ii) extensibility to different input data modalities and configurations, iii) flexibility to incorporate SOTA models to extract motion trajectories, and iv) high fidelity. To meet these requirements, the CROMOSim pipeline consists of three function modules as shown in Fig. 5.2 : an input data processing module that extracts global human motion sequences from

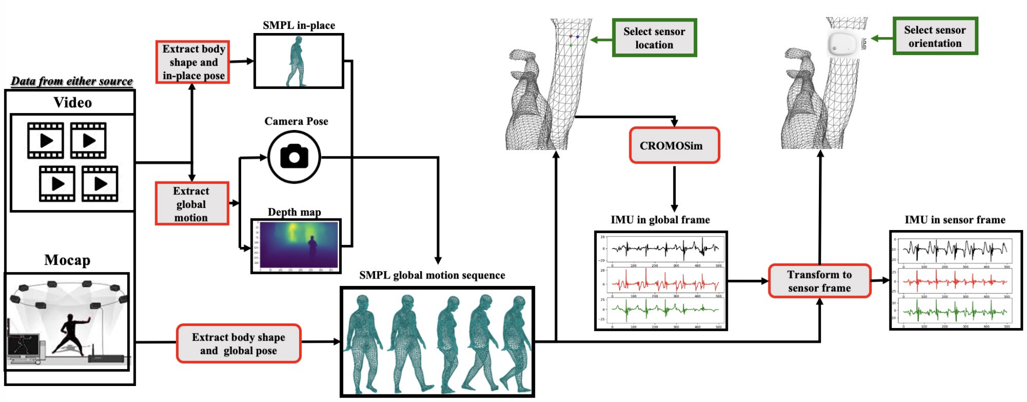


Figure 5.2: The proposed CROMOSim pipeline. It takes either MoCap or monocular camera video data as input and converts them into SMPL represented global motion and body shape. The simulator then takes the SMPL model, specified sensor placement and orientation as input; predicts simulated IMU readings and transforms them back to the sensor coordinates frame.

source data, a human body model that fully represent the extracted sequences and can be sampled from any on-body location, and a simulator module that transforms noisy motion sequences into high-fidelity 3-axis accelerometer and gyroscope readings. Though the pipeline is extensible to other possible input data modalities such as millimetre wave radar and depth camera, we will focus on MoCap and monocular camera video here. Each component will be discussed in detail in the remaining Chapter.

5.3.2 SMPL Model

An SMPL model represents 3D human body poses and shapes with a fine-grained full-body tri-mesh. Unlike skeleton or cylinder models that only capture joint poses, this parametric 3D representation provides a widely applicable and differentiable way to visualize a realistic 3D human body. There are three reasons to choose SMPL over other body models in CROMOSim. First, instead of measuring the movements of bones, IMU readings reflect the soft tissue dynamics at the location to where a sensor is attached. Second, SMPL provides a pose and shape-dependent full-body tri-mesh that can be sampled at any on-body location. Third, since it is widely used in HPE research, many off-the-shelf models are available to extract SMPL representations from different data sources.

To see the difference between movements of joints in a skeleton model and SMPL skin mesh, we compare accelerations computed by taking second-order derivatives of the corresponding motion trajectories and ground-truth accelerometer readings over time. In Fig. 5.3, red curves denote the calculated 3-axis accelerations while the black ones are accelerometer ground truth. Figures in the left column compare the accelerations at the pelvis joint in a skeleton model while figures in the right column compare those at SMPL lower back skin mesh vertices. Clearly, the use of the SMPL skin mesh provides better agreements with the ground truth (e.g., in the interval [100,300]). Simulated data from the pelvic joint, on the other hand, fails to capture high-frequency acceleration components, which are most likely due to muscle and soft issue movements. SMPL enables users to sample from any on-body position on the skin surface while the skeleton model represents the motion of bones only. In most cases, IMUs are attached to body surfaces rather than directly to bones or anatomical landmarks. This observation indicates that SMPL is a good candidate for an intermediate data representation of the CROMOSim pipeline.

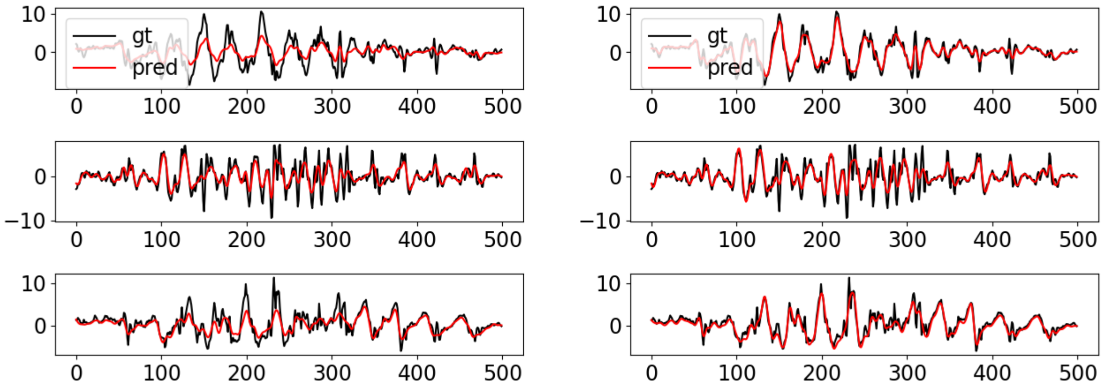


Figure 5.3: Comparison between analytically computed 3-axis accelerations from a skeleton representation and an SMPL model. Left: taking the motion sequence of pelvis joint positions as input, right: taking the motion sequence of SMPL lower back skin mesh positions as input.

5.3.3 Input Data Processing

From MoCap Data to SMPL Models

MoCap data consists of raw marker sequences collected by an optical motion capture system of high precision (usually with a position error $<$ 1 mm). With commercial Mocap systems like OptiTrack [145] and Vicon [146], both body shape and pose data can be captured. Such data has been widely used in gaming and movie industries [147]. MoCap data is commonly used as ground truth labels in markerless human pose estimation with cameras or wearable sensors [36]. MoSH++ [24] allows the fit of an SMPL model to MoCap data from a set of sparse markers. Prior to motion capture, a global tracking coordinate system needs to be established during the calibration phase. As a result, the collected motion trajectories are expressed in the defined global frame. Under the assumption that the global frame is aligned with the inertial frame[1] , the SMPL mesh model can be used directly in subsequent processing.

From Video Clips to SMPL Models

Extracting 3D human poses and shapes from monocular RGB videos is not trivial, especially when they are captured from moving cameras with unknown parameters, which is common in a locomotion-related video recorded in the wild. We propose to decompose such a problem into two sub-problems: a reconstruction of human global displacement and rotation; and an estimation of 3D in-place human motion and body shape.

Estimating root joint global trajectory A precise calculation of global displacement for the human subject is essential for a high-fidelity simulation of IMU data from RGB videos. This requirement can be achieved by reconstructing the 3D motion trajectory of a fixed body position (a.k.a, the root joint), which can be inferred from the per frame depth map of the human subject and known camera parameters [148].

In CROMOSim, we adopt robust consistent video depth estimation (Robust CVD) method [142], a SOTA model to estimate consistent dense depth maps and camera poses from a monocular video. Robust CVD jointly estimates both outputs by solving an optimization problem over the entire video sequence. It is advantageous as the two outputs are intrinsically coupled and thus lead to higher accuracy (compared to the pipeline adopted by IMUTube). In the implementation, we locate the 2D torso joint positions in video frames using OpenPose [149], and designate the pelvis

[1] Such an assumption is not restrictive as a random rotation can be applied in further data augmentation to obtain data if the global and inertial frames differ.

as our root joint. With the detected 2D joint position and depth map per video frame, we can calculate the global 3D torso coordinates as follow. Denote the 3D coordinates of the root joint in the camera frame and the global frame at time k by $P^C(k) = [X^C(k), Y^C(k), Z^C(k)]$ and $P^G(k) = [X^G(k), Y^G(k), Z^G(k)]$ respectively. Let its corresponding 2D pixel coordinates in the camera image be $[x(k), y(k)]$. Given the camera intrinsic parameters f_x and f_y from robust CVD, we have

$$\begin{aligned} X^C(k) &= \frac{(x(k) - \frac{W}{2}) \times Z^C(k)}{f_x} \\ Y^C(k) &= \frac{(y(k) - \frac{H}{2}) \times Z^C(k)}{f_y} \\ Z^C(k) &= d(x(k), y(k)), \end{aligned} \tag{5.3.1}$$

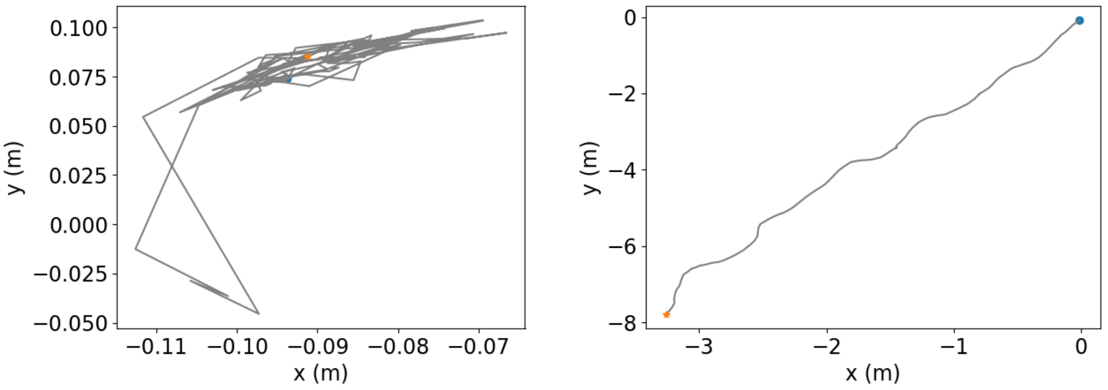
where $d(x, y)$ is a depth retrieving function with a 2D pixel coordinates x, y, and W and H are the width and height of the pixel image. Next, using the camera extrinsic parameters R_k and t_k, we transform the root joint position from the camera frame F^C to global frame F^G at time k follows:

$$P^G(k) = R_k^T \times (P^C(k) - t_k) \tag{5.3.2}$$

In addition, depth reconstructed by robust CVD is reasonably accurate up to scale. To resolve scale ambiguity, an object of known size (its real height h_r or real width w_r) in the scene is needed, as real depth at time k can be calculated with $d_r(k) = (f_y \times h_r)/h_p(k)$, where $h_p(k)$ is the object height in pixels. The scale factor can be estimated with $s = d_r(k)/d(x(k), y(k))$, and it is a constant value per video clip processed by Robust CVD. Prior knowledge regarding heights of subjects in the video, or dimensions of fixtures (e.g., street lamps, road lanes) can be utilized. Subsequently, the predicted depth of the pelvis joint is re-scaled by the estimated scale factor to recover the real global root joint trajectory.

Since in some frames, the root joint is not visible or cannot be located accurately due to occlusion or poor lighting, we only extract root joint coordinates from the frames with high confident scores by OpenPose. Root joint coordinates in the remaining frames are then interpolated from the estimated ones, and a Kalman filter is applied to further smooth the resulting trajectory.

Body pose and shape estimation in camera frames We adopt VIBE [39], a SOTA method to directly estimate realistic 3D human poses and shapes from monocular videos. In the implementation, we make two extensions to VIBE. First, VIBE assumes a fixed camera configuration and in-place human motion only, losing track of human subjects' global motion trajectory. Fig. 5.4 shows the difference between motion trajectories of a lower back SMPL mesh vertex near a subject's pelvis. The figures are extracted by VIBE only, and by our proposed pipeline, respectively, when the straight-line running subject was captured by a handheld camera. Clearly, the trajectory in the left figure fails to reflect the actual motion. As elaborated in the previous paragraph, robust CVD is adopted to complete the missing information. It helps to estimate the 3D global translation of the subject's root joint per video frame even if there is relative motion between the camera and the human subject. We acquire a full 3D human pose representation by adding the global translation to the translation parameters of the SMPL model from the VIBE output. Second, VIBE estimates body shapes for every video frame and a frequent re-scaling of the human subjects can be observed when there are drastic motions or the camera is moving fast. This is unnecessary since people's body shapes are unlikely to change



x

Figure 5.4: Extracted motion trajectories of a subject's low back from a 4-seconds running outdoor video clip captured with a handheld camera. The subject in the video runs along a straight line. Left: results from VIBE only, right: results from the proposed pipeline. Each plot was generated by projecting 3D trajectory on the ground plane.

in a short period and is prone to errors. Instead, we assume that the estimated body shape estimation can be modeled as the ground truth shape plus zero-mean random noise. Thus, shape estimation errors can be mitigated by averaging the estimated body shapes for the same subject in each frame in a 10-second video sequence.

Finally, by combining the aforementioned steps, we can extract 3D body poses in a global frame and shape parameters from monocular RGB video, which can serve as input to generate SMPL body meshes.

5.3.4 From SMPL Models to IMU Data

Given the 3D human pose and shape represented by SMPL tri-mesh over time, accelerations and angular velocities in a global frame can be computed analytically. In particular, accelerations can be calculated by taking second derivatives of positions over time; angular velocities can be determined from the changes in the normal vector of a plane associated with three non-collinear mesh points (e.g., the vertices of a mesh triangle). However, SMPL tri-meshes generated by the models in Chapter 5.3.3 tend to be noisy, erroneous and incomplete. Furthermore, accelerations and angular velocities measured by IMUs are subject to hardware imperfection such as noises, biases, and non-orthogonal axes, which are not easily replicated by analytical calculation.

To address the aforementioned issues, we design two neural network models, an accelerometer and a gyroscope network, to learn the mapping between motion trajectories of SMPL tri-mesh points and actual acceleration or angular velocity measured by IMUs in a global frame, respectively. The neural networks are capable of generating data from any arbitrary unseen region over the human body by training with real data from some selected on-body positions of various motion ranges (such as the head, chest, one side of the wrist, and ankle). Both models take the same design, with three convolutional and two bidirectional long-short term memory (LSTM) layers as the feature extractor, and a following linear layer for regression output. The model is fed a user-specified skin area, with three mesh triangles chosen near the area's center as input. In each triangle, the vertices are traversed counter-clockwise to ensure the norm direction always points outside of the human body.

The collected IMU data are usually in the local sensor frame while the predictions of CROMOSim are in the global frame. Therefore, a coordinates transformation step is required. A user needs to select the skin region a virtual sensor affixes to and define its alignment represented as a rotation matrix (R_S^B). With the rotation matrix from the bone frame to the sensor frame $(R_S^B)^{-1}$, we can transform IMU data into the sensor frame from the accelerations $\mathbf{a}_G$ and angular velocities ω_G in the global frame as follows:

$$\mathbf{a}_S = (R_S^B)^{-1} \times (R_B^G)^{-1} \times (\mathbf{a}_G + g), \tag{5.3.3}$$

and

$$\omega_S = (R_S^B)^{-1} \times (R_B^G)^{-1} \times \omega_G, \tag{5.3.4}$$

where g is the gravity acceleration and R_B^G is obtained from the SMPL model for the corresponding skin region.

Due to noisy data sources and modelling errors, domain gaps exist between simulated and real data. Such gaps are more pronounced in the simulated data from videos. To mitigate these gaps, we adopt the same distribution mapping technique [150] as IMUTube. Let $G(X \leq x_r)$ and $F(X \leq x_s)$ be the cumulative density functions (CDF) for real IMU x_r and simulated data x_s, respectively. Under the assumption that $G(\cdot)$ is invertible, it can be proven that $x'_s = G^{-1}(F(X \leq x_s)$ follows the same distribution as x_r.

To apply distribution mapping, we need to estimate the CDF of simulated and real data along each axis, then apply the mapping separately. Empirical results from IMUTube show that a small number of real data ($\sim$ 1000 samples per class or equivalently 33-second long with a sampling rate of 30 Hz) are sufficient to give a good estimation of $G(\cdot)$.

5.4 Evaluation

In this Chapter, we will evaluate CROMOSim in two sets of experiments. Firstly, we evaluate the fidelity of simulated sensor data both qualitatively and quantitatively. Then, we evaluate the utility of CROMOSim in data augmentation for downstream

HAR and HPE tasks.

5.4.1 Experimental Setup

Datasets

To train the simulator network and evaluate the fidelity of simulated data, we use the TotalCapture dataset, a benchmark for 3D HPE from marker-less multi-camera capture [43] which has all three data modalities (MoCap, IMU and video). For HAR evaluation, Realworld [151], the Physical Activity Monitoring version 2 (PAMAP2) [23] and Opportunity [60] datasets are used in task model training and testing. For knee angle estimation tasks, we also take Totalcapture in our experiments. A detailed description of each dataset is listed below:

1. **TotalCapture [43]:** It is the first dataset to have fully synchronized multi-view video collected from eight RGB cameras at a frame rate of 60Hz, 12 IMU sensors (affixed to a subject's head, right and left upper arms, right and left wrists, right and left upper legs, right and left lower legs, right and left feet and pelvis) sampled at 60Hz and Vicon labels for a large number of frames (∼1.9M). It contains 5 subjects performing *acting, walking, rolling arms, and freestyle motions* indoor.

2. **Realworld [151]:** It has 8 activities including *climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking* performed by 15 subjects. Each subject wore mobile devices on 7 body positions (chest, forearm, head, shin, thigh, upper arm and waist). Videos were recorded by a moving handheld camera followed the subjects. Each activity lasted 10 minutes, except

for jumping, which was around 2-minute long. Data was collected naturally. In some indoor trials, the light conditions were poor. In some outdoor trials, the videos contain passers-by not part of the subject pool.

3. **PAMAP2 [23]:** The Physical Activity Monitoring version 2 (PAMAP2) consists of data collected from IMU sensors (accelerometer and gyroscope) on subject's chest, dominant ankle and wrist during 8 activities, i.e., *lying, sitting, walking, running, standing, rope jumping, ascending stairs and descending stairs*. Eight subjects performed these activities freely without time constraints and had the option to skip some activities. There exist missing classes in some subjects' data and the data samples are unbalanced across the classes. During data collection, IMU sensors are instrumented on different subjects at a sampling rate of 100Hz.

4. **Opportunity [60]:** The Opportunity dataset contains IMU measurements from 4 subjects during 5 mobility-related activities. The activities are *sitting, standing, lying, walking and null*, where 'null' include any activity outside the first four. Data was collected from 7 body-mounted sensors (left and right forearms, left and right arms, back, left and right feet) at a sampling rate of 30Hz.

Data Preprocessing

In the fidelity evaluation, we divide data from TotalCapture with all modalities into 2-seconds sliding windows with 80% overlapping for model training and without overlapping for prediction. For HAR, to make the results directly comparable to baseline approaches, we follow the same procedure described in IMUTube, where simulated and real IMU data are low-pass filtered, normalized and divided into sliding windows with 1-second length and 50% overlapping. In the case of HPE, the real and simulated IMU data are standardized, and then divided into 1-second windows without overlapping.

Evaluation Metrics

To evaluate the fidelity of CROMOSim, we compute the root mean square error (RMSE) between simulated IMU data and ground truth. In HAR tasks, as the classes in datasets are imbalanced, we use mean F1 score and its standard deviation to evaluate the random single-subject-out experiments. In multi-class classification, the F1 score is computed as the weighted average of the F1 score of each class. In 3D HPE tasks, we measure the RMSE between predicted knee angles against the ground truth in the unit of degrees.

Baseline Methods

We consider IMUSim and an analytical method as baselines to compare the fidelity of our simulated data because IMUTube also utilizes IMUSim to generate IMU data from 3D global motion trajectories. The analytical method we adopt to compute linear acceleration is Richardson's extrapolation [152, 153]. Compared to taking second-order derivatives, Eq. (5.4.1) gives a more accurate estimation with a 4th order error term (as opposed to 2nd order).

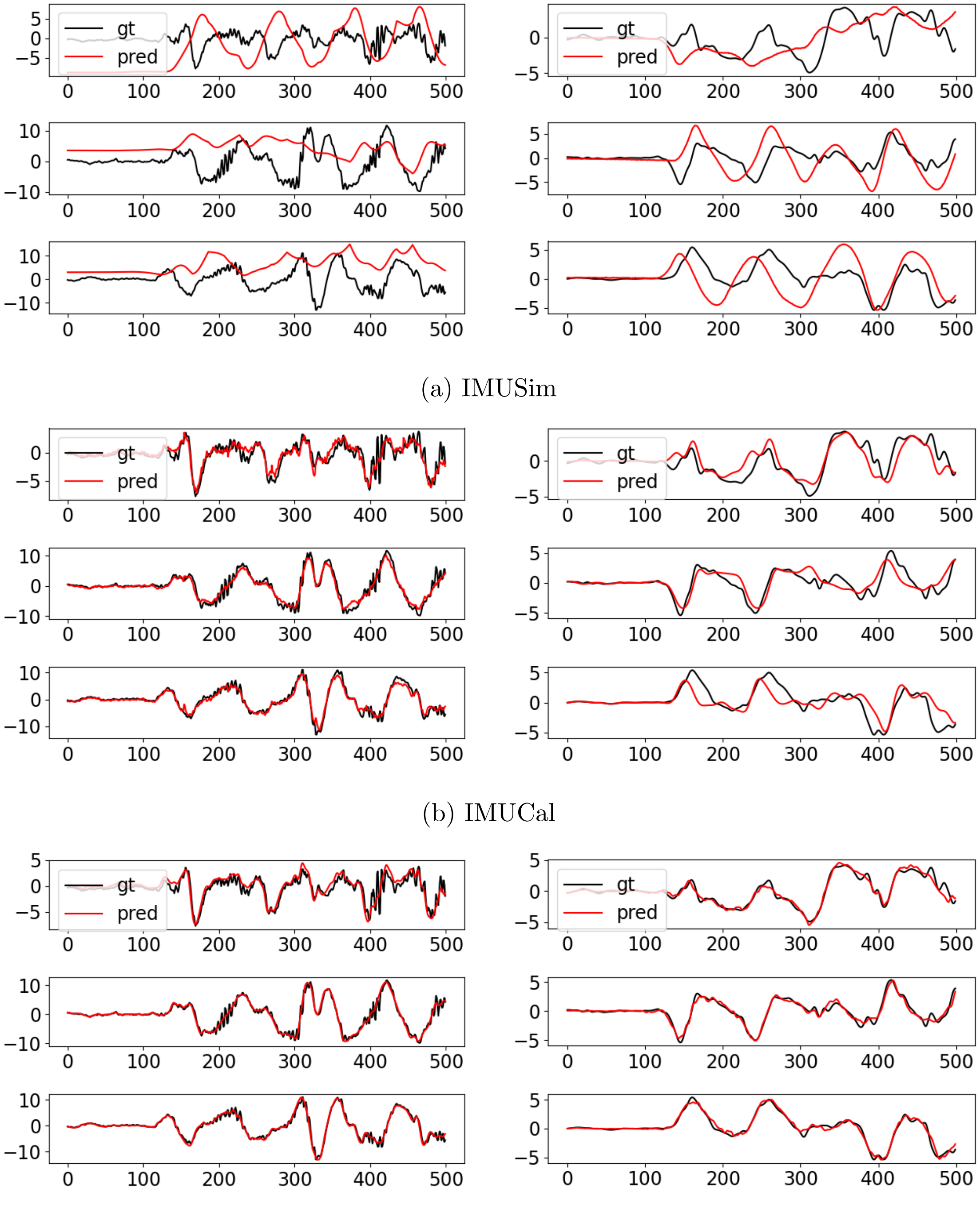
$$acc = \frac{-p(t-2) + 16p(t-1) - 30p(t) + 16p(t+1) - p(t+2)}{12\Delta t^2} \tag{5.4.1}$$

The angular velocity of a selected skin region on an SMPL body mesh is calculated by tracking the rotation of its norm vector. The tri-mesh of SMPL model follows the right hand rule, which ensures that the norm vectors of the triangles always point out of the corresponding subject's body. Rotations between consecutive frames are expressed in unit quaternions. Angular velocities in rad/s are computed by multiplying the rotation vector of each frame with the sampling rate. To reduce jitters, we take the average angular velocities of three nearby triangles on the tri-mesh centred in the designated skin region. Lastly, a 4th order ButterWorth low-pass filter is applied to both simulated accelerometer and gyroscope readings for noise reduction [91].

For HAR tasks, we take IMUTube as the baseline, but due to the lack of open source implementations, we include the reported performance on PAMAP2 and Opportunity datasets from [1].

5.4.2 Fidelity of CROMOSim

In this Chapter, we first provide qualitative and quantitative comparisons between CROMOSim and two baseline methods, namely, the analytical method (IMUCal) and IMUSim in terms of fidelity. We use TotalCapture in this experiment since it contains data from all three required modalities. Two sets of CROMOSim models are trained using MoCap and video data from Subjects 1 – 3 with sensor positions at their right wrist, right foot and pelvis. The models are used to predict accelerometer and gyroscope data on both left and right wrists of Subject 5 from the respective data sources. Next, we analyze the sources of errors in video-based simulations. Figures 5.5 and 5.6 show the simulated IMU readings from different methods with MoCap and RGB video data, respectively. In these cases, the sensor placement is

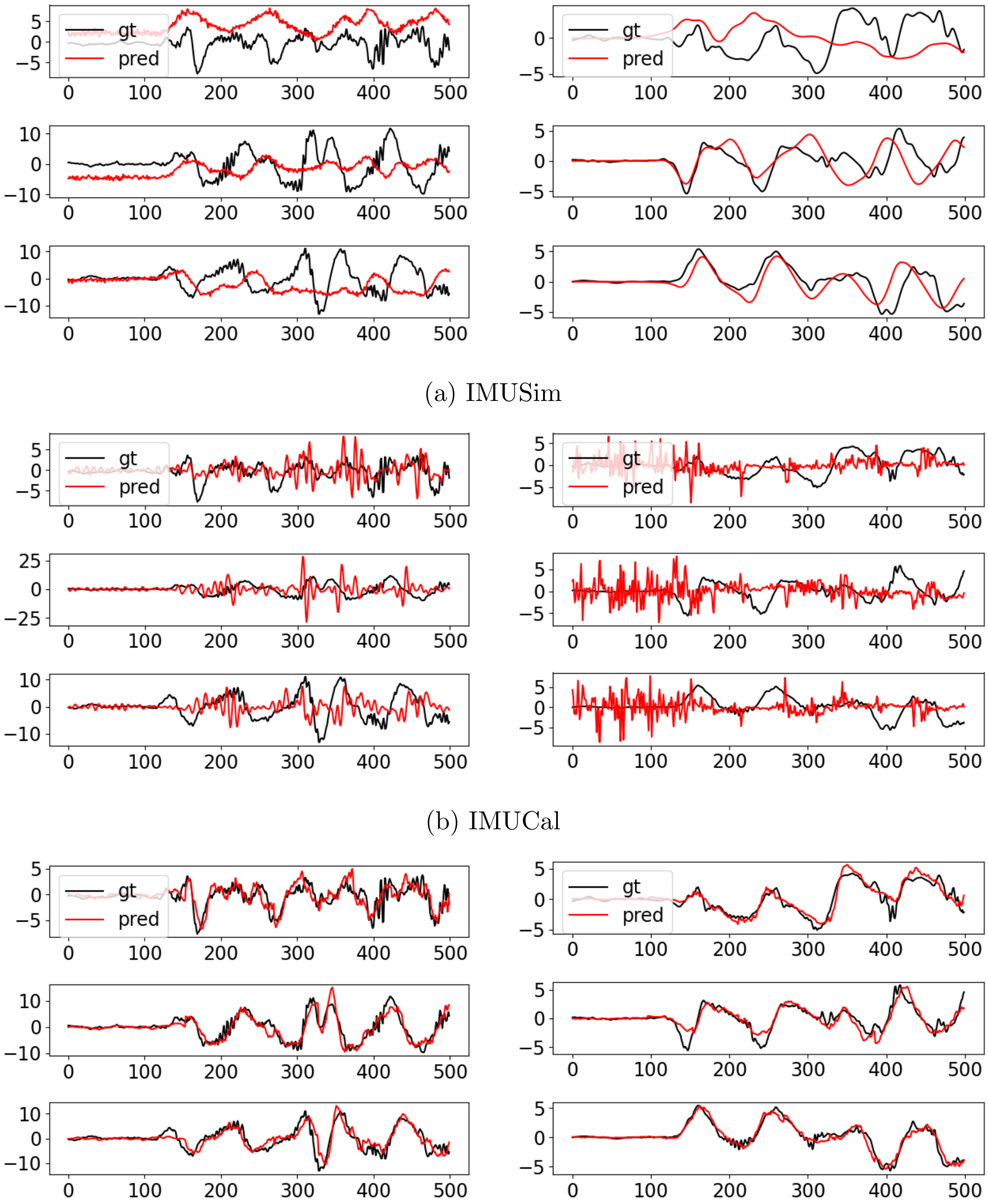


(a) IMUSim

(b) IMUCal

(c) CROMOSim

Figure 5.5: Simulated IMU readings on the right wrist of Subject 5 from the MoCap data in TotalCapture. Left: accelerometer data. Right: gyroscope data.



(a) IMUSim

(b) IMUCal

(c) CROMOSim

Figure 5.6: Simulated IMU readings on the right wrist of Subject 5 from monocular RGB camera video in TotalCapture. Left: accelerometer data. Right: gyroscope data.

Table 5.1: RMSEs of simulated IMU readings on Subject 5's left wrist across all data trials.

	Acceleration (m/s^2)			Angular velocity (rad/s)		
	IMUSim	IMUCal	CROMOSim	IMUSim	IMUCal	CROMOSim
MoCap extracted SMPL	4.606	1.785	1.602	1.500	1.272	0.801
Video extracted SMPL	6.158	11.824	3.342	1.848	2.578	1.104

known but the subject is unseen to the simulator model. From the figures, we observe that the fidelity of IMUSim is low across the board. It is because the default setting of IMUSim filters out too much high-frequency components. IMUCal works well for simulating accelerometer and gyroscope data with MoCap inputs. However, its performance significantly degrades when monocular RGB videos are taken as the source modality. This can be attributed to large noise and relative low accuracy of extracted SMPL body tri-mesh. In contrast, CROMOSim consistently outperforms baseline methods for both data modalities.

Qualitative and quantitative results

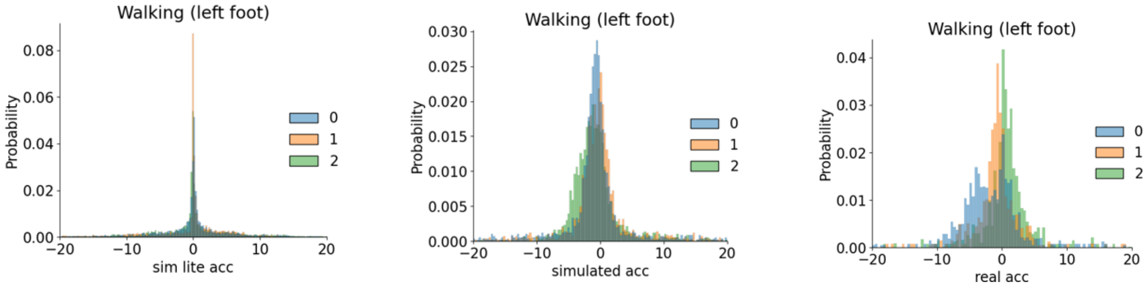
Table 5.1 reports the case where both subject and sensor position are unseen to the simulator networks. The quantitative results are consistent with those in qualitative ones shown in Fig. 5.5 and 5.6. With MoCap data, the accuracy of CROMOSim is 187.5% and 11% higher than that of IMUSim and IMUCal for accelerations, respectively, and 87% and 58% for angular velocities. The advantage of CROMOSim is more pronounced with monocular RGB videos, outperforming the next best method (IMUSim) by 84% and 67% for accelerometer and gyroscope data.

Table 5.2: The analysis of error sources with monucular camera video data.

	VIBE only				Robust CVD	GT global motion
	MPJE (rad)		PVE (m)		PVE (m)	PVE (m)
	RMSE	MAE	RMSE	MAE	RMSE	RMSE
ROM	0.2203	8.3634	0.8099	1.7451	/	/
walking	0.1972	7.9263	1.3741	1.9215	1.1679	0.5046
freestyle	0.2087	8.3627	1.1930	2.1327	0.9607	0.5015

Error Analysis

From Table 5.1, we see that simulated IMU readings from video extracted SMPL have larger errors than those from MoCap. To understand the sources of errors, we conduct further empirical study. Specifically, we analyze the effectiveness of the global trajectory estimation module for root joint, and present the results here. Table 5.2 summarizes the quality of extracted human pose data on TotalCapture dataset by three approaches, namely, VIBE indicates when the estimation of global trajectory is unavailable, Robust CVD denotes a global motion estimate by the CVD method, while GT global motion refers to align the root node position per video frame with MoCap ground truth. We take the mean per joint error (MPJE, in rad) and per vertex error (PVE, in meters) between the estimated SMPL body mesh from videos and from MoCap data as metrics here. Three types of activities are analyzed: the range of motion sequence (ROM) contains in-place motions with human subjects standing at the center of a laboratory field; the walking sequence involves a person walking around the laboratory; the freestyle sequence corresponds to a freestyle acting and roaming around the room. Clearly, ROM is not affected by global motion trajectory estimations, while the other two are. As the joint angles are extracted by VIBE only, they remain the same with Robust CVD or GT global motion.



(a) VIBE only (b) Robust CVD (c) Ground truth

Figure 5.7: Simulated accelerometer readings on the left foot of Subject 5 from monocular RGB camera video in TotalCapture.

From Table 5.2, there exists a clear gap between the PVE calculated with VIBE only and GT global motion for walking and freestyle, indicating the need to accurately estimate global motion trajectories when motions are not in-place. PVE dropped ∼20% when the Robust CVD is used in video data pre-processing. Fig. 5.7 shows the probability density function of 3-axis accelerations in a global frame from the two methods in comparison to ground truth. The plots further demonstrate that simulated data are more similar in distribution to the ground truth when global trajectories of the root node are incorporated.

The differences between the estimated global trajectory from Robust CVD and the ground truth can be attributed to two factors. First, we use OpenPose to detect the root node of human subjects in each video frame. OpenPose fails when the resolution is low and the background is complex. Two examples are shown in Fig. 5.8, where in the left figure a person is running on a trail and in the right figure he is climbing downstairs. Both fail cases are captured from Realworld dataset. The wrongly detected root node will lead to errors in extracted global motion trajectories. Second, calculation of the scale factor is another potential source of errors. To recover real world global motion trajectories from the output of robust CVD, a scale factor is

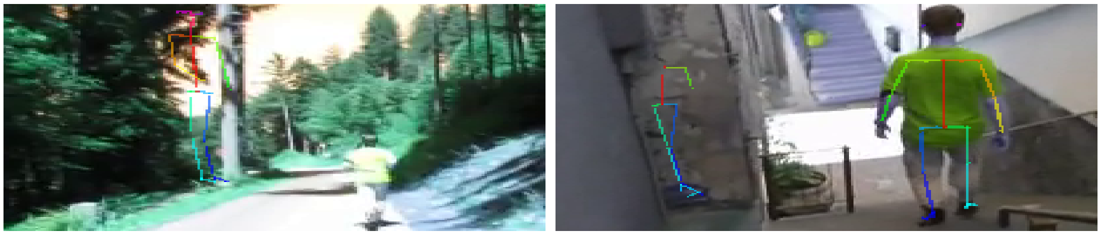


Figure 5.8: Typical fail cases of OpenPose in our video data preprocessing, with downscaled video frames, background objects are wrongly recognized as human.

required. In our experiments, it is calculated for 10-seconds video clips. If the human subject in the first video frame is not standing up straight, the scale factor computed using the method in Chapter 5.3.3 will be larger than the actual values.

5.4.3 Applications of CROMOSim in downstream Tasks

HAR Tasks

In this Chapter, we evaluate the utility of CROMOSim in data augmentation for training HAR models. Here we consider three settings: i) R2R, where models are both trained and tested with real IMU data; ii) V2R, where models are trained with simulated data but tested with real data; iii) Mix2R, where models are trained using a mixture of real and simulated data while tested with real data.

We adopt the DeepConvLSTM network proposed in [97] as the task model, while the same simulator neural network trained on the TotalCapture dataset is used here to simulate sensor readings from videos. Evaluations are made on the Realworld, PAMAP2 and Opportunity datasets respectively, with data simulated from the same video source (Realworld dataset). An ablation study was conducted by removing robust CVD from the proposed pipeline, and the resulting approach is called *CROMOSim Lite.* To make the result directly comparable, we followed the experiment

Table 5.3: Average F1 scores of random single-subject-hold out experiments on the RealWorld dataset. IMUTube* corresponds the scores reported in [1].

	R2R	V2R	Mix2R
IMUTube*	0.730±0.007	0.546±0.008	0.778±0.007
IMUTube	0.729±0.007	0.552±0.005	0.781±0.011
CROMOSim Lite	0.729±0.007	0.580±0.047	0.802 ±0.013
CROMOSim	0.729±0.007	**0.593±0.012**	**0.821±0.003**

Table 5.4: Random single subject hold out evaluation on PAMAP2 dataset with mean F1-score. IMUTube* corresponds to the scores reported in [1].

	R2R	V2R	Mix2R
IMUTube*	0.700±0.016	0.552±0.017	0.702±0.016
CROMOSim Lite	0.702±0.021	0.638±0.009	0.726±0.014
CROMOSim	0.702±0.021	**0.689±0.012**	**0.769±0.009**

Table 5.5: Random single subject hold out evaluation on Opportunity dataset with mean F1-score. IMUTube* corresponds to the scores reported in [1].

	R2R	V2R	Mix2R
IMUTube*	0.887±0.007	0.788±0.010	**0.884±0.007**
CROMOSim Lite	0.862±0.008	0.778±0.013	0.870±0.008
CROMOSim	0.862±0.008	**0.803±0.011**	0.879±0.008

protocol in IMUTube [1].

Table 5.3 reports the average F1 scores of five single-subject-hold out experiments on the RealWorld dataset. Since the authors of IMUTube provide their simulated data on this dataset, we directly replicated their experiments and the results are in the second row. For comparison purposes, we also include the scores reported in [1] as the first row. It can be seen the two are quite similar to one another. Even CROMOSim Lite outperforms IMUTube in V2R and Mix2R experiments, while CROMOSim works the best. Moreover, Mix2R achieves much higher F1 scores compared to R2R and V2R, demonstrating the utility of data augmentation with simulated data.

Table 5.4 and 5.5 summarize the results from CROMOSim and those reported in [1]. Due to the different sensor placements in the PAMAP2 and the Opportunity datasets, the simulated data provided by the authors of IMUTube cannot be used, so we take their reported performance here. Similar to the RealWorld dataset, CROMOSim outperforms IMUTube for the PAMAP2 datasets but with a more prominent margin; the HAR model trained from Mix2R is still superior to those from R2R and V2R. With the Opportunity data, however, the improvement of Mix2R over R2R is marginal while IMUTube* reports negative results for Mix2R. Although the Mix2R results are lower than those of IMUTube*, the difference is consistent with that for R2R. Therefore, one may consider the two perform comparably for this dataset. The reason for the small benefit of Mix2R in CROMOSim can be attributed to the small number of subjects in Opportunity. With a small number of training subjects, the DeepConvLSTM model does not generalize well to unseen subjects. Despite of the higher level of subject diversity in RealWorld, distribution mapping in IMUTube and CROMOSim in fact forces the distribution of simulated data to be close to the two subjects in the training set. Therefore, the benefit of data augmentation is diminished. As part of our future work, we will investigate domain adaptation approaches that retain diversity of subjects in simulated data while reducing the domain gap between simulated and real data. A possible solution is to train GAN models per pair of source and target subjects for unsupervised domain adaptation [154, 155].

HPE Tasks

Unlike HAR tasks that are essentially pattern recognition on sensory data, HPE aims to estimate the joint angles of a human body, and requires accurate IMU sensor readings. Therefore, in this Chapter, only MoCap simulated data is utilized.

Table 5.6: Knee angle estimation. Average RMSE and standard deviation are measured per each axis in degrees.

	X	Y	Z
R2R	15.4550±0.6217	8.3279±0.4751	3.1384±0.0403
V2R	20.8303±1.2644	**7.7459±0.3409**	3.4441±0.1727
Mix2R	**13.9236±0.5875**	8.2440±0.6053	**3.0355±0.2971**

We have previously designed a DeepBiLSTM network for knee joint estimation. It takes accelerometer and gyroscope readings from sensors on one's thigh and shank to predict 3D knee joint angles. In this set of experiments, We use Subject 1 – 3 in the TotalCapture dataset for HPE model training, and Subject 4's data for validation and real IMU data from Subject 5 for testing. Two sensors (virtual or real) are placed on proximal thigh (ProxTh) and right tibial (RTib) (see Fig 5.9). Similar to the HAR tasks, three DeepBiLSTM networks are trained using real data only, virtual data only and a mixture of virtual and real data. The size of real data samples from the three training subjects is around 143k, which is 39 minutes long. MoCap simulated data is on the same scale. In R2R and V2R we have 143k real or simulated data for model training, while in Mix2R the training data doubled by mixing the two.

Table 5.6 summarizes the average RMSEs and standard deviations of 3D knee joint angles in different settings. Note that the RMSEs should be put in the context of range of motions in the TotalCapture dataset, which are $[-11.5220, 152.4866]$, $[-44.3173, 41.3192]$ and $[-17.9953, 30.6022]$ around the x-, y- and z-axes.

From Table 5.6, we observe that in general Mix2R gives the most accurate estimations followed by R2R. Though the model trained on V2R has lower accuracy in the x-axis, its predictions are comparable to that from R2R in y-axis and z-axis. This phenomenon implies that MoCap generated virtual data using CROMOSim can

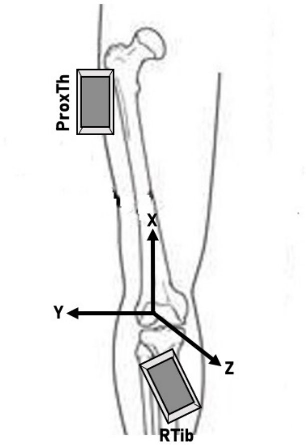


Figure 5.9: The sensor placement of knee angle estimation task. Real sensor readings are only available at ProxTh and RTib positions.

produce reasonable good HPE models. The observation is consistent with the high fidelity of MoCap simulated data in Chapter 5.4.2.

5.5 Conclusion

In this chapter, we implemented CROMOSim, a pipeline that simulates accelerometer and gyroscope readings at arbitrary user-designated on-body positions from MoCap and monocular RGB camera videos. A pair of DNN models are trained to learn the functional mapping between imperfect trajectory estimations in a 3D body tri-mesh to IMU data. Experiments showed that CROMOSim can generate higher fidelity data than baseline methods and is useful for downstream HAR and HPE tasks.

Chapter 6

Concluding Remarks

6.1 Conclusion

In this dissertation, to mitigate data scarcity in human motion analysis, we developed three solutions to tackle the problem from different aspects, namely, a data-efficient model that requires a small amount of labeled data from a target user, a new LNL method that can take advanrtage of a noisy crowdsourced data for model training and a cross-modality sensor simulator to synthesize IMU sensor data from other sensing modalities.

For adapting a deep learning model trained on multiple human subjects to an unseen one, we proposed a domain invariant feature learning framework based on a multi-task learning strategy and introduced a similarity metric to further reduce the amount of data required from the target domain. Experiments on three public and one in-house dataset demonstrated the superior performance of IFLF in a few-shot learning scenario, especially when the number of shots is 1 or 2.

Next, to address the unique challenges posed by learning with crowdsourced data, a novel invariant feature learning for wearable sensor-based HAR in the wild, VALERIAN is proposed. It consists of three components: self-supervised pre-training, invariant feature learning with noisy labels, and fast adaptation to new subjects. By training a multi-task model with separate task-specific layers for each subject, VALERIAN allows noisy labels to be dealt with individually for each subject while benefiting from shared feature representation across subjects. Experimental results show that VALERIAN significantly outperforms baseline approaches.

Finally, to take advantage of more abundant sources of human motion data, we designed CROMOSim and simulated IMU data from either MoCap or monocular RGB video data. It utilizes SMPL for 3D body pose and shape representations to enable simulation from arbitrary on-body positions. A DNN model is then trained to learn the functional mapping from imperfect trajectory estimations in a 3D SMPL body tri-mesh due to measurement noise, calibration errors, occlusion and other modelling artifacts, to IMU data. We evaluated the fidelity of CROMOSim simulated data and its utility in data augmentation on various HAR and HPE datasets.

6.2 Future Work

As part of our future work, we will continue to improve the usability and performance of the proposed frameworks, including IFLF, VALERIAN and CROMOSim. Specifically, we will work on the following aspects.

First, for the domain invariant feature learning, though only domain shifts due to human and device variations have been considered in this work, we believe it can also be applied to handle sensor placement diversity, which will be investigated as part of our future work. Another research topic is to explore the application of domain generalization (DG) in sensor-based HAR. With the knowledge acquired from source domains, a model with DG can be directly utilized on the target domain without any data from it.

Second, as part of future work, we are interested in developing a theoretical understanding of the aforementioned behaviour of LNL. Another venue of further efforts will be to build in-the-wild HAR datasets that can benefit research on this topic at large.

Lastly, in CROMOSim, as part of the future work, we are implementing a graphical user interface and wrapping up CROMOSim as an easy-to-use tool now. Hopefully, it will be open-sourced to the public by this summer. Other directions of further improvements include accelerating the video data processing, proposing a better domain adaption solution to bridge the gap between the distribution of simulated and real data, and experimenting CROMOSim with other data modalities as input such as millimetre wave radar.

Bibliography

[1] H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Ploetz, "Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–29, 2020.

[2] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.

[3] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE pervasive computing*, vol. 16, no. 4, pp. 62–74, 2017.

[4] K. Woodward, E. Kanjo, A. Oikonomou, and A. Chamberlain, "Labelsens: enabling real-time sensor data labelling at the point of collection using an artificial intelligence-based approach," *Personal and Ubiquitous Computing*, vol. 24, no. 5, pp. 709–722, 2020.

[5] S. Ramirez, X. Liu, P.-A. Lin, J. Suh, M. Pignatelli, R. L. Redondo, T. J. Ryan, and S. Tonegawa, "Creating a false memory in the hippocampus," *Science*, vol. 341, no. 6144, pp. 387–391, 2013.

[6] "Aging and health," https://www.who.int/news-room/fact-sheets/detail/ageing-and-health, accessed: 2022-06-20.

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

[8] 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*, 2019, pp. 85–96.

[9] J. Schmidhuber, "Evolutionary principles in self-referential learning," *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, vol. 1, p. 2, 1987.

[10] Y. Bengio, S. Bengio, and J. Cloutier, *Learning a synaptic learning rule.* Citeseer, 1990.

[11] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. Moura, "Few-shot human motion prediction via meta-learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 432–450.

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

[13] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, "Metasense: few-shot adaptation to untrained conditions in deep mobile sensing," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 110–123.

[14] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[15] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *International Conference on Learning Representations*, 2019.

[16] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," *Advances in neural information processing systems*, vol. 33, pp. 20 331–20 342, 2020.

[17] H. Ohashi, M. Al-Nasser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, and A. Dengel, "Augmenting wearable sensor data with physical constraint for dnn-based human-action recognition," in *ICML 2017 times series workshop*, 2017, pp. 6–11.

[18] X. Lin, Y. Chen, X.-W. Chang, X. Liu, and X. Wang, "Show: Smart handwriting on watches," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–23, 2018.

[19] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017, pp. 216–220.

[20] O. Steven Eyobu and D. S. Han, "Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network," *Sensors*, vol. 18, no. 9, p. 2892, 2018.

[21] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "Sensorygans: An effective generative adversarial framework for sensor-based human activity recognition," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[22] S. Zhang and N. Alshurafa, "Deep generative cross-modal on-body accelerometer data synthesis from videos," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 223–227.

[23] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[24] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5442–5451.

[25] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (imu) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[26] "Coriolis force," https://en.wikipedia.org/wiki/Coriolis_force, accessed: 2022-06-20.

[27] "Hall effect," https://en.wikipedia.org/wiki/Hall_effect, accessed: 2022-06-20.

[28] "Chapter 9 - magnetometer technology," in *Space Microsystems and Micro/nano Satellites*, ser. Micro and Nano Technologies, Z. You, Ed. Butterworth-Heinemann, 2018, pp. 341–360. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128126721000096

[29] M. Kok, J. Hol, and T. Schön, "Using inertial sensors for position and orientation estimation. arxiv 2017," *arXiv preprint arXiv:1704.06053*, 2017.

[30] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[31] J. C. Davis, S. Bryan, J. R. Best, L. C. Li, C. L. Hsu, C. Gomez, K. A. Vertes, and T. Liu-Ambrose, "Mobility predicts change in older adults' health-related quality of life: evidence from a vancouver falls prevention prospective cohort study," *Health and quality of life outcomes*, vol. 13, no. 1, pp. 1–10, 2015.

[32] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International conference on pervasive computing.* Springer, 2004, pp. 1–17.

[33] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 849–860, 2008.

[34] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *International workshop on human behavior understanding*. Springer, 2010, pp. 38–51.

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

[36] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[37] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[38] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *European Conference on Computer Vision (ECCV)*, sep 2018.

[39] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5253–5263.

[40] G. Cooper, I. Sheret, L. McMillian, K. Siliverdis, N. Sha, D. Hodgins, L. Kenney, and D. Howard, "Inertial sensor-based knee flexion/extension angle estimation," *Journal of biomechanics*, vol. 42, no. 16, pp. 2678–2685, 2009.

[41] T. Seel, J. Raisch, and T. Schauer, "Imu-based joint angle measurement for gait analysis," *Sensors*, vol. 14, no. 4, pp. 6891–6909, 2014.

[42] X. Yi, Y. Zhou, and F. Xu, "Transpose: real-time 3d human translation and pose estimation with six inertial sensors," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.

[43] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. P. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors." in *BMVC*, vol. 2, no. 5, 2017, pp. 1–13.

[44] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601–617.

[45] T. Von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll, "Sparse inertial poser: Automatic 3d human pose estimation from sparse imus," in *Computer graphics forum*, vol. 36, no. 2. Wiley Online Library, 2017, pp. 349–360.

[46] Z. Zhang, C. Wang, W. Qin, and W. Zeng, "Fusing wearable imus with multi-view images for human pose estimation: A geometric approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2200–2209.

[47] M. Zhang and A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036–1043.

[48] G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and smartwatch-based biometrics using activities of daily living," *IEEE Access*, vol. 7, pp. 133 190–133 202, 2019.

[49] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *2015 IEEE International conference on image processing (ICIP)*. IEEE, 2015, pp. 168–172.

[50] B. Bruno, F. Mastrogiovanni, and A. Sgorbissa, "Wearable inertial sensors: Applications, challenges, and public test benches," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 116–124, 2015.

[51] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *Biomedical engineering online*, vol. 14, no. 2, pp. 1–20, 2015.

[52] T. Sztyler, "Sensor-based human activity recognition: Overcoming issues in a real world setting," Ph.D. dissertation, Mannheim, Germany, 2019, http://ub-madoc.bib.uni-mannheim.de/49914/.

[53] Y. Vaizman, K. Ellis, and G. Lanckriet, “Recognizing detailed human context in the wild from smartphones and smartwatches,” *IEEE pervasive computing*, vol. 16, no. 4, pp. 62–74, 2017.

[54] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, “Guide to the carnegie mellon university multimodal activity (cmu-mmac) database,” 2009.

[55] S. Ghorbani, K. Mahdaviani, A. Thaler, K. Kording, D. J. Cook, G. Blohm, and N. F. Troje, “Movi: A large multipurpose motion and video dataset,” 2020.

[56] T. Von Marcard, G. Pons-Moll, and B. Rosenhahn, “Human pose estimation from video and imus,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1533–1547, 2016.

[57] “xsens motion capture system,” https://www.xsens.com, accessed: 2022-06-20.

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

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

[60] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha *et al.*, “Collecting complex activity datasets in highly rich networked sensor environments,” in *2010 Seventh*

international conference on networked sensing systems (INSS). IEEE, 2010, pp. 233–240.

[61] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

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

[63] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.

[64] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living.* Springer, 2012, pp. 216–223.

[65] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and kda-based svms," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 503291, 2014.

[66] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[67] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable

sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, pp. 1–16, 2018.

[68] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 351–360.

[69] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[70] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[71] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5715–5725.

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

[73] B. Wang, J. A. Mendez, M. Cai, and E. Eaton, "Transfer learning via minimizing the performance gap between domains," in *in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, p. 10644–10654.

[74] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8050–8058.

[75] A. Mathur, A. Isopoussu, F. Kawsar, N. Berthouze, and N. D. Lane, “Mic2mic: using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems,” in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, 2019, pp. 169–180.

[76] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, “Stratified transfer learning for cross-domain activity recognition,” in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2018, pp. 1–10.

[77] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[78] X. Sun, H. Kashima, R. Tomioka, N. Ueda, and P. Li, “A new multi-task learning method for personalized activity recognition,” in *2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 1218–1223.

[79] A. Saeed, T. Ozcelebi, and J. Lukkien, “Multi-task self-supervised learning for human activity detection,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[80] A. Kumar and H. Daume III, “Learning task grouping and overlap in multi-task learning,” *arXiv preprint arXiv:1206.6417*, 2012.

[81] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

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

[83] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services.* IEEE, 2014, pp. 197–205.

[84] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition.* Springer, 2015, pp. 84–92.

[85] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[86] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[87] "Metamotionr product description," accessed: 2020-04-01. [Online]. Available: https://mbientlab.com/metamotionr

[88] "Fitbit Versa product description," https://www.fitbit.com/us/products/smartwatches/versa, accessed: 2020-04-01.

[89] "MOX1 product description," https://www.accelerometry.eu/mox1/, accessed: 2020-04-01.

[90] "Actigraph product description," https://actigraphcorp.com/actigraph-wgt3x-bt/, accessed: 2020-04-01.

[91] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[92] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[93] L. Long, "Maml-pytorch implementation," https://github.com/dragen1860/MAML-Pytorch, 2018.

[94] Y. Bengio, "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization," *corr abs/1502.04390*, 2015.

[95] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[96] G. Nithin, M. Chhabra, Y. Hao, B. Wang, and R. Zheng, "Sensor-based human activity recognition for elderly in-patients with a luong self-attention network," in *2021 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2021, pp. 97–101.

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

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

[99] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” *Advances in neural information processing systems*, vol. 31, 2018.

[100] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, “A survey of label-noise representation learning: Past, present and future,” *arXiv preprint arXiv:2011.04406*, 2020.

[101] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, “A closer look at memorization in deep networks,” in *International conference on machine learning.* PMLR, 2017, pp. 233–242.

[102] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[103] L. Yi, S. Liu, Q. She, A. I. McLeod, and B. Wang, "On learning contrastive representations for learning with noisy labels," *arXiv preprint arXiv:2203.01785*, 2022.

[104] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1134–1143.

[105] S. Braun, D. Neil, and S.-C. Liu, "A curriculum learning method for improved noise robustness in automatic speech recognition," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 548–552.

[106] Y. Lyu and I. W. Tsang, "Curriculum loss: Robust learning and generalization against label corruption," *arXiv preprint arXiv:1905.10045*, 2019.

[107] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2304–2313.

[108] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 726–13 735.

[109] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021.

[110] Z. Zhang, H. Zhang, S. O. Arik, H. Lee, and T. Pfister, “Distilling effective supervision from severe label noise,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9294–9303.

[111] J. He, Q. Zhang, L. Wang, and L. Pei, “Weakly supervised human activity recognition from wearable sensors by recurrent attention learning,” *IEEE Sensors Journal*, vol. 19, no. 6, pp. 2287–2297, 2018.

[112] K. Wang, J. He, and L. Zhang, “Sequential weakly labeled multiactivity localization and recognition on wearable sensors using recurrent attention networks,” *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 355–364, 2021.

[113] ——, “Attention-based convolutional neural network for weakly labeled human activities’ recognition with wearable sensors,” *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7598–7604, 2019.

[114] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1038–1047.

[115] H. Dinkel, M. Wu, and K. Yu, “Towards duration robust weakly supervised sound event detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[116] S. Adavanne, H. Fayek, and V. Tourbabin, “Sound event classification and detection with weakly labeled data,” 2019.

[117] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4951–4958.

[118] F. Liu, J. Lu, B. Han, G. Niu, G. Zhang, and M. Sugiyama, "Butterfly: A panacea for all difficulties in wildly unsupervised domain adaptation," in *NeurIPS LTS Workshop*, 2019.

[119] X. Yu, T. Liu, M. Gong, K. Zhang, K. Batmanghelich, and D. Tao, "Label-noise robust domain adaptation," in *International Conference on Machine Learning.* PMLR, 2020, pp. 10 913–10 924.

[120] Y. Hao, R. Zheng, and B. Wang, "Invariant feature learning for sensor-based human activity recognition," *IEEE Transactions on Mobile Computing*, 2021.

[121] L. Jiang, D. Huang, M. Liu, and W. Yang, "Beyond synthetic noise: Deep learning on controlled noisy labels," in *International Conference on Machine Learning.* PMLR, 2020, pp. 4804–4815.

[122] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning.* PMLR, 2020, pp. 1597–1607.

[123] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2051–2060.

[124] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[125] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang, "Adversarial domain adaptation with domain mixup," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6502–6509.

[126] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[127] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[128] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7122–7131.

[129] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 459–468.

[130] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3383–3393.

[131] A. D. Young, M. J. Ling, and D. K. Arvind, "Imusim: A simulation environment for inertial sensing algorithm design and evaluation," in *Proceedings of*

the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks. IEEE, 2011, pp. 199–210.

[132] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[133] X. Li, J. Luo, and R. Younes, "Activitygan: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 249–254.

[134] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–15, 2018.

[135] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," in *International Conference on Spatial Data and Intelligence.* Springer, 2020, pp. 261–270.

[136] S. Takeda, T. Okita, P. Lago, and S. Inoue, "A multi-sensor setting activity recognition simulation tool," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1444–1448.

[137] Y. Liu, S. Zhang, and M. Gowda, “When video meets inertial sensors: Zero-shot domain adaptation for finger motion analytics with inertial sensors,” in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 2021, pp. 182–194.

[138] V. F. Rey, P. Hevesi, O. Kovalenko, and P. Lukowicz, “Let there be imu data: generating training data for wearable, motion sensor based activity recognition from monocular rgb videos,” in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 699–708.

[139] V. F. Rey, K. K. Garewal, and P. Lukowicz, “Yet it moves: Learning from generic motions to generate imu data from youtube videos,” *arXiv preprint arXiv:2011.11600*, 2020.

[140] V. Fortes Rey, K. K. Garewal, and P. Lukowicz, “Translating videos into synthetic training data for wearable sensor-based activity recognition systems using residual deep convolutional networks,” *Applied Sciences*, vol. 11, no. 7, p. 3094, 2021.

[141] H. Kwon, B. Wang, G. D. Abowd, and T. Plötz, “Approaching the real-world: Supporting activity recognition training with virtual imu data,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–32, 2021.

[142] J. Kopf, X. Rong, and J.-B. Huang, “Robust consistent video depth estimation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[143] J.-H. Mun, M. Jeon, and B.-G. Lee, "Unsupervised learning for depth, ego-motion, and optical flow estimation using coupled consistency conditions," *Sensors*, vol. 19, no. 11, p. 2459, 2019.

[144] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 71–1, 2020.

[145] "Optitrack system," https://optitrack.com/, accessed: 2022-07-09.

[146] "Vicon system," https://www.vicon.com/, accessed: 2022-07-09.

[147] "Mocap system," https://en.wikipedia.org/wiki/Motion_capture, accessed: 2022-07-09.

[148] S. J. Prince, *Computer vision: models, learning, and inference.* Cambridge University Press, 2012.

[149] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019.

[150] W. J. Conover and R. L. Iman, "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, vol. 35, no. 3, pp. 124–129, 1981.

[151] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2016, pp. 1–9.

[152] L. F. Richardson, “The approximate arithmetical solution by finite differences with an application to stresses in masonry dams,” *Philosophical Transactions of the Royal Society of America*, vol. 210, pp. 307–357, 1911.

[153] L. F. Richardson and J. A. Gaunt, “Viii. the deferred approach to the limit,” *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, vol. 226, no. 636-646, pp. 299–361, 1927.

[154] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International conference on machine learning.* PMLR, 2015, pp. 1180–1189.

[155] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.