Protective Behavior Detection in Chronic Pain Rehabilitation: From Data Preprocessing to Learning Model

Chongyang Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

UCL Interaction Centre

University College London

June 11, 2022

I, Chongyang Wang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Chronic pain (CP) rehabilitation extends beyond physiotherapist-directed clinical sessions and primarily functions in people's everyday lives. Unfortunately, self-directed rehabilitation is difficult because patients need to deal with both their pain and the mental barriers that pain imposes on routine functional activities. Physiotherapists adjust patients' exercise plans and advice in clinical sessions based on the amount of protective behavior (*i.e.*, a sign of anxiety about movement) displayed by the patient. The goal of such modifications is to assist patients in overcoming their fears and maintaining physical functioning. Unfortunately, physiotherapists' support is absent during self-directed rehabilitation or also called self-management that people conduct in their daily life.

To be effective, technology for chronic-pain self-management should be able to detect protective behavior to facilitate personalized support. Thereon, this thesis addresses the key challenges of ubiquitous automatic protective behavior detection (PBD). Our investigation takes advantage of an available dataset (EmoPain) containing movement and muscle activity data of healthy people and people with CP engaged in typical everyday activities. To begin, we examine the data augmentation methods and segmentation parameters using various vanilla neural networks in order to enable activity-independent PBD within pre-segmented activity instances. Second, by incorporating temporal and bodily attention mechanisms, we improve PBD performance and support theoretical/clinical understanding of protective behavior that the *attention* of a person with CP shifts between body parts perceived as risky during feared movements. Third, we use human activity recognition (HAR) to improve continuous PBD in data of various activity types. The approaches proposed above

are validated against the ground truth established by majority voting from expert annotators. Unfortunately, using such majority-voted ground truth causes information loss, whereas direct learning from all annotators is vulnerable to noise from disagreements. As the final study, we improve the learning from multiple annotators by leveraging the agreement information for regularization.

Impact Statement

The advancement of artificial intelligence creates new opportunities in the way healthcare can be offered to everyone. In particular, efforts have been undertaken in recent years to enable the provision of ubiquitous and tailored medical cares in a variety of scenarios. Of particular interest to this thesis is the advance in using wearable devices (*e.g.*, accelerometers, inertial measurement units, pressure sensors, and smartphones) to support remote physiotherapy for people with chronic diseases [1, 2], outpatient health deterioration monitoring [3], depression monitoring [4], and movement assessment for perinatal stroke screening [5]. The overarching goal of this growing body of work is to harness the power of ubiquitous technology for healthcare, therefore extending medical assistance to diverse out-of-hospital contexts, offering personalized support and therapy. Bearing this objective, this thesis dives into the first step for intelligent chronic pain (CP) rehabilitation, namely automatic protective behavior detection (PBD) across various functional activities.

Rehabilitation for people with CP is a significant societal challenge that affects 30.7% of adults in the United States [6] and 19% of the population in Europe [7]. Furthermore, people with CP may come with various conditions, *e.g.*, sports injury, post-stroke recovery, and cancer. Thus, a technology able to support the rehabilitation beyond the clinic can alleviate the burden on public healthcare system. Ubiquitous technology, with the capacity of functioning anywhere and anytime, is of great potential for such end. The role of ubiquitous technology here is to first capture people's movement with wearable devices, then automatically detect protective movement behavior, and finally respond with feedback, suggestion, and intervention that simulates the role of a physiotherapist.

In this thesis, we propose a series of studies inspired by the challenges that may arise in the real-world deployment of this technology, by using data collected from people with CP. Therein, we contribute not only to CP rehabilitation, but also to a larger domain dealing with movement data and healthcare. In terms of data pre-processing for using deep learning, the insights gained from our study provide a set of criteria for selecting possible ideal parameters for future datasets of PBD and possibly other emotional behavior detection tasks. Our attention-based approach produce competitive results in the relevant task of activity recognition, and is used to develop sonification software to assist physiotherapists in movement interpretation. Our proposed model integrating activity recognition and PBD demonstrates the significance of context recognition for emotional behavior detection, and the possibility of leveraging context recognition for healthcare personalization.

Based on the PBD capacity demonstrated in this thesis, we may inspire the human computer interaction community to propose novel apps or interaction paradigm like sonification to enrich the user-machine interaction for CP rehabilitation. In addition, our application of ubiquitous technology for CP rehabilitation may raise the interest of industry in developing more comfortable and affordable movement sensors or garments.

Acknowledgements

I would like to thank my fantastic supervisors, Nadia Bianchi-Berthouze (University College London), Nicholas D. Lane (University of Cambridge), and Amanda C. De C. Williams (University College London), who spent days and years on guiding me to pursue real research questions, write wonderful papers, and provided countless help during my career development. Your devotion to science, kindness for students, and thoughtful guidance for me are inspiring ever since the first day we met.

I thank University College London for awarding me the prestigious scholarship to aid my PhD, with which I was able to enjoy very much the life and study in this lovely foreign country without many worries. I wish my past work meet the expectation, and will continue contributing to the wellbeing of mankind as a PhD from UCL.

I am lucky to receive the guidance from my undergraduate mentor, Tong Chen (Southwest University) in China, who used his expertise and wisdom to shape my road to research. I also thank Hongying Meng (Brunel University London) for his rich advice and recommendation of me to Nadia that made my PhD application a great success. I am grateful to have received plentiful supports from Guangyuan Liu, Yan Zhang, and Min Peng in the past years.

I am more than happy to have worked with numerous brilliant people in the past years. I enjoyed very much the collaboration and discussion with Akhil Mathur, Temitayo A. Olugbade, Tao Bi, Amid Ayobi, Nicholas Gold, Albert Higgins, Roxana Ramirez Herrera, and Ahmed Alqaraawi at UCL, and Siyang Song, Joy Egede, Yuan Gao, Chenyou Fan, Junjie Hu, Su-Jing Wang, Xinwen Xu, Lingfeng Xu, Xintao Qiu at places across the world.

I dedicate this thesis to my girlfriend Erqiu, my parents, and grandparents, who have been accompanying me ever since the beginning. I am more than happy to have you by my side.

At this moment, my thought could not help but going back to the night I first met my supervisors in the interview hosted when I was in Beijing. This journey is wonderful in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chronic pain (CP) is a prevalent condition in 30.7% of adults in the United States [6] and 19% of the population in Europe [7]. People with chronic musculoskeletal pain (a prevalent type of CP) exhibit protective behavior (*e.g.*, guarding, stiffness, hesitation, use of support, and jerky motion) during physical activity [21], providing important information not only about their physical condition, but more specifically about their anxiety of movement, and ability to self-manage their condition [22, 23]. Unfortunately, this fear of movement impel many people with CP to avoid or minimize functional activity or the use of painful body parts, leading to further physical deterioration.

In clinical settings, physiotherapists respond to their patients' protective behavior with education/information, feedback, and exercise plan adaptation, with the aim to help them overcome fear [24]. This tailored support is important to reduce fears of injury from movement, incrementally build patients' self-efficacy, and maintain their engagement in physical activity [25, 26]. However, such support is expensive and only available to few people with CP. Furthermore, the behavior displayed in the clinic may not provide a clear understanding of the psychological barriers that patients meet in their daily life. As such, the support provided may not easily translate to self-management outside the hospital [27]. A particular issue in self-management is the limited awareness people with CP have of their way of responding to their anxiety toward movement [26]. Such lack of awareness makes it difficult for people with CP to apply strategies learned in the clinic to overcome their anxiety [24]. As

a result, people often disengage, thereby losing valued activities including family, work, and social involvement [23].

Ubiquitous sensing and computing technology offer new opportunities to provide such support to people with CP in their daily life. Patients describe technology capable of detecting protective behavior as a 'second pair of eyes', increasing their awareness and helping application of pain management strategies learned in the clinic [27]. In [25], patients and physiotherapists discussed how such technology could help better control activity pacing and breathing when protective behavior is detected. These are important for disrupting unhelpful movement habits and establish new and beneficial movement strategies. The technology may also, *e.g.*, replicate physiotherapists' advice on chair height if the patient lacks confidence in sitting down or standing up, and learn to help build confidence of the patient by gradually decreasing the height of the chair. These studies also show that awareness of habitual protective behavior can help reduce it (*e.g.*, reminding the person to bend the trunk as standing up from a chair), which is critical to facilitate functioning. Aside from providing personalized feedback, technology capable of sensing protective behavior can be adopted to evaluate clinical interventions' effects on people's daily life [28].

This thesis targets the first step in building a ubiquitous technology to support people with CP in their everyday management, which is to enable continuous protective behavior detection (PBD) during their various functional activities. This thesis investigates and proposes novel learning models and impactful data pre-processing approaches to enable continuous PBD across a set of basic activities (*e.g.*, stretching forward) that are feared by people with CP and that form the building blocks of a variety of more complex functional activities commonly conducted in everyday life (*e.g.*, one needs to stretch forward to clean the trunk of a car). To validate our approaches, we make use of an existing multimodal EmoPain dataset [12] gathered from people with CP and healthy people engaged in such basic activities. In this research, we also aim to derive and contribute to fundamental research questions that shall benefit a broader research area, with extra evaluation on relevant movement and medical datasets.

1.1 Research Questions, Challenges, Contributions

In this section, we plot the research map of this thesis by presenting the research questions that need to be addressed to develop automatic detection of protective behavior. Within each research question, we present the challenges from different aspects, and discuss how our work contributes to this research area.

1.1.1 Continuous PBD in Pre-Segmented Activity Instances

The first question this thesis investigates is if the use of deep learning could lead to activity-independent continuous PBD in pre-segmented instances. Since CP rehabilitation extends to everyday life, the PBD function needs to be activity independent since activities are not known a priori as in an exercise session. Even supposing that data could be pre-segmented per activity type, various challenges remain.

- The knowledge provided in previous PBD approaches is very limited, as interesting results were only achieved per overall activity instance and their feature engineering methods were not evaluated across different activity types [12, 29]. To tailor support for self-management, it is important to continuously track the occurrence of protective behavior to understand which (temporal) part of an activity is more feared. This calls for the use of deep learning methods able to acquire generalizable features from the movement data across different activity types, which also raises the need to search for the suitable parameters for sliding-window segmentation during continuous processing.

- While deep learning generally needs a large size of data for training, datasets in the context of movement-oriented medical analysis are generally small. This is particularly the case for CP physical rehabilitation. Indeed, it is widely acknowledged that collecting data from special groups, *e.g.*, people with CP, is challenging given the demand the condition imposes on the people (*e.g.*, pain, anxiety, depression). In addition, increasingly strict data protection regulations make it difficult to share data across research groups. Whilst we expect to see such datasets growing, it is important for us to think about how the available data for CP rehabilitation can be augmented to enable PBD with deep learning.

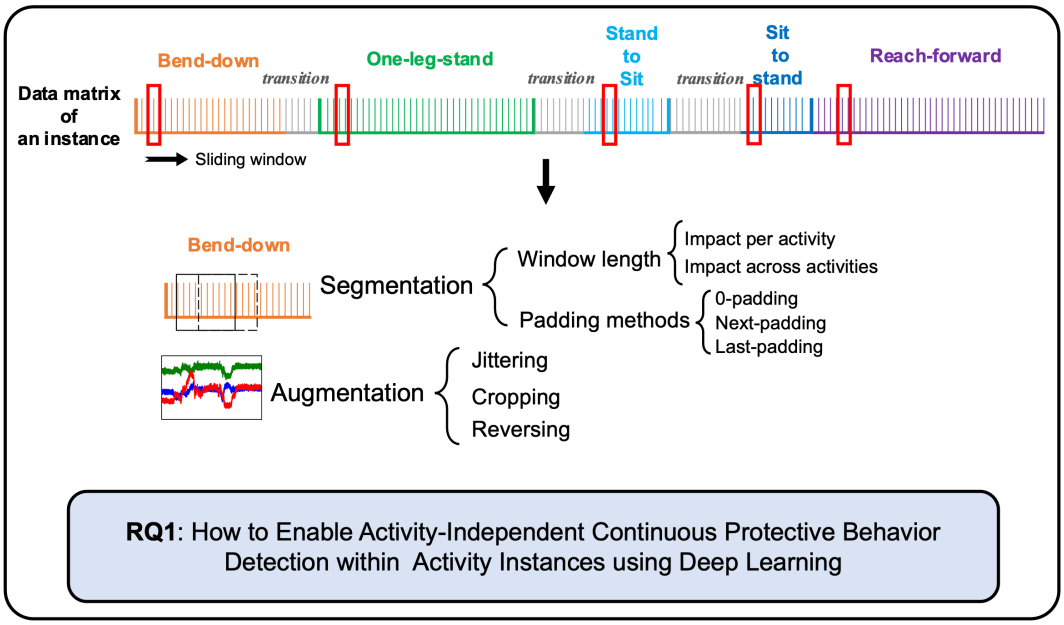


Figure 1.1: Chapter 4 studies different data preprocessing methods for raw data sequences of protective behavior across different activity types (data segmentation and augmentation), using various vanilla neural networks on data collected from real people with CP. The aim is to identify methods for activity-independent tracking. This work is published in ACM HEALTH and ISWC/Ubicomp'19. [8, 9].

Chapter 4 presents our study (as illustrated in Figure 1.1) that is preliminary but still the first to explore continuous PBD within each activity instance and across different activity types. First, we explore how to transform the raw data sequences into practical training and testing sets to aid model development. As the first study on this topic, our study covers different important issues in data preprocessing. Specifically, we explore what type of data augmentation is more effective to help train a robust model toward real-life use. We further study the impact of sliding-window length used in data segmentation to understand its relation with each activity type and with data comprising different activities. This is important to understand how parameter choices may be affected by other datasets of similar type.

The EmoPain dataset [12] that we use throughout this thesis comprises 18 CP and 12 healthy participants, the data collection and annotation of which took nearly a year to finish, according to the authors of the dataset. While a dataset of such size is not equivalent to the ones typically used in more popular deep learning research, it is important to start addressing critical questions in relation to PBD while

larger datasets are being created. As discussed in Chapter 4, the basic everyday types of movement present in this dataset together with the in-depth exploration and analysis of the above questions show great potential for applying our findings on preprocessing data for using deep learning on other relevant tasks and datasets.

1.1.2 Capturing Variety in Protective Behavior Detection

The second research question we explore is how to improve the PBD performance given the variety among people with CP in exhibiting protective behavior. This question is raised according to the following observations. The physical and mental capabilities of a person vary given different functional activities, *i.e.*, he/she may find it harder when standing up but easier during sitting down. In addition, for the same functional activity, different people may have varying capabilities in performing it. As a result, these varieties increase the difficulty for a model to detect protective behavior across subjects and activity types.

In Chapter 5, we investigate the use of attention-based deep learning architecture to improve the detection of protective behavior by capturing the most informative temporal and bodily cues characterizing specific movements and the strategies

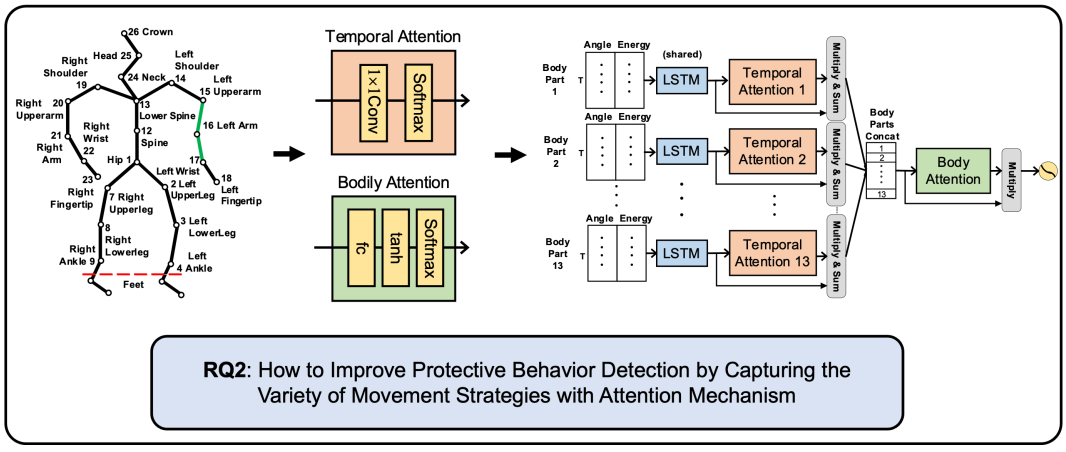


Figure 1.2: Chapter 5 proposes a novel model named BANet that combines the learning of temporal and bodily attention to improve the PBD performance by capturing the variety among people with CP in performing protective behavior. Informed by effects of protective behavior on movement, the analyses of the temporal and bodily attention scores reveal the larger variety of movement strategies and the continuous shift in attention paid to the feared body parts of people with CP. This work is published in a workshop at ACII'19 [10].

used to perform them (see Figure 1.2). We propose an end-to-end deep learning architecture named BodyAttentionNet (BANet). With two consecutive attention modules, BANet is able to learn temporal segments and body parts that are more informative to the detection of protective behavior. With such a data-driven self-attention mechanism, the approach operating on low-level features independently of the type of movement conducted appears to capture the variety of ways people execute a movement (including healthy people).

An analysis of the attention scores produced by BANet when fed with the testing data reflects the typical characteristic of protective behavior highlighted by previous pain literature [21, 23, 30, 31]. Therein, attention scores vary more significantly from one body part to another over time during the modeling process of PBD given data from people with CP than what in healthy people. Such patterns may relate to how each body part become the center of attention during specific phases of a movement, since either the body part is perceived in danger or it is used to avoid the use of parts in danger. Instead, the attention scores that emerge from the modeling of healthy people suggest more homogeneous values across body parts and across time. Finally, the proposed learning architecture is proved to work for activity type recognition, achieving very competitive if not better performances than previous state-of-the-art methods for activity recognition on benchmark datasets.

1.1.3 Continuous PBD in Sequence of Various Activities

The third research question targeted in this thesis is how to enable PBD across a continuous data sequence comprising different activity types without pre-segmenation (see Figure 1.3). Given a data sequence of activities and transitions between them, an ideal system should be able to perform behavior detection continuously. In order to do so, the following challenges are identified.

- In real life, activities and transitions between them are conducted in a continuous manner, hence the system, with the aim of providing real-time interaction with the user, has to continuously detect the behavior across such a sequence of activities and transitions. For this requirement, since the protective behavior changes according to the type of activity being performed, not knowing the type of activity makes

the detection more difficult. Although in Chapter 4 and Chapter 5 we show the possibility of PBD across different activities without knowing the activity type, it is seen that padding with the following samples beyond the current activity instance leads to reduction in performance. Additionally, in these two studies, the use of pre-segmentation already removed the extra influence of the transition activities.

- Furthermore, some behavior detection tasks are usually carried out in a simplex context, *e.g.*, with a fixed activity background. Under the rehabilitation scenario, a study presented in [32] explored the continuous detection of pain and anxiety in stroke patients during arm-based exercises. While this is a critical step in such direction, the patient was constrained to a sit-down position with only the arm engaged in the movement. In another study [33], the authors explored the emotion recognition during walking, which is also a simpler (stereotypical) problem to address. When the behavior is exhibited in a more complex context, *e.g.*, in varying activity types, these methods may easily fail to reach an acceptable performance, due to the inference of the changing activity background.

- Finally, people with CP usually tend to take relaxations to manage their pain

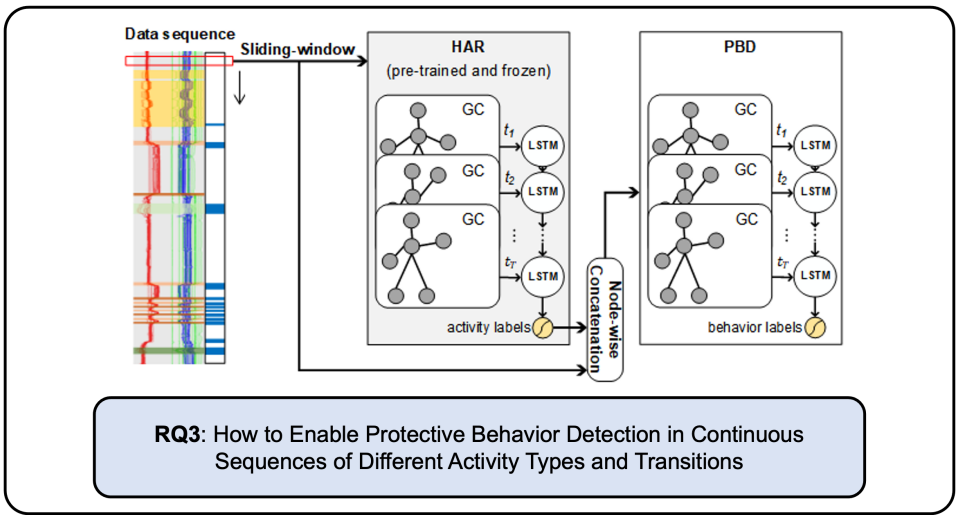


Figure 1.3: Chapter 6 investigates how to enable PBD across continuous data comprising different activity types without pre-segmentation, by leveraging activity recognition for contextualization. The backbone of the proposed model is a graph convolutional network, and the loss function is designed to counter class imbalances during training. This work is published in IMWUT/Ubicomp'21 [11].

during the transition between activities-of-interest. This leads to increased class imbalances, as not only more non-protective samples are introduced but also the movements irrelevant to activities-of-interest are added.

In short, one challenge is the inference of noisy context of varying activity types for behavior detection, and another challenge is the possible class imbalance existing in the continuous movement sequence caused by transition/irrelevant movements. As we start from activity-independent PBD, we now move to fully leveraging activity recognition to improve PBD in continuous data. Hence, in Chapter 6, we approach continuous PBD with continuous recognition of the activity (HAR) being performed. We propose a novel hierarchical HAR-PBD architecture (see Figure 1.3), where the activity type when recognized is continuously leveraged to build activity-informed input for concurrent PBD. Our experiments show that the activity-type information is important for PBD in continuous data, which leads to improved performance than the method that only alleviates the class imbalances that comes with the use of continuous data. This finding shall contribute to future studies working on movement behavior detection or affect recognition from body movements that the contextual information of the behavior or affective bodily cues, *e.g.* the type of activity being performed, is beneficial to improve the detection/recognition performance.

1.2 Thesis Structure

Chapter 2 provides the background on protective behavior, deep learning for movement-based tasks, and several machine learning works relevant to real-life challenges that could exist in the deployment of our method.

Chapter 3 presents the dataset that we mainly use for the evaluation of our methods, together with data preprocessing methods, vanilla neural networks, and validation methods and metrics that we use throughout the thesis.

Chapter 4 studies Research Question 1 with comprehensive experiments and analyses conducted to evaluate the impact of data preprocessing methods, including data segmentation and augmentation, on the PBD performance.

Chapter 5 works on Research Question 2 by proposing a novel attention-based

learning model to combine temporal and bodily attention mechanisms to improve the PBD performance; analysis of the attentional scores help us understand the various movement strategies adopted by people with CP; an extra evaluation of our method on the movement dataset for activity recognition demonstrates its generalizability.
Chapter 6 explores Research Question 3 with a novel hierarchical learning model that leverage activity recognition to improve PBD in continuous data; the use of graph convolutional network and a refined loss function designed for alleviating class imbalances also contribute to the improved performance.
Chapter 7 lays the conclusion, demonstrates the possible future use cases, reasons our limitations and shed light on future works.

1.3 Research Publications

Here, we list the peer-reviewed publications that originated from the studies described in this thesis. Publications that originated from my collaborations with other researchers beyond this thesis are also reported. * denotes equal contribution.

1.3.1 Publications from this Thesis

- **Chongyang Wang**, Temitayo A. Olugbade, Akhil Mathur, Amanda C. De C. Williams, Nicholas D. Lane, and Nadia Bianchi- Berthouze. "Recurrent Network Based Automatic Detection of Chronic Pain Protective Behavior using MoCap and sEMG Data." 23rd International Symposium on Wearable Computers (ISWC/Ubicomp'19), ACM, 2019. Oral presentation. Presented in Chapter 4.

- **Chongyang Wang**, Temitayo A. Olugbade, Akhil Mathur, Amanda C. De C. Williams, Nicholas D. Lane, and Nadia Bianchi- Berthouze. "Chronic-Pain Protective Behavior Detection with Deep Learning". ACM Transactions on Computing for Healthcare (ACM HEALTH), 2, 3, 2021. Presented in Chapter 4.

- **Chongyang Wang**, Peng, M., Olugbade, T. A., Lane, Nicholas. D., Williams, A. C. D. C., and Bianchi-Berthouze, Nadia. "Learning Bodily and Temporal Attention in Protective Movement Behavior Detection". 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW'19),

IEEE, 2019. Oral presentation. Presented in Chapter 5.

- Gold, N. E., **Chongyang Wang***, Temitayo Olugbade, N. Berthouze, and A. Williams. "P[l]aying Attention : Multi-Modal, Multi-Temporal Music Control". International Conference on New Interfaces for Musical Expression (NIME), 2020. Poster presentation. Presented in Chapter 5.

- **Chongyang Wang**, Yuan Gao, Akhil Mathur, Amanda C. De C. Williams, Nicholas D. Lane and Nadia Bianchi-Berthouze. "Leveraging Activity Recognition to Enable Protective Behavior Detection in Continuous Data". Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), 5, 2, 2021. Presented in Chapter 6.

1.3.2 Publications from Collaborations beyond this Thesis

Beyond the studies included in this thesis, I have also contributed to the following publications during my PhD in collaborations with other researchers.

- Min Peng, **Chongyang Wang***, and Tong Chen. "Attention Based Residual Network for Micro-Gesture Recognition". Proceedings of 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG'18), IEEE, 2018. Oral presentation.

- Min Peng, **Chongyang Wang***, Bi, Tao, Chen, Tong., and Zhou, X. "A Novel Apex-Time Network for Cross-Dataset Micro-Expression Recognition". 8th International Conference on Affective Computing and Intelligent Interaction (ACII'19), IEEE, 2019. Poster presentation.

- **Chongyang Wang**, Min Peng, Tao Bi, and Tong Chen. "Micro-Attention for Micro-Expression recognition". Neurocomputing, 410, 2020.

- Min Peng, **Chongyang Wang***, Yuan Gao, Shi Yu, and Xiangdong Zhou. "Multi-level Hierarchical Network with Multiscale Sampling for Video Question Answering". IJCAI, 2022.

1.3.3 Hosting Workshop and Challenge to Boost PBD Research

During my PhD, I also contributed to boosting the research in the area of pain-related state detection. I contributed to the hosting of a workshop [34] about recognition, treatment, and management of pain and distress (hosted at ACII'19), and was in charge of the website establishment and promotion. I was the data co-chair for the EmoPain Challenge [35] (hosted at FG'20) and in charged of the sub-challenge of protective behavior detection. Results of the challenge are published in the paper reported below (the order of authors is related to the order of the sub-challenges).

I also contributed as data co-chair for the EmoPain challenge 2021 [36], a sub-challenge of Affect Movement Recognition Challenge (hosted at ACII'21). These workshop and challenges held in the past three years have attracted more than 20 international research groups to participate, while the studies presented in this thesis are among the essential references for their works.

- Egede, Joy O., Siyang Song, Temitayo A. Olugbade, **Chongyang Wang***, C. De C. Amanda, Hongying Meng, Min Aung, Nicholas D. Lane, Michel Valstar, and Nadia Bianchi-Berthouze. "EmoPain challenge 2020: Multimodal pain evaluation from facial and bodily expressions". 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG'20), 2020. (Egede and Song chaired on pain recognition from facial expression, Olugbade chaired on pain recognition from movement, Wang chaired on protective behavior detection.)

Chapter 2

Background

In this chapter, we review the literature that plot our background, help highlight past limitations, and provide knowledge to inform the research presented in this thesis.

We first review works on protective behavior to understand what it is and what have been achieved so far for automatic Protective Behavior Detection (PBD). Secondly, we review the literature on deep learning for body movement behavior detection and human activity recognition to understand how it can be leveraged and advanced in the case of continuous PBD. Thirdly, we present studies from a broader research community to gain inspirations for the problems of sensor set optimization and using context recognition to improve the task-of-interest that exist given a more complex learning scenario. Finally, we summarize the key takeaways from the literature review and formulate the research questions that emerge.

2.1 Protective Behavior in Chronic Pain

Physical rehabilitation is an important part of the management of Chronic Pain (CP), where pain associated with dysfunctional changes in the nervous system persists and leads to reduced engagement in everyday physical activities despite the lack of injury or tissue damage [7, 37]. According to the fear-avoidance theory, reduced engagement with physical activity and use of maladaptive protective movement strategies [21], collectively referred to as 'pain behaviors', are often the result of fear of pain rather than injury or pain itself. This fear is mostly due to the association of movement with pain [23, 30].

Protective behaviors have been particularly highlighted as observable pain behaviors that can provide insight into the psychological capability of a person to manage their condition, and hence inform intervention [21, 31, 38]. It is correlated with self-reported pain and fear-related beliefs [21, 39] but its correlation with pain is often mediated by anxiety rather than directly explained by pain levels [19]. Further, unlike facial and vocal expressions which primarily act as tools in communicating to an empathetic audience, protective behavior has been found to have a primary role to protect (from which it derives its name) from perceived danger of injury or of increase in pain and so is more reflective of perceived physical demand [31].

Here, we first review the background study on protective behavior in the pain literature, and then previous works on its automatic detection.

2.1.1 Protective Behavior in Chronic Pain Literature

The popular method to systematically analyze protective behavior was proposed in [21]. Using trained observers to manually label videos of patients performing specific movements [21, 31], they showed that specific protective behaviors were exhibited by people with lower-back CP and that such analysis is critical to understand how well a person with CP is coping with their condition and is able to engage with their everyday life. Table 2.1 provides a detailed description of the identified protective behaviors that used in this thesis according to the previous studies [21, 12, 40].

We focus on the behavior in 5 categories: guarding/stiffness, hesitation, use of support/bracing, abrupt motion, and rubbing/stimulation. Particularly, the first two categories refer mainly to alteration in the movement dynamics and trajectory,

Table 2.1: The Five Categories and Definitions of Protective Behavior used in this Thesis

Category	Definition
Guarding/Stiffness	Stiff, interrupted or rigid movement.
Hesitation	Stopping part way through a continuous movement with the movement appearing broken into stages.
Support/Bracing	Position in which a limb supports and maintains an abnormal distribution of weight during a movement which could be done without support.
Abrupt Motion	Any sudden movement extraneous to be intended motion; not a pause as in hesitation.
Rubbing/Stimulation	Massaging touching an affected body part with another body part, or shaking hands or legs.

suggesting that the movement is broken into stages as a way to better control the movement and protect oneself. The third category indicates the use of support (either with objects such as a chair or other body parts of oneself) to avoid or alleviate the engagement of the painful part. The last category is referring to movements that may aim to relieve pain.

The way these protective behaviors are instantiated depends on the type of activity performed (*e.g.*, sitting down vs. reaching forward to the car roof) and what parts of the body the person perceives as reliable to protect the painful ones [12, 19]. They also depend on the environment, *i.e.*, sitting on a hard chair vs. a lower soft sofa where the use of support strategies may be less helpful. This results in a variety of protective movement strategies that people with CP use to cope with their condition and the environment as they engage in physical exercises or daily functioning.

Unfortunately, expert visual assessment is expensive and impractical given the prevalence of CP [41, 42], limiting observation to clinical settings, where a patient's behavior does not often reflect the person's abilities (or struggles) to move and manage during the more complex everyday environment [43]. As such, the need to better understand such behavior in real-life has raised the necessity to use technology as a way to monitor such behavior and provide the necessary advice [44, 45].

However, the approaches used in the past have been limited to analyzing coarse behavior, such as studying how far and where a person moves with respect to their home using Fitbit and GPS-based technology [26]. The findings from these studies showed limited correlations with key affective variables that characterize the ability of the person to self-manage their conditions. Moreover, it is not the quantity of activity that matters, but the quality and the type of activity (or aspect of activity) that are avoided (*e.g.*, the tendency to avoid bending the trunk when sitting down) that provide insights on the ability of the person to cope with and manage their condition [24]. Such avoidance behavior can lead on to further debilitation and increase in pain as the person loses physical capabilities and normal muscle strength and efficiency.

In addition, as physical rehabilitation in chronic conditions transitions from clinician-directed into self-managed (in the form of self-managed activities or func-

tional tasks such as doing housework as a way to exercise [26]), visual inspection in situ becomes unfeasible. At the same time, self-report of pain behaviors [43] in everyday functioning is unreliable, as people with CP may not be conscious of their responses to pain or feared situations [26, 27]. More importantly, self-report does not allow for fine-grained assessment of adopted movement strategies, which are necessary for insights into subjective experiences [21, 46] and for adjusting exercise plans or other forms of feedback (*e.g.*, just-in-time reminders to breathe deeply to reduce tension during the feared part of a movement).

Despite the limitation, the systematic analysis of movement proposed in the above pain literature suggests that protective pain behavior could be automatically tracked, and such capability could be embedded in ubiquitous rehabilitation technology to enable a more personalized and on-the-movement support to people with CP during their everyday life.

2.1.2 Automatic Analysis of Pain Behavior

The use of body movement as a modality for automatic pain-related detection has been largely ignored, even though bodily behavior such as protective behavior is possibly more pertinent to pain experiences than facial or vocal expressions [31]. The relevance of the body, as an affective modality in general, is in its indication of action tendency [47], which in the case of pain is to protect against perceived harm or injury [31]. It has been indeed suggested that the body [47] is an effective modality for automatic detection of affect, although most of the work in this area has been focused on the so-called basic affective states like happiness, sadness, anger, surprise, and scare [48], and on some sport related emotional states [49, 50].

More recently, researchers have also started to explore the importance of body expressions in the diagnosis and management of various medical conditions such as autism [51], depression [52], stroke [5, 1], and more relevant to this thesis, CP [31]. The interest on modeling body expressions is increasing thanks to the availability of increasingly robust vision-based skeleton tracking software and also low-cost wearable motion capture technology. Here, we review the research on automatic detection of body expressions in relation to pain behavior. A summary of the works

Table 2.2: Summary of past works before this thesis on pain-related recognition tasks.

Study	Dataset	Task	Method	Result
[53]	EMG data from the right and left vertebral muscles	Analyze the differences between people with CP and healthy people during static and dynamic postures	Two-group discriminant function analysis with features like mean bilateral level and lumbar curvature etc.	Statistically significant differences are found between the two groups of people in muscle activity
[39]	sEMG data from the lumbar paraspinal muscles	Analysis of factors related to muscle relaxation status of people with CP during exercises	Statistical analysis (linear regression analysis etc.) based on flexion relaxation ratio	EMG data could provide evidence to the appearance of protective behavior
[54]	3D coordinate data of head and torso	Classification of neck movement patterns related to Whiplash-associated disorders	1-layer neural network with principal component analysis	Accuracy of 0.89
[55]	kinematic data for spinal movement	Classification of self-reported pain levels	3-layer neural network	Prediction share high relevance with reported pain level ($R^2 = 0.997$)
[56]	EmoPain dataset	Classification of three pain levels in the activity of reaching forward	SVM with movement and muscle activity descriptors	F1 of 0.63 and 0.69 for the classification with movement and sEMG data, respectively, and 0.8 for combined input
[57]	EmoPain dataset	Classification of three pain levels in the activities of bending and sit-to-stand	SVM with movement feature optimization	Accuracy of 94% and 80% for the classification in bending and sit-to-stand respectively
[19]	EmoPain dataset Ubi-EmoPain dataset	Classification of Movement-related self-efficacy in the activities of reach-forward and sit-to-stand	SVM with movement features	Mean F1 of 0.95 and 0.78 for the classification in reach-forward and sit-to-stand respectively, and 0.79 for Ubi-EmoPain dataset
[38]	EmoPain dataset Ubi-EmoPain dataset	Classification of two distress levels and three pain levels in the activities of bending and sit-to-stand	SVM and RF with movement features	Mean F1 of 0.88 (0.67) and 0.83 (0.85) and 0.86 (0.81) for the stress classification in bend, reach-forward and sit-to-stand respectively, and 0.85, 0.84 for pain level classification in bend and sit-to-stand, for EmoPain (Ubi-EmoPain) dataset
[29]	EmoPain dataset	Classification of guarding behavior in the activities of sit-to-stand and one-leg-stand	RF with posture and velocity features	Mean F1 of 0.81 and 0.73 for the classification in sit-to-stand and one-leg-stand respectively

is shown in Table 2.2.

The majority of the work done in relation to automatic detection of pain behavior have been on automatic differentiation of people with CP from healthy control participants using movement and electromyography (EMG) data, and recognition of anxiety and pain levels of the people with CP.

In an earlier study [53], researchers used a set of features computed from the EMG data like Mean Bilateral Level (mean of the average scores of the EMG data collected from right and left vertebral muscles) and Right/Left difference (difference of such average scores) to analyze the differences between people with CP and healthy controls during both static and dynamic postures. Obvious differences in those EMG features between the two groups of people were only found during dy-

namic postures (trunk flexion and rotation), especially when the range of movement was high, which suggest an altered muscle tension in people with CP during specific dynamic postures.

A later study [39] used surface electromyography (sEMG) to explore the dynamic activity of the lumbar paraspinal muscles in people with CP during exercises. They used a movement feature called Flexion Relaxion Ratio (FRR) to represent the Flexion Relaxation Phenomenon (FRP), which is a typical indicator for a reduction or silence of myoelectric activity of the lumbar erector spine muscle during full trunk flexion seen in normal people. Their analysis has shown that, independently of the range of motion and pain level, the FRP measured by FRR disappeared or was limited due to the fear of injury and low self-efficacy beliefs.

The above studies show that EMG can be an informative signal representing the appearance of psychological responses to feared movement (which lead to protective behavior in many situations) in people with CP during activities. In other words, by looking at the movement of people with CP and together with the EMG data collected from the sensitive muscle groups of the back, it could be feasible to detect events of protective behavior.

A more recent study [54] used an artificial neural network with one hidden layer to help the diagnosis of Whiplash-associated disorders (WAD) based on neck movement. They calculated the rotation angle and angle velocity given the 3-dimensional coordinate data collected with 6 markers attached to the head (3 markers) and torso (3 markers), then performed Principal Component Analysis (PCA) to reduce the dimensionality as well as to improve the performance of the shallow network. Through an experiment on 59 WAD subjects and 56 control subjects, the result were very promising, with an accuracy of 0.89. Although the scenario of neck movement is much simpler than studies working on full-body movement during everyday activity, the study demonstrated the possibility of using an artificial neural network and the angular movement features of the affected body part to detect the movement patterns triggered by some medical conditions.

Studies reported in [55, 56, 57, 38] further discriminated levels of self-reported

pain in people with lower-back CP. The kinematic 3D coordinate data collected from the markers attached to the inter-pedicle screws placed at right and left L4 (or L5) and S1 segments for the spinal movement was used in the first study, and full-body motion data and sEMG data collected from four sections of the back were used in the later three studies. A common finding in these studies is that the way a person with CP handles a painful anatomical body segment provides information about their subjective pain experiences, such as the flexion range of the lumbar spine represents the confidence of the subject during activity like forward reaching.

A more directly relevant study to the one discussed in this thesis is presented in [12]. Rather than detecting the level of pain, the study aimed to automatically detect the presence or absence of protective behavior. Like in [19, 56, 57, 38]), this study used the EmoPain dataset presented in the same paper. This dataset comprises full-body movement and sEMG data recorded while people with CP (and healthy control participants) performed 6 physical activities typically challenging for this cohort. Figure 2.1 show some image samples of a participant performing a reaching forward movement during the data collection of the EmoPain dataset. A more detailed description of the EmoPain dataset can be found in Chapter 3. The authors extracted a single feature vector including the range of angles for 13 full-body joint angles, the mean energy for these angles, and the mean sEMG recorded bilaterally in the lower and upper back muscles for each exercise instance. These feature vectors were used to predict the mean (across 4 raters: two physiotherapists and two clinical psychologists) of the proportion of the instance that had been binarily labelled as guarding based on Random Forests (RF). For all the activity types, they obtained mean squared error between 0.019 and 0.034 (average = 0.027, standard deviation

Figure 2.1: Image samples from the EmoPain dataset of a participant doing reaching forward. The sensors used are Inertial Measurement Units (IMUs) and surface Electromyography (sEMG) sensors. (Taken from [12])

= 0.005), and Pearson's correlation was between 0.16 and 0.71 (average = 0.44, standard deviation = 0.16). The low correlation ratio despite low error rate suggests that the predicted values are not very consistent with the ground truth across the different activity types. Previous classification of a subset of these data employed the same feature extraction strategy and RF method, but turned to detecting the existence of protective behavior per overall activity instance and only focused on two of the activities (sitting to standing and standing on one leg), achieving F1-scores of 0.81 and 0.73 respectively [29].

From another perspective, [19] investigated movement behaviors (*e.g.,* guarding and hesitation) that clinicians use in judging pain-related self-efficacy and showed the feasibility of automatic detection based on these cues. Interestingly, guarding behavior (a major type of protective behavior) was one of the cues of low self-efficacy specified by clinicians.

[19] further provide evidence that low-cost body sensing technology can enable the detection of pain related experiences in functional activities (beyond just exercises). In their first experiment with data collected using a full-body motion capture suit comprising 18 IMUs (MetaMotion IGS-190) and 4 sEMG sensors, they explored the impact of features computed from different body parts for pain intensity recognition. These features were designed based on the visual inspection of different types of activity and from the physiotherapy report, *e.g.* the range of trunk flexion was extracted as a feature for bending and the knee and pelvic angles at the point of lift was extracted for siting-to-standing. They managed to only use 4 IMUs sensors (SparkFun MPU9150) attached to the head, trunk, right upper and lower leg and 2 sEMG sensors (BITalino) attached to the right upper arm and trunk for pain-level recognition. The experiment result of 0.79 for F1-score proved that a smaller low-cost sensor set can also be applied for pain intensity analysis during functional activity.

As we can see, for modeling the pain-related status and movement behavior, the above feature engineering methods that aimed to conduct the task per activity type are quite straightforward in methodology, *i.e.*, using posture and velocity-

based features together with shallow classifiers like SVM and RF, but also are not discriminative enough for the task of PBD as the promising results are only achieved on the modeling per each overall activity instance separately and only for few of the activity types.

Unfortunately, at the time of the studies carried out in this thesis, the use of deep learning approach on pain-related tasks had focused on facial behavior only. This raises the question if deep learning could lead to better performances in the detection of protective behavior than the one obtained with traditional machine learning methods. In addition, it would be also important to understand if the advanced learning techniques could lead to activity independent PBD so that it could be used in everyday support to people. To this purpose, in the next section, we review the literature on deep learning in (non-affect related) body movement-based tasks to gain some inspirations.

2.2 Deep Learning for Body Movement Analysis

To better understand how to improve the recognition of protective behavior, we review here the previous state-of-the-art deep learning approaches used for movement-related tasks. In particular, we first review the literature on using vanilla deep learning for Human Activity Recognition (HAR) with wearable sensors. Secondly, we shift the focus to more relevant studies from the HAR community on abnormal movement behavior detection. Table 2.3 summarizes the used datasets, data preprocessing parameters, validation methods, models, and results of these studies.

2.2.1 Deep Learning for Human Activity Recognition

Deep learning is the leading approach in many very challenging tasks such as object detection, video understanding, and speech recognition, with increasing use toward applications in healthcare domain [64].

The core merit of modern neural networks, where the use of traditional fully connected layers is drastically reduced, is the ability to learn from large sets of high dimensional (and low-level representation of) data [64]. In comparison to traditional feature engineering methods, the generalization ability of a deep neural network is

Table 2.3: Summary of past works exploring vanilla deep learning methods for wearable human activity recognition and abnormal behavior detection.

Study	Dataset	Preprocessing	Validation	Model	Results on the Respective Dataset (F-measure)
[58]	Opp (10 low-level activities, 1 accelerometer) Skoda (10 activities, 1 accelerometer) Actitracker (6 activities, cellphone)	Fixed window length of 64 timesteps and 50% overlapping.	Hold-out validation	CNN with partial sharing	Acc: 76.83% 88.19% 96.88%
[59]	Opp (18 mid-level activities, accelerometers and IMUs) Hand Gesture (12 gestures, 1 accelerometer and 1 gyoscope)	Fixed window length of 1s and sliding step of 3 timesteps.	Hold-out validation	CNN with temporal convolution	Acc: 0.56 0.90
[13]	Opp (18 mid-level activities, accelerometers and IMUs) PAMAP2 (12 lifestyle activities, IMUs, temperature, heart rate) DG (gait, 3 accelerometers)	Sliding window segmentation with 1s length, 50% overlapping 5.12s length, 78% overlapping 1s length, 50% overlapping	Hold-out validation	DNN 1 layer CNN b-LSTM LSTM	0.904, 0.575, 0.633 0.937, 0.591, 0.684 0.868, 0.745, 0.741 0.882, 0.698, 0.76
[60]	Opp (18 mid-level activities, accelerometers and IMUs) PAMAP2 (12 lifestyle activities, IMUs, temperature, heart rate) Skoda (10 activities, 10 accelerometers)	Bag-wise training with different window sizes and testing per sample	Hold-out validation	Ensemble of 2-layer LSTM	0.73 0.85 0.92
[14]	Opp (18 mid-level activities, 5 IMUs) Skoda (10 activities, 20 accelerometers)	Non-overlapping sliding window with length of 9.7s	Hold-out validation	3 layers CNN with LSTM layer	Transfer from Opp to Skoda 0.85 Transfer from Skoda to Opp 0.25
[61]	DG (gait, 3 accelerometers)	Fixed window length of 58s. Augmentation with Rotation and Permutation	5-fold cross validation	9 layers CNN	86.76% (Acc)
[62]	SMMs (Behavior detection during seating and 3 classroom activities, 3 accelerometers)	Sliding window with 1s length and 87% overlapping	Leave one subject out	3 layers CNN	0.74 on lab data 0.5 on classroom data
[63]	SMMs (Behavior detection during seating and 3 classroom activities, 3 accelerometers)	Sliding window with 1s length and 87% overlapping	Leave one subject out	3 layers CNN with transfer learning 3 layers CNN plus LSTM layer Ensemble of LSTM	0.78 on lab data 0.56 on classroom data 0.75 on lab data 0.48 on classroom data 0.77 on lab data 0.53 on classroom data

better as high-level representation and data semantics could be automatically learned from the raw data or low-level representations under the guidance of the task, which usually does not require prior knowledge about the type or characteristic of the input data (*e.g.,* the activity type that usually used for the feature engineering in pain-related movement tasks as seen in the last section).

Similar to the other domains listed above, advances in HAR have also benefited from the exploration of deep learning. Particularly, the successful application of deep learning in HAR was enabled by an effort in the HAR community to create a series of benchmark datasets collected with inertial sensors: *e.g.*, the Opportunity dataset (Opp) [65], the PAMAP2 dataset [66], and Skoda dataset [67]. The aim of these datasets is to foster research on recognizing the type of movement, and activity like functional activities, rather than evaluating their qualities such as detecting protective behavior or other affect-influenced movement behaviors [68]. Still, as the data share very similar structure, the knowledge in HAR studies using movement data can be leveraged for our work on PBD. An illustrative diagram summarizing these works is shown in Figure 2.2.

In the following, we review in detail some impactful wearable HAR studies proposed in the past that are among the first to explore the use of vanilla neural

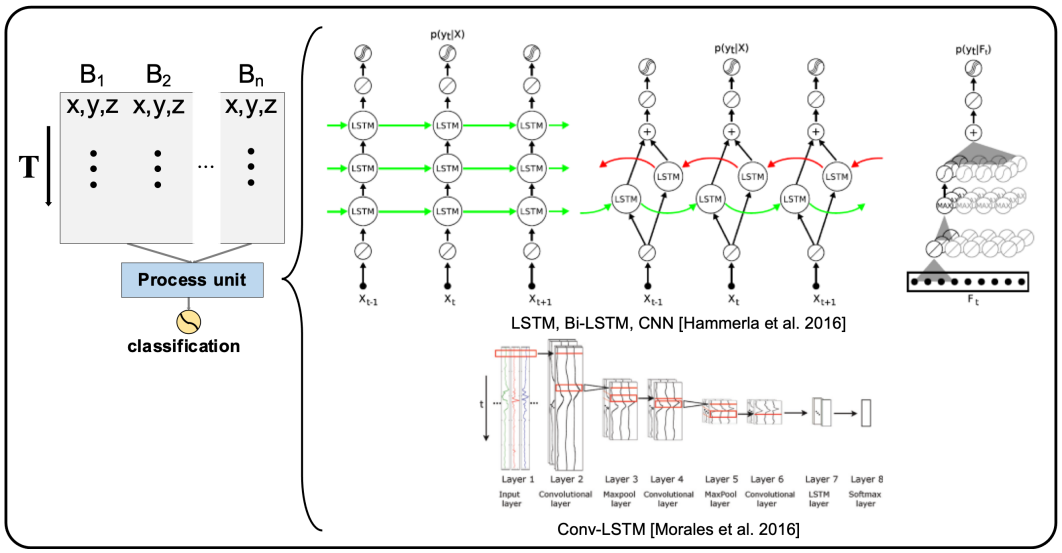


Figure 2.2: Earlier studies proposed for HAR using deep learning treats the movement data collected from different positions as a data matrix, with vanilla neural networks like CNN, LSTM applied directly on it. (partially taken from [13, 14])

networks to gain a basic idea about how deep learning is adapted in this scenario. It should be noted that, although data for PBD comprise full-body coordinates transferred from the raw signal collected from 18 IMUs, other HAR studies using full-body skeleton (coordinates) data that usually collected from visual sources (*e.g.*, the NTU RGB+D [69] and Kinetics [70] datasets) would be reviewed in later study chapters, as at the beginning of our research we do not use the full-body coordinate data for modeling.

Based on the subsets of Opp (10 low-level activities plus 1 null activity with one sensor on the right arm) and Skoda (10 activities related to the right arm with one accelerometer on the right arm) datasets, [58] adopted a CNN network with a partial weight sharing strategy designed for temporal signal processing to take the temporal signal frames of 64 timesteps at X, Y, and Z axis as input separately. The network achieved classification accuracies of 88.19% and 76.83% on the two datasets, respectively. The study in [59] extended the capacity of CNN to adopt a temporal convolution to process data collected from accelerometer as well as other sensor types like gyroscope and magnetometer. For the 18 mid-level activities of the Opp dataset, their method achieved an average F1 score of 0.56.

A study seen in [14] used a stack of two convolutional layers followed by max

pooling, one (more) convolutional layer, long-short term memory (LSTM) layer, and dense (with softmax activation) layer for wearable HAR. They further applied a transfer learning strategy by training with the Opp dataset, although without the feet data, to classify the activities in the Skoda dataset based on data collected from the arms. Interestingly, the authors obtained mean F1-scores of 0.40 and 0.85 on the source and target datasets, respectively. When the Skoda dataset was used to train the network instead, the mean F1-score achieved for the Opportunity dataset was only 0.25. As they reasoned in the end, the performance of such transfer learning-based method depends a lot on the correlation between the source and target datasets. Therefore, in their case, when the source dataset comprising activities of richer variability (Opp) is used, the performance on the target dataset with lower variability (Skoda) is largely improved.

To compare the performance of networks comprising convolutional or LSTM layers, [13] used a pure LSTM network comprising three bidirectional LSTM layers (Bi-LSTM), a convolutional network that contains one convolutional layer with max pooling and one fully-connected layer, and the convolutional LSTM network (Conv-LSTM) proposed in the study above to classify 18 types of everyday physical activities in the Opp dataset. The best result was obtained using the Bi-LSTM network with mean F1-scores of 0.745 (compared with 0.591 and 0.704 using CNN and Conv-LSTM) using hold-out validation, based on movement data recorded from the upper limbs, feet, and trunk of 4 participants in the dataset. In this study, data samples were windows of lengths of 1 and 5.12 second (s), with overlapping ratio of 50% and 78% respectively, segmented from activity instances.

Studies presented above show that the recurrent networks using LSTM units or the network comprising recurrent LSTM layers show better performance in dealing with sequential movement data than pure CNNs. Meanwhile, a sliding window segmentation method was applied to extract consecutive frames from the raw movement sequence to make input of the model. However, the segmentation length, an impactful parameter as reasoned in [18], was not further studied in their works to see how it may affect the performance on activity recognition. While we

are inspired by their success in applying vanilla deep learning models on HAR tasks, our research presented in Chapter 4 aim to further study the parameters (*e.g.,* sliding window length) used in data preprocessing to understand how they may impact the model performance and what factors should be considered to find a suitable parameter set for future relevant datasets.

The study in [60] achieved mean F1 scores of 0.73 and 0.85, based on hold-out validation, respectively on the Opp and PAMAP2 datasets using an ensemble of two-layer LSTM networks with dropouts added after each layer and a dynamic windowing approach. This method further led to a mean F1 score of 0.92 on the Skoda dataset for car manufacturing activity. Instead of using a fixed sliding window segmentation, and also in order to avoid the selection of window length, they performed a bag-wise training where random sizes of windows were used.

As we can learn from these works, aside from better performances gained with deep learning and particularly LSTM-involved networks, studies on PBD could be conducted to also evaluate the segmentation procedure as it remained unknown how the segmentation parameter may impact the model performance and what factors should be considered to find a suitable set given a new dataset in the future. Specifically, if we use a fixed-length window segmentation, a further study on how the window lengths may affect performance on PBD within each activity type or across different types shall be studied. This is particularly important as we want to derive knowledge from our PBD study on the existing dataset to future works using datasets having the same or relevant tasks about affect-influenced movement behavior detection. Furthermore, as can be inferred from [2] that the discussion on choosing window lengths may get avoided if one conduct training with frames of various lengths and make prediction per timestep, we also evaluate this approach in our experiments to verify its effectiveness when the learning scenario is transferred from activity recognition to movement behavior detection.

2.2.2 Deep Learning for Abnormal Behavior Detection

Studies on activity recognition reported above provided insights on the comparison of different vanilla network structures in dealing with wearable data sequences.

Instead of only recognizing the type of activities, it is interesting for our work to understand how such network architectures can be used for specific altered or anomalous behavior detection.

Aside from the analysis of protective behavior in CP that we discussed in Section 2.1, here we present two datasets that have been released to the ubiquitous computing community before this work that led to relevant deep learning studies on the analysis of anomalous movement behaviors.

First is the Daphnet Gait (DG) dataset [71] that used for the detection of freezing gait behavior in people with Parkinson's disease (PD) during walking. The data were collected from ten idiopathic PD patients using accelerometer and gyroscope attached to the ankle, knee, and trunk. This task is more straightforward than PBD that involved in a variety of functional activities: i) freezing gait behavior is marked by a clear and brisk interruption of walking; ii) although the dataset contains a variety of walking scenarios, *e.g.,* walking in a straight line, random walking in a room, and walking to fulfill daily tasks (entering a room, and getting something to drink), the activity of interest generally remains the same. Nevertheless, it is still relevant to our work, as hesitation or guarding (even if not brisk) may have a similar profile.

In [13], the researcher used a three-layer one-directional LSTM network to automatically detect freezing gait behavior using data from the DG dataset, which obtained a mean F1 score of 0.76 with hold-out validation. This result suggests again the advantage of using LSTM on temporal data sequences with respect to other network. Indeed, Deep Neural Network (DNN) and CNN respectively achieved only F1 scores of 0.633 and 0.684.

In [15], a LSTM network that was extended to use temporal attention mechanism has also been tested on the DG dataset, which was expected to be able to highlight the freezing moment during walking. Their result showed the success of using a temporal attention mechanism to differentiate the abnormal behavior from normal movement along the temporal dimension of the movement. This study suggests that attention mechanisms could be also useful for PBD in CP.

However, differently from dealing with just one type of altered behavior (freez-

ing) during one type of activity (walking), our aim is to detect the existence of protective behavior across a variety of functional activities. In this case, protective behavior does not just occur temporally but also spatially as it is adopted by the person, at that moment, to protect the body parts felt in danger and use the body parts felt of aid and strong. Actually, temporal and spatial (configurational) attention mechanisms were used in different types of architectures [15, 16, 17], showing clear improvement in wearable HAR. Hence, it becomes interesting to explore how they may improve PBD. We will review these architectures in Chapter 5 where we present our own attention-based approach to PBD.

Addressing the problem of limited data size of the DG dataset, [61] proposed to use a deep CNN with 7 layers, where each layer contains a convolutional layer, a batch normalization layer, and an activation layer using rectified units (ReLUs). Instead of using a fully-connected max-pooling layer, at the end, a global averaging pooling layer was applied. Particularly, to fit such a deep CNN network with the DG dataset, multiple data augmentation methods for movement data were tested. The results (Accuracy=86.76%) showed a noticeable improvement acquired by combining data augmentation methods called Rotation (manually change the placement of sensors, with accuracy of 82.62% when used alone), Permutation (re-organize the temporal location of within-window events, with accuracy of 81.16% when used alone), and Time Warping (distort the time intervals between samples to change the temporal locations of them, with accuracy of 82.00% when used alone) than just use the original dataset (Accuracy=77.54%). These results show that data augmentation methods particularly designed for movement data sequences can help alleviate the challenge of training a deep learning model with a dataset of limited size.

Another dataset called Stereotypical Motor Movements (SMMs) was developed for the research of detecting stereotypical movements showed by people having autism spectrum disorder (ASD), and was collected in a longitudinal study that first presented in [51], and later in [72]. The data were collected from six participants diagnosed with ASD using three accelerometers attached to both wrists and the torso. Two data collection scenarios were considered, one is to let the participant sit in a

lab alone interacting with the teacher who is familiar with the participant; another is to let the participant sit in a classroom together with other students conducting typical classroom activities (*e.g.*, eating lunch, spelling program, sorting). In [72], data were only collected in the classroom.

It is interesting to see that, in [51], the authors mentioned their pilot efforts in evaluating the window lengths from 200ms to 5s using their method combining time-frequency features and RF. The window length of 1s turned out to be the best for the task, and was continuously used in later works built on the dataset. However, details of such pilot study are missing, thus the knowledge provided in their work on data segmentation is limited to the choice of this 1s window length.

[62] used a network of 3 convolutional layers, each followed by an average pooling layer, on movement data in the SMMs dataset to detect movements stereotypical of this cohort within window lengths of 1 second (overlapping ratio of 87%). Their result of mean F1 score of 0.74 with the lab data outperformed the traditional feature engineering method with Support Vector Machines (SVM) and RF used in [51, 72]. Unsurprisingly, the mean F1 score obtained was only around 0.5 with data collected in the classroom, where the movement is less constrained and noisier. The poorer performance may also be due to the smaller volume of data, as the convolutional network used usually relies on a large set of data for training.

In a later work by them [63], they applied a more complex method for the same dataset. First, they used the same CNN network that was proposed in [62] to extract a discriminative feature space from the data, acting as a replacement for manual feature engineering. Second, a single layer LSTM network was employed to learn the temporal-dynamic feature, whilst an ensemble of LSTM learners was further used to improve the accuracy. Third, transfer learning was performed for the CNN network to use the knowledge learned from the lab to improve the detection on data collected from the classroom. On a balanced training set where the categories are equally distributed, the result of using CNN led to improved F1 scores of 0.78 and 0.56 on the two streams of lab and classroom data respectively after transfer learning. In comparison to their first study mentioned above, the use of balanced training and

transfer learning leads to improvement on the two data streams. Meanwhile, on the original unbalanced training set, the result achieved by using LSTM and ensemble of LSTM learners outperformed the CNN (F1-score of 0.75, 0.77, 0.71 on data collected in lab, and 0.48, 0.53, 0.40 on data from classrooms by CNN, single LSTM, and ensemble of LSTM, respectively).

Given the results from the above studies on the two datasets for abnormal behavior detection, we can again observe that the temporal information stored in the data sequence is critical for movement behavior analysis and was better modeled with LSTM-involved network, whilst the data augmentation technique was also beneficial to improve the performance on data of a smaller size.

2.3 Advanced Methods for Movement-based Tasks

The ongoing development of deep learning has resulted in a variety of spontaneous responses in movement-based research. Aside from the preceding successes in this area using vanilla deep learning approaches, we will now look at the more recent progresses using advanced models. Given that we are employing full-body movement data in our thesis, we also look at the literature that targets modeling with a similar data structure.

2.3.1 Wearable HAR with Attention Mechanism

One of the trend for wearable HAR studies with data collected from sensors attached to human body or vision-based motion capture systems is to make the model better learn the hidden bodily and temporal information of body movement. Recently, attention mechanisms have been explored to improve performances of wearable HAR, with the advantage of enabling the network to focus on more informative sensors (body positions) at important moments (temporal positions).

To encapsulate understanding of the relevance of each sensor attached to the human body, Zeng *et al.* [15] proposed an attention-based LSTM architecture, where a sensor-oriented attention module was used at the input level per timestep, with an additional temporal attention module embedded in a layer following it (see Figure 2.3 (a)). Their sensor-oriented attention module was implemented with a

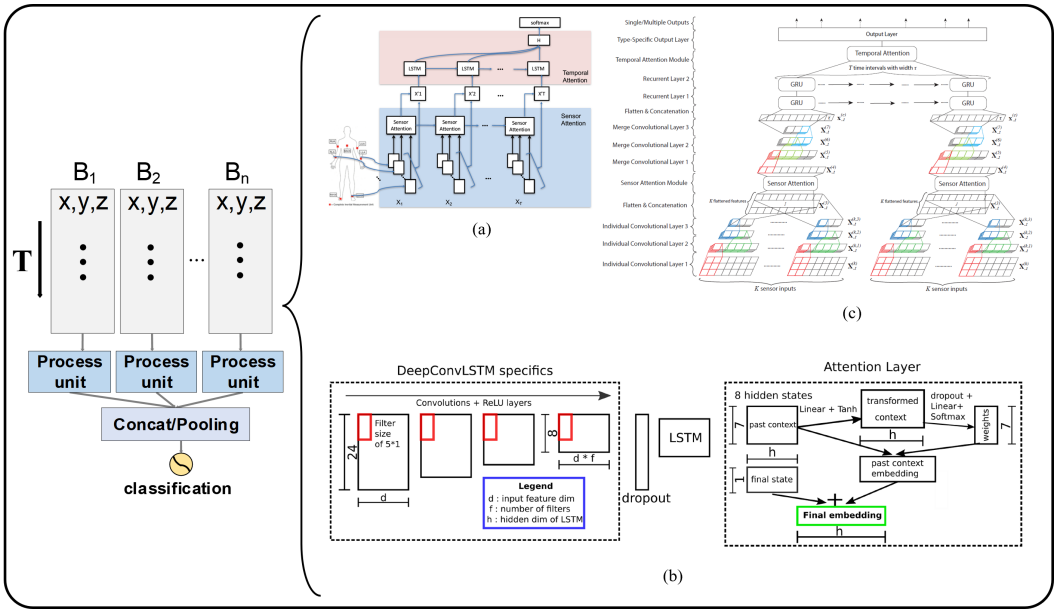


Figure 2.3: The trend we saw in recent models (a)(b)(c) proposed for sensor-based HAR is to use attention mechanism to capture the informative local movement per sensor position and the temporal saliency. Partially taken from [15, 16, 17].

softmax and bilinear function with input from different sensors at each timestep, while tanh and softmax activation functions were together applied to compute the temporal attention based on the output of the LSTM layer. Their method improved the performance on three HAR benchmarks (PAMAP2 [66], DG [71] and Skoda [67]) comparing to the performances of vanilla neural networks. Visualization of the attention scores showed that the network is able to do subset learning of sensors at important moments, especially for the detection of freezing behavior in DG dataset.

Along the same idea, Murahari *et al.* [16] focused on the implementation of temporal attention for HAR by embedding it at the end of a convolutional LSTM network (Conv-LSTM) [14] (See Figure 2.3 (b)). Similar to [15], they used tanh plus softmax functions to compute the attentional scores, with the difference being that they used the weighted sum of all previous LSTM hidden states instead of only using the last one (the output of Conv-LSTM) for classification [15]. Their experiments showed improved performance in comparison with plain Conv-LSTM and Bi-LSTM.

Another related approach called QualityDeepSense (See Figure 2.3 (c)) was proposed by Yao *et al.* [17], which was motivated by the problem of sensor reliability in mobile sensing. That is, while multiple sensors are deployed at the same time, this

problem assumes that only a hidden subset of sensors is able to provide the reliable information. The concept is slightly different from the studies above but still it can be approached using attention mechanisms. The deep sense framework [73] they used comprises a convolutional neural network at a lower level used to extract information from sensors at each timestep and a Gated Recurrent Unit (GRU) network used to learn the temporal dynamics through all timesteps at a higher level. To implement the attention mechanism, they used two softmax functions to compute the attentional scores specific for sensor attention at a lower level and temporal attention at a higher level. Such combination of sensor and temporal attention is similar to [15], and led to better performance on the HAR dataset [74] compared with the original Deepsense framework. Through statistical analysis, they also demonstrated that the model is able to pay less attention to the sensor set that has less sensing quality.

Studies presented above suggest that explicitly designing an attention mechanism can help a model to better learn patterns in data from multiple sources (*e.g.*, sensors attached to multiple anatomical points of a human body). However, we noticed two key limitations.

- Sensor attention and temporal attention are computed on different scales of information, *i.e.*, sensor attention is computed with low-level input data at each single timestep and temporal attention is computed with the output of the LSTM/GRU layer over a period of time spanning multiple timesteps, resulting in a gap between the two attention results. Therein, potential conflicts between the learning of two attentional scores given the knowledge acquired at different levels may hinder further performance improvement.

- It is problematic to compute sensor attention directly from input data per timestep, as the data may be too limited to justify the relevance of a sensor, especially in the context of protective behavior.

2.3.2 Skeleton-based HAR with GCN

More recently, the re-introduction of graph convolution network (GCN) [75, 76] offers a new method for HAR. One reason for the successful use of GCN on skeleton-

like movement data [77, 78, 79, 80, 81, 33] is that the human body can be naturally presented as a non-directed graph. Graph representation helps a model learn the biomechanical relationships between body segments without imposing knowledge about specific activities-of-interest. Noticeable improvements by GCNs are seen on several benchmark visual HAR datasets (*e.g.*, NUS RGB+ [6] and Kinetics [70]).

For implementing GC for skeleton-like movement data, some have altered the GC itself to facilitate a spatial-temporal operation [77, 80, 81, 33]. Others connect the GCN and LSTM via extra layers [78] or integrate GC within the gates of each LSTM unit [79] to enable a recurrent computation across time. The performance of these approaches fluctuates on visual HAR benchmarks [6, 70], and they have never been applied in the context of emotional bodily behavior across different activities.

Whilst the concept of body configuration is very much leveraged in visual HAR studies, enabled by the full-body MoCap data therein, it is not the case for ubiquitous wearable HAR and movement behavior detection. The wearable HAR literature has focused on using a small set of sensors to classify activity, with each study examining specific activities [82] or benchmark datasets [67, 65, 66]. Using a small network of sensors also increases applicability and reduces cost in real-life deployment.

However, as in the case of CP rehabilitation, critical information may not be in the movement of the main body segments involved, but in other body parts recruited to protect the body [21, 23, 22, 19]. For example, Olugbade *et al.* [38] showed the importance of head stiffness as protective behavior during sit-to-stand-to-sit and reach-forward, although the head movement is not needed to perform such activities. Psychology studies in CP point to the importance of assessing activity quantity as well as movement quality [24].

As a result, using full-body movement data (as in the EmoPain dataset) rather than a small set of sensors, to detect protective behavior across activities, is based on three arguments: i) full-body movement data is beneficial to capturing detailed movement behavior of multiple body parts for PBD across activities; ii) patients and clinicians see benefits and opportunities that such sensing technology offers, and are open to using it [28]; iii) full-body sensing is becoming more convenient as

wearable sensors are becoming smaller and integrated into clothes [83]. We evaluate the efficacy of our method on the simulation of small sensor sets in Chapter 6.

To the best of our knowledge, only one paper has investigated the use of GCN in bodily affective expressions [33], but considers just one scenario (gait) and acted emotional bodily expressions, a much simpler (stereotypical) problem to address. As such, they explored GCN alone and do not need to address the variety of activity and class imbalances of continuous data.

2.4 Addressing Challenges in Real-Life Scenarios

All the progresses witnessed above are based on an experimental setting where first data are collected with an ideal set of sensors, and second data are usually pre-segmented or specially treated according to different types of activity, thus the model is trained and tested on data without the noise introduced by irrelevant or transition activities.

Unfortunately, these settings do not always properly reflect real-life scenarios. In real-life situations, the number of sensors that can be used is often limited and sometimes malfunctioning. In addition, physical rehabilitation within continuous functional activities (rather than exercise sessions) at home is usually conducted in an unconstrained way where activities are connected with casual transition movements and thus activities conducted are not known in advance.

Driven by the possible challenges that could exist in real-world applications of our research, we review the literature on optimizing the sensor set and leveraging contextual information for improving the task-of-interest that may further inspire our work for PBD in ubiquitous and heterogeneous activity sequences. An illustrative diagram summarizing the literature reviewed in this section is shown in Figure 2.4.

2.4.1 Optimizing the Sensor Set

As reasoned in a comprehensive study [18] about using body-worn inertial sensors, for building a wearable HAR system, challenges considering the sensor set are: (i) diversity of human activities as the recognition of which requires careful selection of heterogeneous sensors that have different capabilities and characteristics, (ii)

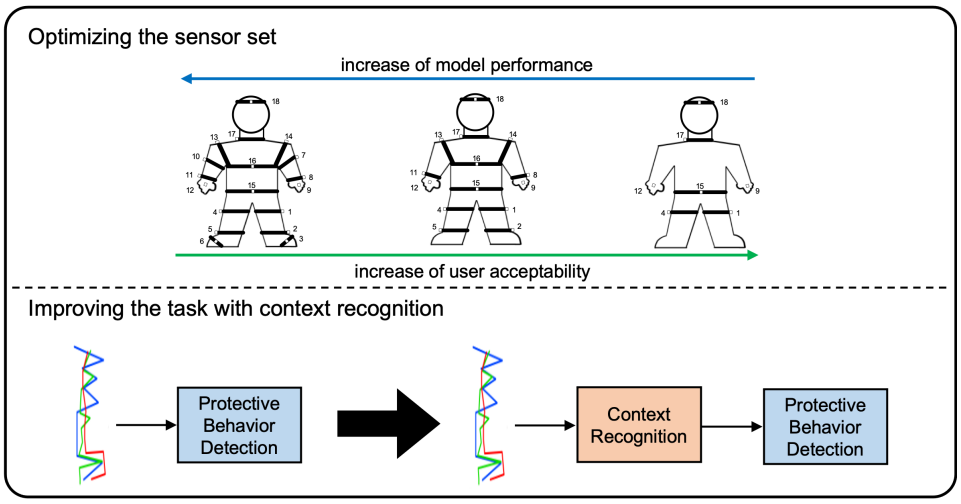


Figure 2.4: We review the literature toward solving two challenges that could exist in real-life scenarios. The first is the need for a compact sensor set and how to approach it. The second is, given more realistic and continuous data, how to improve the performance of a detection task with context recognition.

sensor composition as sensors can be added and removed opportunistically based on different application requirements.

For the first one, although the selection of sensor types is easier as we only care about the movement and muscle activity of the subject, we need to consider the sufficiency in capturing different activity types when we try to reduce the number of sensors. For the second one, aside from the removal of sensors given the users' or application's need, users normally prefer the sensor system to be compact, embedded, and easy to operate and maintain [84]. As a result, here, we review some relevant works on exploring more efficient sensor set for specific movement-related tasks to acquire some ideas to aid our research.

A case study is provided in [18] that analyzed the impact of the placement of sensors on activity recognition performance. The data for the experiments were collected with body-worn accelerometers and gyroscopes attached to hand, lower-arm, and upper arm. Activities performed include opening and closing a window, drinking with a bottle, cutting with a knife *etc.*, which are all performed by the upper limb. Their experimental setting is depicted in Figure 2.5. Through extensive experiments, the best result was achieved by using data collected with all the sensors (precision of 94.1%), with a competitive result found when only using data from the

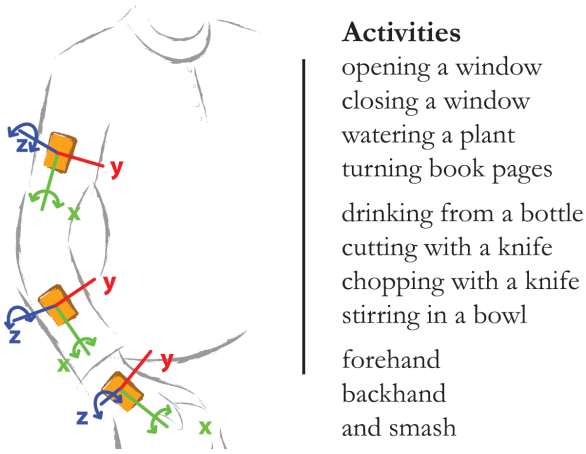


Figure 2.5: The sensor setup and activities used to analyze the impact of sensor placements on model performance in a study about wearable HAR. (taken from [18])

sensor attached to the hand (precision of 87.2%). The applicability of only tracking the hand could be owed to the type of activities included in their experiments, which all involved hand movements.

This study first provides us with a priori of conducting such a grid-search experiment to analyze the impact of sensor placements on model performance. Additionally, their results suggest that the direction of sensor reduction may fall into keeping the sensor(s) of most involved body parts of the targeted activities. Then, a question we would meet in PBD is that, since the targeted activity types involve almost all the body parts, what strategy should we use for sensor set optimization aside from grid search.

A more relevant study is conducted in [19] that developed an Ubi-EmoPain dataset with a smaller set of sensors for movement-related self-efficacy (MRSE) level detection. Unlike the EmoPain dataset that used a full-body motion capture suit comprising 18 IMUs, the Ubi-EmoPain dataset only adopted 4 IMU sensors placed on the head, trunk, upper leg and lower leg, with 2 sEMG sensors attached to right trapezius and L4/5 lumbar paraspinal muscles respectively (as shown in 2.6). Thereon, experiments based on the Ubi-EmoPain dataset achieved an average F1-score of 0.78 for pain-level recognition in three activity types.

An interesting point of this study is, the setup of sensors is discovered from another experiment based on the EmoPain dataset. During this experiment, a com-

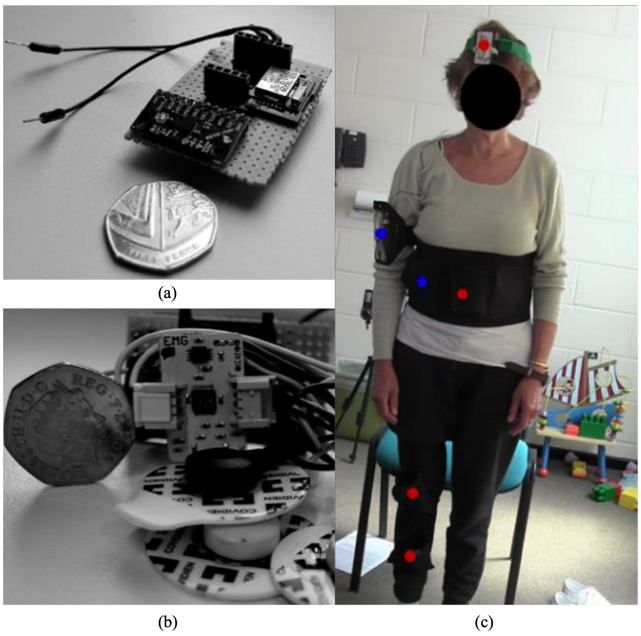


Figure 2.6: The more compact sensor set designed for pain-related behavior analysis seen in [19]: (a) The IMU sensor, SparkFun MPU9150. (b) The sEMG sensor, BITalino. (c) The placements of sensors on a participant, where the red dots are IMUs and blue dots are sEMG sensors. (taken from [19])

prehensive set of spatial and movement-based features for MRSE level detection was developed. These features were all computed based on one or several body parts. In order to see the importance of the features, they adopted the linear fixed models and wrapper-based breadth-first tree search for feature selection. By acquiring the frequency in which a feature appeared in the subsets returned to the selection, the importance of each feature is computed. Instead of analyzing the sensor placements directly, in this way they optimized the sensor setup by understanding the importance of features related to respective body parts during the detection task. Their results (mean F1 score of 0.79 for binary MRSE level detection) also show that sensors placed on one side of the body are also practical for movement behavior analysis in the context of CP rehabilitation.

However, their exploration on sensor set optimization is largely dependent on the activity types (*i.e.*, reach-forward and sit-to-stand), since their feature engineering process also relied on the characteristics of the activity type. Nevertheless, their results provide us the confidence that experiments on sensor set optimization con-

ducted on the existing dataset could be the important basis for designing future data collection protocol and maybe the integrated ubiquitous systems.

In general, in order to shed light on future data collection with wearable sensors and the establishment of a rehabilitation system, we could consider conducting experiments exploring the sensor-set dependency on PBD in a more challenging setting with continuous data sequences of various activity types through a grid search on the existing data collected from 18 IMUs, and referring to the knowledge presented in the study above to see that if movement data collected from one side of the body is informative enough for accurate PBD.

2.4.2 Improving the Task with Context Recognition

In real-life situations, physical activity, be it functional or exercise, is conducted in an unconstrained way, with boundaries between movements being less evident as often one movement transitions directly into another before completing it. Therefore, it is important to investigate if automatic detection of anomalous movement behavior could benefit from the detection of the context in which it occurs.

Such a problem could be approached with ideas explored in studies that aim at discovering the relevant contextual information to aid the task-of-interest. We look at three deep learning studies published in recent years that receive a high citation, and particularly their tasks-of-interest were conducted given the existence of different contexts. Therein, we aim to acquire a basic idea about how the informative context is located in their applications and what is the learning strategy used to leverage it.

In a study [85] on facial expression recognition in image or video, the contextual information they found useful includes the background scene and the action of the target subject or of other subjects in the scene, which were leveraged to aid the recognition of emotion expressed by the target subject's face. The method they proposed comprises two streams of face encoding and context encoding, and fusion layers that concatenate the two pieces of information by assigning the learned fusion weights of them with an attention module. Their experiments on two dynamic emotion recognition datasets showed improved performance in comparison to the variants without directly leveraging such contextual information (accuracy improvements are

2.91% and 3.38% on CAER [85] and AFEW [86] datasets, respectively).

While this study provides evidence of the importance of context, their method requires cropping the facial area out of the video frames (with the help of an extra face detector [87]) to build the input for the context recognition stream both during training and testing. Whereas, an ideal method shall automatically understand and extract the contextual information from the raw data simultaneously to aid the task-of-interest in a more integrated and end-to-end manner, at least during inference.

Instead of separating the construction of context information from data related to the task, a more unified learning framework for unconstrained person identification in video is seen in [88], which proposes a Region Attention Network (RAN) to acquire the useful visual context (different regions of all the person instances, *e.g.*, face, head, upper body, and whole body) and to integrate with the social context model (discovering/understanding the event or other persons each subject attends to) via a unified formulation for better person recognition performance (accuracy improvements are 7.39% and 6.28% on PIPA [89] and CIM [88] datasets, respectively).

With the help of RAN and the unified formulation, this method removed the need to manually prepare the image regions to aid context recognition. Additionally, this method explored the use of different types of context information, from low-level local spatial regions to high-level person-to-person interactions, and the combination of which leads to the best performance. Therefore, we can see that the ongoing advance in modeling concept and technique contributes a lot to the better integrated approach of using context recognition to aid the task-of-interest.

Aside from mining the visual context (*e.g.,* surrounding objects or the dynamic interaction between the object and the environment) that can be directly inferred from the image or video to aid the task-of-interest, the work in [90] further discovered the context that not directly present in data to aid object recognition in robot perception. To help the audience understand the contextual information other than visual cues, the authors provide two examples. First, they reason that the location of a camera could help recognition of alligator from crocodile, since both species live at different geographic locations on earth. Second, as a more relevant example to their method,

they describe the conceptual context that takes the form of associations between related concepts as: *I was told to look for a banana, so I may be likely to see one soon*. To provide such conceptual context for modeling, they adopted a cognitive architecture [91] to quantify the link (relevance) between the current situation (input data) and the chunks of buffer contents (working memory) so that the model could estimate what the object is not only given the visual content but also based on a conceptual assumption of it.

Although their improved performance on object recognition was seen in experiments conducted on a selected set of objects (*e.g.,* apple, raisins, coyote, and wire), the use of context beyond existing data is inspiring about how to search the space for context recognition. For example, in PBD with multiple annotators and without objective groundtruth, if the model is aware of the annotation trend among the annotators (*i.e.*, more agreed on protective or non-protective behaviors) at each input sample, its prediction could be better in line with all the annotators.

The above studies on leveraging context recognition show that i) instead of approaching the task-of-interest directly, recognition of the context present in data could aid the main task; ii) contextual information may exist beyond existing data but can be manually defined and provided to the modeling given our high-level conceptual understanding of the task.

2.5 Summary and Discussion

Through this literature review, the following gaps and insights have emerged, which will inform the studies in the following chapters.

- Past researches in psychology and clinical health have identified the existence of protective behavior in people with CP. Specifically, protective behavior is expressed as maladaptive strategies during functional activities by people with CP, mainly because of their fear of movement as a cause for increased pain or injury. Protective behavior is visually observable and provides information about the physical and psychological capabilities of the person during functional activities. Therefore, the detection of persistence of such behavior with respect to certain activity could

inform and help personalize the deployment of interventions and support, like when to send reminder of breathing, when encouragement is needed, and what kinds of adjustment in the rehabilitation plan could be beneficial.

- For facilitating the rehabilitation of people with CP by leveraging protective behavior analysis, past studies mainly focused on visual assessment and clinical person-to-person interaction, which are not easily accessible to a large group of people. Meanwhile, self-report based methods (*e.g.*, diaries) obviously lacks the timely intervention and suffer from the limited self-awareness in people with CP of their movement behavior, thus are not practical for improving the engagement of people with CP in physical rehabilitation beyond the clinic.

- Although ubiquitous technology appears as a great opportunity to facilitate self-directed rehabilitation, the research on automatic detection of protective behavior has received limited attention. The obvious limitation of previous studies is that separate models were built for different types of functional activity. In addition, detection was only per overall activity instance rather than a more fine-grained level as necessary for providing more personalized support in a continuous manner.

  While the literature in HAR provide us the evidence about the advantage of using recurrent networks for tasks based on movement data, a lack of investigation on data preprocessing methods especially parameters used in segmentation is generally found in previous HAR studies, despite they have been shown to be impactful on the model performance. Therefore, we identify the research questions here as: i) how to enable deep learning for activity-independent PBD in a more continuous manner? ii) how to transform the existing movement and sEMG data into practical training and testing sets for model development? and iii) what are the generalizable knowledge in data preprocessing we can derive from our experiments to future datasets on PBD or relevant tasks? We explore these in Chapter 4.

- Another important gap that emerged from the literature is the lack of a model that able to capture the variety in movement strategies adopted by people with CP across different activity types. This is critical as it would allow personalized

intervention at finer granularity by understanding what part of the body (spatially) or an activity (temporally) is feared. For this point, the more recent advances in wearable HAR studies using attention mechanism have shown a clear advantage in learning local movement dynamics than previous vanilla models.

We argue that the relevance of a sensor (one of the selected joint angles in our case) for PBD shall be better understood over a period (*i.e.,* over a movement segment), rather than at single timesteps. Whereas in a normal HAR task, activities may be recognized based on temporally-local relationships of body positions, protective behavior is exhibited in a dynamic process, which may necessitate a longer period of perception before a clear judgment can be made (please refer to Chapter 4 for details about how the duration of data frames may impact the model performance in PBD). Especially given the variability in how people express protection from harm or pain, the recognition of the activity type here may also benefit from a certain period of movement process.

In chapter 5, we develop a novel learning model that combines temporal and bodily attention mechanisms to improve the performance as well as to help capture such variety. A further evaluation of this method on a wearable HAR dataset shows better performances than other previous state-of-the-art methods for HAR.

- The advantage of using GCN in skeleton-based HAR, the need to model a large set of sensors, and high variability in body configuration information in PBD all suggest the importance of exploring the use of GCN in the context of PBD. It also brings together research work on HAR and PBD (or in general emotional movement behavior detection) that have surprisingly evolved separately, despite clearly representing activity and emotional bodily expressions that co-occur in real life with each altering the other. In chapter 6, graph convolution is employed to model the movement data captured by multiple IMUs per timestep. Given the success of LSTM in capturing temporal patterns of protective behavior [9, 10], LSTM layers are added to model the temporal dynamics.

- Furthermore, improved performances are seen in works that leveraged the context

recognition to aid the task-of-interest on the same piece of data. Aside from the context that can be easily inferred from existing data, efforts were also made to acquire higher-level contextual information to aid the task. As we aim to improve PBD in more challenging settings that help examine the deployment of our method in the real world, we wonder if recognizing the context is beneficial for our scenario and how should we locate the contextual information in the existing data. We study this in chapter 6 to improve PBD in continuous data of different activity types.

- Studies concerning the challenges present in real-life scenarios first showed the applicability of using smaller number of sensors for activity recognition, pain-level recognition, and movement-related self-efficacy estimation. However, it remains unknown whether the reduction of number of sensors may impact the PBD performance in continuous data, and what is the proper strategy to guide such reduction if we want to achieve a balance between user acceptability and model performance. In chapter 6, we look into this while modeling the continuous data.

Before reporting each of the studies carried out in this thesis, in the next chapter we provide a description of the EmoPain dataset [12] that has been used across all studies to evaluate the proposed approaches. We also present some of the basic techniques that have been used for benchmarking the performance of our proposed methods. Other datasets and techniques used to explore our questions will be presented in the respective study chapters.

Chapter 3

Methodology

In this chapter, we present the EmoPain dataset that we use to evaluate our methods in the thesis. We further describe the data preprocessing methods and vanilla models that we explore in Chapter 4, which are also the important building blocks for the experiments conducted in other chapters. Finally, we present the validation methods and metrics that we use across the thesis.

3.1 The EmoPain Dataset

The EmoPain dataset [12] contains Inertial Measurement Unit (IMU) and surface Electromyography (sEMG) data collected from 26 healthy and 22 Chronic Pain (CP) participants performing physical activities selected by physiotherapists. Healthy participants (non-athletes) were included in the dataset to capture natural idiosyncratic ways of moving, rather than considering a gold standard model of activity execution that is no longer an approach used by physiotherapists in CP physical rehabilitation. Healthy participants were assumed to show no protective behavior in the data collection.

Although the original dataset contains data from 22 people with CP, 4 of them are left out because of errors in their sEMG data recordings. To avoid biasing the model toward healthy participants, 12 healthy people are randomly selected for the experiments conducted in this thesis. The number of healthy people is kept smaller than the number of CP participants because some CP participants did not repeat all the activity types. As a result, data used are from 12 healthy and 18 CP participants.

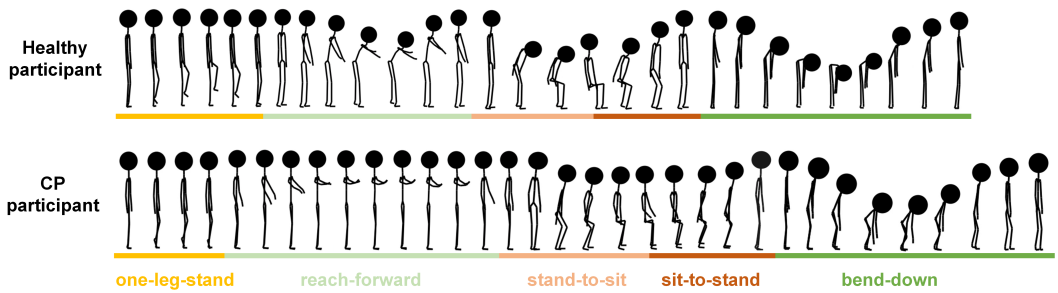


Figure 3.1: Avatar examples made from movement data in the EmoPain dataset of a healthy and a CP participant performing the five functional activities.

3.1.1 Building Blocks for Complex Functional Activities

Each data sequence of the dataset comprises five activities-of-interest (AoIs) in addition to data related to the transition or short relaxing activities (*i.e.*, standing still, sitting still, and walking). In Chapter 4 and Chapter 5, we focus on the continuous detection within each individual AoI instance, where the transition data is left out through manual pre-segmentation. In Chapter 6, we move to continuous detection within the full data sequence of activities, leaving out the last instance of walking.

The five AoIs are bend-down, one-leg-stand, sit-to-stand, stand-to-sit, and reach-forward. Figure 3.1 shows the avatars of a healthy and a CP participants performing these five activities. The AoIs were selected by physiotherapists in the development of the EmoPain dataset for two reasons. First, they involve movements that people with CP tend to avoid or be very cautious about, as they are perceived as painful or injury-inducing. In addition, they comprise basic movements that occur in a variety of more complex daily functional activities: a person may need to **bend** to load the dishwasher or tie the shoes, or a person may **reach forward** to pick up something from a high shelf or clean the trunk of their car, and **stand/balance on one leg** to climb stairs or even walk.

Given that the activities used in this work can be considered as building blocks for more complex functional activities, experiments conducted on this dataset should shed some light into future work using other relevant datasets that build on these five basic activities in the context of protective behavior detection (PBD) for CP rehabilitation or in general affect-influenced movement behavior detection.

Participants were asked to perform two trials of the sequence of activities with

two levels of difficulty. In each trial, activities were repeated three times, although some CP participants skipped a few repetitions perceived as too demanding (*e.g.*, bend-down).

During the normal trial, participants were free to perform the activity as they pleased, *e.g.*, they could stand on their preferred leg and start the activity at any time they preferred. For the difficult trial, participants were asked to start on a prompt from the experimenter, and to carry a 2 kg weight with both hands or in each hand during reach-forward and bend-down, respectively. These more difficult versions of the same activities simulated real life situations where a person is under social pressure to move or is carrying bags or other objects. Again, these are often suggested by physiotherapists to help people with CP gain confidence in moving even outside the home [31].

As a result, we treat two trials of activities performed by one participant as two different sequence (a typical sequence is shown in Figure 3.1.

For data we have access to, 5 healthy people and 11 people with CP did activities at both levels of difficulty. Therefore, we have 17 sequences (7+5×2) from the healthy and 29 sequences (7+11×2) from people with CP, which lead to 46 sequences in total. Each of these sequences contains all the selected activities (repeated at least once) performed by one participant at one level of difficulty.

3.1.2 Movement and Muscle Activity Data

A wearable motion capture suit named MetaMotion IGS-190 [92] comprising 18 IMUs was used for the data collection. As provided in the EmoPain dataset [12], at each timestep, 3D coordinates of 26 body joints were calculated from the raw data stored in a Biovision Hierarchy (BVH) format. Within the BVH file, the metadata includes the skeleton proportion of the participant (*e.g.* the length of limbs) and position on the body that each sensor was attached to (Figure 3.2 (a)). Using a Matlab MoCap toolbox [93], the approximate position of 26 body joints in the 3D space was estimated based on the metadata, the gyroscope, and accelerometer data. An illustration of such transformation from IMUs to positional triplets of body joints is shown in Figure 3.2 (b).

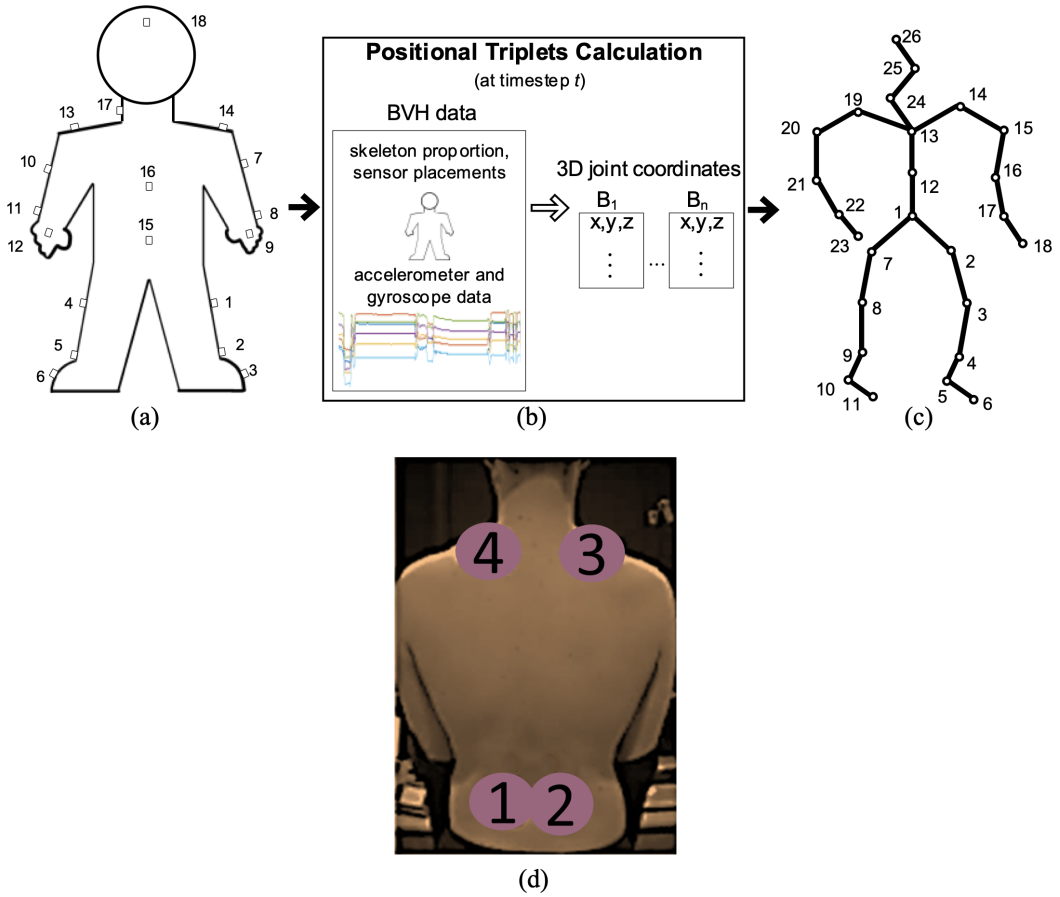


Figure 3.2: Illustrations of a) the placement of 18 IMUs, b) the calculation of 26 sets of 3D joint coordinates, c) the skeleton graph showing the connection of 26 anatomical joints, where each node represents a human body joint, and (d) the placements of the 4 sEMG sensors on trapezius (3, 4) and L4/5 lumbar paraspinal (1, 2) muscles, taken from [12].

It should be noted that data collected from the participants' feet (node number 5, 6, 10, and 11 seen in Figure 3.2 (c)) are noisy and hence not used in this thesis. This was due to interference with the electric cables placed under the floor where the data collection took place.

The muscle activity was recorded with 4 BTS FreeEMG300 wireless sEMG sensors placed on the trapezius and L4/5 lumbar paraspinal muscles, as shown in Figure 3.2 (d). Those sEMG sensors were operating at 1kHz. The position of the sEMG sensors was based on the literature [94] that shows altered activations of the lumbar paraspinal muscles in people with chronic low back pain due to fear of movement. Similarly, the trapezius muscles were selected as people tends to contract

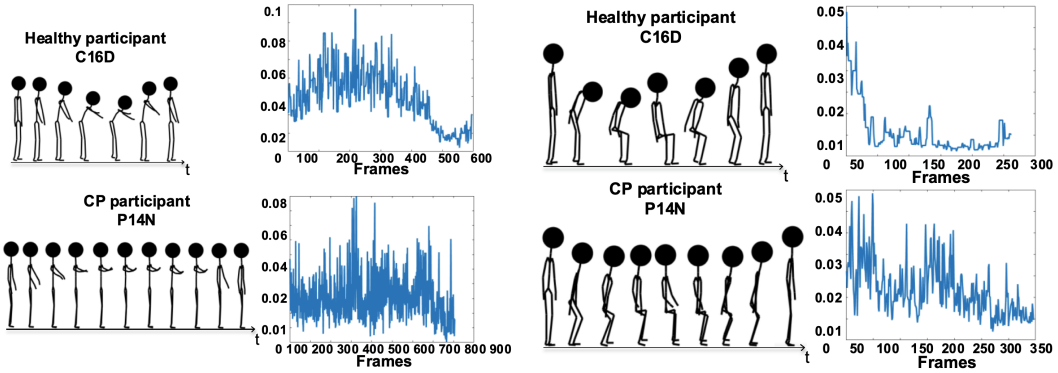


Figure 3.3: Avatars representing the temporal sequences of movement and sEMG data of healthy and CP participants during reach-forward (left) and stand-to-sit and sit-to-stand (right) in the EmoPain dataset. The sEMG signal plotted for each avatar sequence is the average upper envelope of rectified sEMG data collected from two sensors on the lower back.

them when anxious. For the muscle activity diagrams in Figure 3.3, the value was computed as the average upper envelope of rectified sEMG data collected from the two sensors attached to the lower back.

Examples of protective and non-protective behavior samples from the EmoPain dataset are shown in Figure 3.3. These avatars were built directly from participants' movement data and represent instances of activity from the dataset, even though the length of each sequence is not representative of the real duration. Continuous animations of C16D [95] and P14N [96] are available.

As shown in Figure 3.3 (left), for reach-forward, differences between the healthy and CP participants exist in the stretching range and also the different strategies, with the latter simply raising the arms but not bending forward. For C16D, we can see an expected curve of the sEMG data as the lower-back muscles activate to hold the stretching forward position and then relax when returning to the standing position. Instead, for P14N, the activation/deactivation curve is missing with a high variation across the movement, despite the person was not even attempt to stretch forward and only brings the arms up. Such suggest the fear of this participant in preparing for the movement, which continues to the end. We can also observe another strategy from participant P14N who is keeping the feet closer together during reach-forward that makes the bending more difficult. Often, people with CP are

unaware of avoiding facilitating movements/postures, as their attention is on pain rather than proprioceptive feedback.

During the stand-to-sit-to-stand, there is no much use of the lower-back muscles, but this expected pattern is not present in P14N where again the muscles remain active, possibly due to fear despite the lack of trunk bending. These are just examples of strategies used by people with CP, as each person personalizes the strategies to his/her perceived physical capabilities and own understanding of what could be a dangerous movement.

3.1.3 Low-Level Feature Computation

Aside from directly using the raw coordinates of each body joint, we use the 13 low-level features provided in the dataset as suggested in Aung *et al.* [12], corresponding to 13 joint angles in 3D space calculated based on the 26 anatomical joints. In addition, we also use 13 'energy' features provided in the dataset that are computed using the square of the angular velocities of each angle. The description of the 13 joint angles is shown in 3.5.

The dataset also provides the muscle activities captured from four back muscle groups in the form of the rectified sEMG data. We therefore have 30 features in total for each sample: 13 joint angles, 13 energies, and 4 rectified sEMG signals from the original dataset. To maintain the dimension order of the data, the feature matrix is formed as Figure 3.4.

As shallower models are used in Chapter 4 and Chapter 5, to aid the representation learning by reducing the number of dimensions of the original data comprising many coordinates ($22 \times 3 = 66$), we use these low-level features ($13 + 13 + 4 = 30$) as the input of our model. For the study presented in Chapter 6, as motivated by the use of Graph Convolutional Network (GCN), we turn to using the raw coordinates

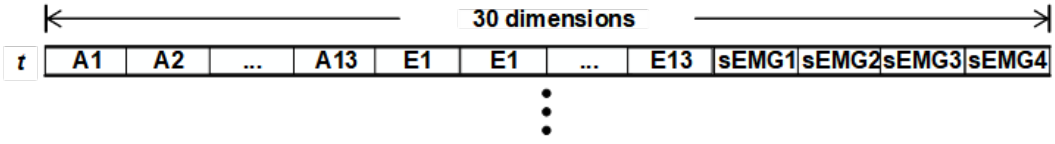


Figure 3.4: The feature matrix at a single timestep t. A1 to A13 are the inner angles, E1 to E13 are the energies and sEMG1 to sEMG4 are the rectified sEMG data.

Angle Description

Angle Name	Respective Anatomical points
1 Crown-Hip-LeftFoot	26-1-4
2 Crown-Hip-RightFoot	26-1-9
3 Spine-Hip-LeftLeg	12-1-3
4 Spine-Hip-RightLeg	12-1-8
5 LeftKnee	2-3-4
6 RightKnee	7-8-9
7 LeftElbow	15-16-17
8 RightElbow	20-21-22
9 LeftShoulder	24-14-15
10 Rightshoulder	24-19-20
11 Leftshoulder-LeftUpperLeg-LeftLowerLeg	16-14-2
12 RightShoulder-RightUpperLeg-RightLowerLeg	21-19-7
13 Neck	12-24-26

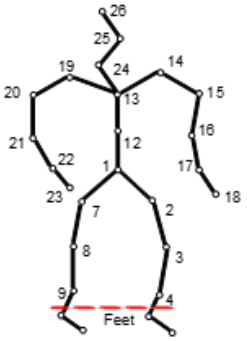


Figure 3.5: Description of the 13 joint angles. Data collected from the participants' feet are noisy and hence not used in this thesis.

of each body joint as the input directly, as the structure of the network tolerates the high-dimensional information about the body configuration.

3.1.4 Data Annotation and Ground Truth

Multiple annotations of data were carried out using participants' self-reports and expert annotations based on videos of trials. In this thesis, we focus on expert annotations of protective behavior, while we also know that reported pain levels are not directed related to presence or absence of pain behavior [22].

Two physiotherapists and two psychologists working with clinical populations were recruited to rate each data sequence by viewing footage from the on-site camera. For each type of protective behavior (*e.g.*, guarding/stiffness, bracing/support, and rubbing/stimulating) definition, the experts marked the timesteps where the specific behavior started and ended. Thus, we have the label of different types of protective behavior per timestep.

However, rather than discriminating between different types of protective behavior, we treat them as a unique class referred to as *protective behavior*. The reason is that the number of instances for each behavior is too limited to investigate the use of deep learning models. In addition, the discrimination that matters in the first place for providing personalized feedback is whether protective behavior has

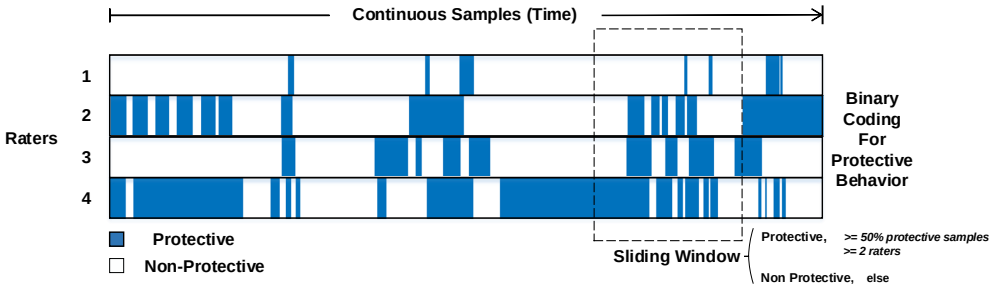


Figure 3.6: The visualization of the binary coding for protective behavior by 4 expert raters. Different types of protective behavior are treated as the same unique class.

occurred. With a larger dataset, a finer analysis could be used to further personalize the feedback and intervention, but it is outside the scope of this thesis.

By merging the annotation of different types of protective behavior, each data sample is associated with 4 binary labels (one from each rather: where 1 stands for the presence of a protective behavior and 0 stands for absence of such behavior according to the specific rater). Figure 3.6 presents a visualization of the coding result of a data sequence of one CP participant. We can see that, a certain level of disagreement exists in the annotations on protective behavior.

Given the disagreements between different annotators, we need to provide the model suitable labels for training and testing. The majority-voting approach is the typical approach used in affective computing [48]. Hence, together with the segmentation using a sliding window, we define the ground truth of each segmented frame as protective if at least two raters each found at least 50% of the samples within it to be protective. This method is used by default to define the ground truth for the EmoPain dataset in this thesis.

3.2 Vanilla Neural Networks

This section describes the vanilla neural networks that we use for comparison or as the building block in this thesis. These methods are commonly used in the literature [13, 60, 14, 62] on activity recognition and behavior detection with sensor-based movement data.

3.2.1 Stacked-LSTM and Dual-Stream LSTM Networks

Unlike the convolutional neural network (CNN), which was usually adopted for spatial feature extraction in image and video and recently for wearable HAR, recurrent neural network (RNN) showed better capability for the learning from time-dependent data sequences. Previous studies [13, 60] show that RNNs, particularly LSTM-involved networks, outperform other network architectures like CNN on processing data sequences collected with wearable sensors.

A forward-passing RNN structure is shown in Figure 3.7. The input to it is a temporal sequence, for which the network computes state information and passes forward along the temporal direction. The core of an RNN architecture is the processing unit, which is an LSTM unit for LSTM networks. The LSTM unit [97] solved the vanishing gradient problem that traditional RNN had faced in back propagation over a long temporal sequence. An LSTM unit updates its internal states based on current input and previously stored information [97]. The LSTM unit that we use in this work is the vanilla variant without peephole connection [98].

A typical and simple method to manage multimodal input (i.e., body movement and muscle activity data in our case) is to concatenate the multimodal data at the input level, as done by default in our stacked-LSTM architecture as shown in Figure 3.7. In addition, we consider another typical approach that is often used in affective computing [19, 38], where a late-fusion strategy is used to first model the data of two modalities separately before fusing the features learned from them at a later stage for final prediction. We refer to this method as Dual-stream LSTM, as shown in Figure

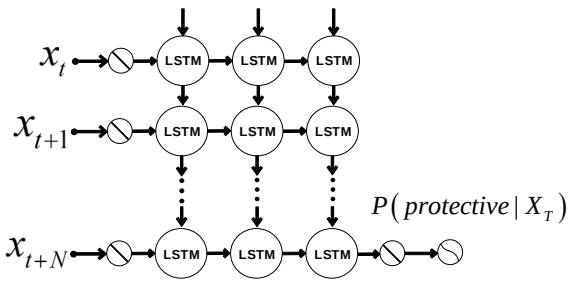


Figure 3.7: The typical recurrent neural network structure using LSTM unit.

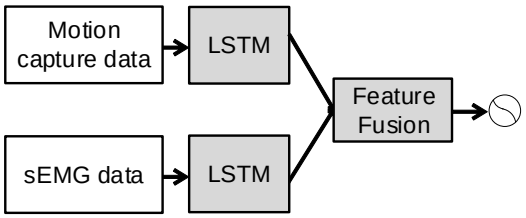


Figure 3.8: The Dual-stream LSTM network, where movement and sEMG data are processed separately. Each LSTM block is stacked-LSTM that without a classifier.

3.8. In the figure, the data of each modality is processed by a *stacked-LSTM* network that without a classifier, and the outputs from both modalities are concatenated along the feature dimension for final prediction.

For both the stacked-LSTM and Dual-stream LSTM, the computation happens within the LSTM unit is the same. At timestep t, the input to the corresponding LSTM unit comprises the current input data X_t, previous hidden state $\mathbf{H}_{t-1}$, and the previous cell state $\mathbf{C}_{t-1}$, while the output comprises the current hidden state $\mathbf{H}_t$ and cell state $\mathbf{C}_t$. By using this form, the output of at each timestep is based on the previously consecutive knowledge acquired.

The states are updated with an *Input Gate* with output $\mathbf{i}_t$, a *Forget Gate* with output $\mathbf{f}_t$, an *Output Gate* with output $\mathbf{o}_t$, and a *Cell Gate* with output $\tilde{\mathbf{c}}_t$. The computation within a LSTM unit at timestep t can be written as

$$\varphi_t = \sigma\left(\mathbf{W}_{x\varphi}X_t + \mathbf{W}_{h\varphi}\mathbf{H}_{t-1} + b_\varphi\right), \tag{3.1}$$

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_{xc}X_t + \mathbf{W}_{hc}\mathbf{H}_{t-1} + b_c\right), \tag{3.2}$$

where $\varphi_t \in \{\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t\}$, $\mathbf{W}_{(\cdot)}$ and $b_{(\cdot)}$ are the weight matrix and bias vector, respectively. $\sigma(\cdot)$ is the sigmoid activation. Then, the output of a LSTM unit is computed as

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{3.3}$$

$$\mathbf{H}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t), \tag{3.4}$$

where $\odot$ denotes the Hadamard product. The processing at the next timestep $t+1$ would take the current output $\mathbf{C}_t$ and $\mathbf{H}_t$ to iterate with the same computation mentioned above.

As we evaluate the parameter impact of the data segmentation, the length of the input layer is adjusted to the length of the input data frame created by each different segmentation size.

Using the output at the last timestep of the last LSTM layer $\mathbf{H}_T$ in a fully-connected softmax layer, the computation of class probability $P = [p_1, ..., p_K]$ where K denotes the number of classes and the final one-hot label prediction Y can be written as

$$P = softmax(\mathbf{W}_H \mathbf{H}_T + b_H), \tag{3.5}$$

$$Y = \arg\max_{[1 \cdots K]}(P), \tag{3.6}$$

where $\mathbf{W}_H$ and b_H are the weight matrix and bias vector of the softmax layer.

For the Dual-stream LSTM network, the last outputs of last LSTM layers of both modality streams (i.e., $\mathbf{H}_t^{movement}$ and $\mathbf{H}_t^{emg}$ for the movement and sEMG streams respectively) are concatenated and processed by the fully-connected softmax layer to repeat the computation written in Equation 3.5-3.6 for prediction.

3.2.2 Relevant Vanilla Models

Aside from LSTM networks, a comparison with the following methods that have been used in movement-based tasks is additionally conducted in Chapter 4.

- CNN [62]. The 3-layer CNN architecture used in this thesis is implemented according to [62], while the classification result is produced by a softmax layer at the final stage instead of using an extra SVM classifier. The convolutional kernel size is 1×10, with max pooling size set to 1×2 and number of feature maps to 10.

- ConvLSTM [14].The architecture is the same as what was used in [14]. The size of the convolutional kernel is set to 1×10, while max pooling size is 1×2 and the number of feature maps in convolutional layers and hidden units in LSTM layers is set to 10 and 32 respectively.

- Bi-LSTM [13]. As a variant of forward-passing LSTM network, bi-LSTM network utilizes context information in both the 'past' and the 'future' to compute the output at each timestep. We implement the bi-LSTM according to [13]. The hidden units in each LSTM layer is set to 16.

- Random Forest [12, 29]. We use a random forest (RF) algorithm with 30 trees for frame-based detection, which is then referred to as RF-frame. We extract length-fixed feature vectors for each frame as the input to the RF algorithm. The feature vectors computed from all the frames are further divided into training and test sets based on the given validation method. The features computed comprises the range of the joint angles, the mean of joint acceleration value, and the mean of rectified sEMG value, which were used in [29]. The dimension of the input feature vector was 30.

3.3 Validation Methods and Metrics

Three different validation methods are included to evaluate the model performance of the approaches in the following chapters.

First, a 6-fold leave-some-subjects-out (LSSO) cross-validation is included, where at each fold data of 5 out of the 30 subjects are left out and used for testing. To balance the number of CP and healthy participants, we ensure that each test fold contains data from 3 CP and 2 healthy participants, respectively. Such validation is similar to the hold-out validation conducted in wearable HAR.

Second, the standard leave-one-subject-out (LOSO) cross-validation is applied to further demonstrate the generalization capabilities of a model to unseen individuals. Here should be noted that, as the model achieves nearly 100% accuracies for all the healthy participants during LOSO that possibly due to the imbalanced distribution of non-protective (majority) and protective (minority) behavior samples, we report the average LOSO performance across all the participants in Chapter 4 and Chapter 5, while the average LOSO result achieved on the 18 patients' folds are reported in Chapter 6 given the obvious imbalance between the non-protective and protective behavior classes.

Finally, we envision that the use of our models will be in the context of personal rehabilitation where the model can be tailored to the same individual, so a cross-validation by leaving some instances out (LSIO) is also used. Therein, data (not from the same instance) from a participant could appear both in training and test sets.

Differently from medical applications for diagnoses, detections of presence and absence of protective behavior in chronic pain rehabilitation are both critical for the management of chronic pain, as they call for different types of support. In fact, physiotherapists operate on observations of both presence and absence of protective behavior to help patients adapt strategies to cope with *bad* and *good* days and gradually build a sense of capability [99]. Therein, advices and opportunities for the patient are provided accordingly [26, 38, 24, 25]. Based on these literatures and a discussion with physiotherapists, a system should detect both to provide the appropriate support and advices. For example, when the absence of protective behavior is frequently detected, the system may help the user avoid overdoing by reminding to take breaks, a critical problem in chronic pain management. Instead, when protective behavior is frequently detected, the system would provide feedback that help increase awareness of capabilities of the user.

Given that in this thesis we treat PBD as a binary classification problem where the detection of both protective and non-protective behavior is similarly important, we report the Macro F1 score as a metric that considers the performance at each class. The Macro F1 Score (Mac.F1) is computed as:

$$F_m = \frac{2}{|c|} \sum_{c} \frac{pre_c \times recall_c}{pre_c + recall_c}, \tag{3.7}$$

where pre_c and $recall_c$ is the precision and recall ratio of class c. Moreover, for completeness, the accuracy (Acc), mean precision (Pre), mean recall (Re), and confusion matrices are also used in the following chapters.

The macro F1 score and accuracy with a fixed threshold are the only metrics used in the first two study chapters (i.e., Chapter 4 and Chapter 5), since the imbalance between the two classes is moderate as we manually removed the transition parts of the data sequences. The class imbalance increases noticeably in Chapter 6 as we

move to using continuous data sequences, thus we further used another metric of Precision-Recall Area Under the Curve (PR-AUC). Still, we consider the importance of both classes, thus confusion matrices and macro F1 scores are reported.

To further understand how different architectures and parameters compare with each other, we also include statistical tests (Friedman test, post-hoc Wilcoxon Signed Rank test, and inter-rater correlation) on the LOSO results (F1 scores for all the LOSO folds). The statistical tests help demonstrate the generalizable significance of the results from our experiments. The 95% confidence interval is provided to each macro F1 score and average accuracy by using the t statistics computed from F1 scores and accuracies of all the LOSO or cross-validation folds of each method in the experiment, respectively. Repeated-Measures ANOVA and post-hoc t test are not used, given that the test of normality is not passed given our LOSO or cross-validation results, *i.e.*, we find $p < 0.05$ in our Shapiro-Wilk tests.

In the next three chapters we report the main studies of this thesis followed by a discussion of the contributions this thesis makes to the relevant fields.

Chapter 4

Exploring Vanilla Models and Data Preprocessing Methods

In this chapter, we aim to answer the first research question that is how deep learning could be leveraged to conduct activity-independent protective behavior detection (PBD). We approach this question by exploring and evaluating several fundamental factors of the research on this topic.

First, we verify the advantage of recurrent neural networks in PBD by comparing different types of vanilla neural networks, which have been commonly used in the previous literature on activity recognition. Second, we explore the impact of data preprocessing methods, namely data segmentation and augmentation methods, on model performance of PBD. In particular, we explore the relationship between the use of the different techniques with respect to the nature of different movement types critical to chronic pain (CP) self-directed rehabilitation. The aim is not only to set basic building blocks for the main studies of this thesis, but also to start to understand how the findings emerging from this study could extend beyond our dataset.

In summary, this chapter has the following contributions.

- We extend the state-of-the-art by showing the feasibility of activity-independent PBD using deep learning across and in a continuous manner within pre-segmented activity instances. This moves the field one step closer to being able to continuously detect pain-related behavior in everyday life without knowing the type of activity in advance.In addition, it allows knowing at what stage of the activity the behavior

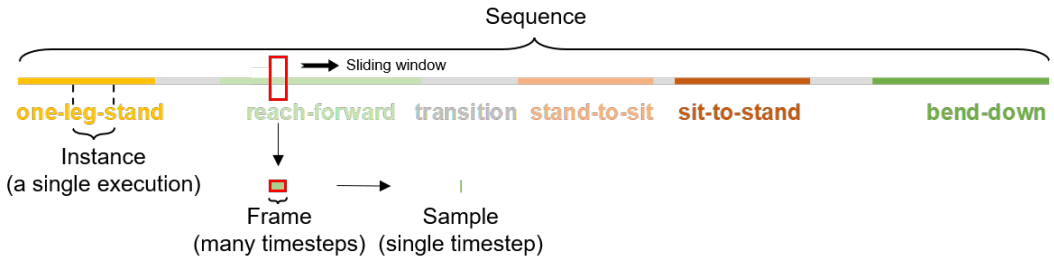


Figure 4.1: Illustration of the different references of data at various scales.

appears to better inform personalized interventions.

- A set of data augmentation methods and their combinations is investigated for dealing with the limited size of the existing dataset. An analysis and discussion of these methods shed light into how each of them could contribute to PBD beyond our dataset.

- The impact of data segmentation parameters on detection performance is also analyzed. Despite the optimal segmentation window length for PBD being dependent on the activity type, we provide a set of criteria to identify values for this parameter that work across different activities, showing how our approach could generalize to other datasets for PBD and in general affective movement behavior detection.

4.1 Data Preprocessing Methods

In this section, we describe the data pre-processing methods, namely segmentation and augmentation, that are first evaluated in this chapter as well as commonly adopted in this thesis to enable the use of deep learning models.

To avoid ambiguity, we clarify that: 'sequence' refers to the data sequence containing all the activities performed one after the other by a subject during one trial; 'instance' stands for data of a single activity execution; 'frame' is a set of consecutive timesteps extracted from the sequence; 'sample' is a single timestep (for our case is at 1/60 second as the sampling rate is 60Hz), and each sample is associated to a vector containing the movement and sEMG data. An illustration of such naming is shown in Figure 4.1.

4.1.1 Data Sequence Segmentation with Sliding Window

Given the temporal nature of movement data, we use a sliding window segmentation approach [18] to create the input for the neural networks. Therein, two situations are considered. In the first two studies of this thesis, we focus on recognizing protective behavior within activity instances where the transitions are manually left out, before going to conduct continuous detection with the long data sequence in the last study.

Therefore, we first apply the segmentation within the activity instances of the same activity type. Figure 4.2 gives an illustration of the segmentation conducted within each activity type. Note that the model does not take the type of activity as an input in the modeling process, but instead aims at generalizing PBD across all activity types.

In such segmentation within activity instances, one issue is to handle edge cases, *i.e.,* how to pad data when the sliding window is at the end of an activity area. We explore three typical ways of handling such case in the context of sensor data with an aim to understand their effect on PBD.

- Zero-padding, which is to pad the frame with zeroes. This is a typical approach used in activity recognition tasks of computer vision literature. [100, 101].
- Last-padding, which is to use the last sample of the current activity instance and copy it to the frame until the number of samples equal to the window length.
- Next-padding, which is to directly use the samples following the activity instance within the same data sequence for padding, as a way to simulate continuous natural

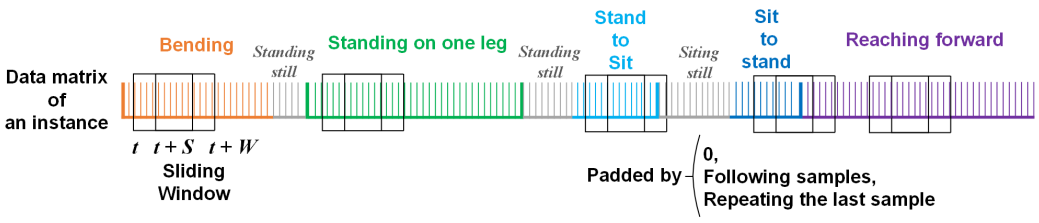


Figure 4.2: The sliding-window segmentation applied in the first two studies is conducted separately for each activity type, where different padding methods are considered for each window sliding outside an activity instance. t is the starting timestep of a window, S is the sliding step, W is the window length.

transition between activities.

In this chapter, we first compare these different padding methods. Then, we analyze the impact of different sliding window lengths on the model performance per activity type and across all the activity types.

4.1.2 Data Augmentation

Data augmentation is critical for mitigating the risk of over-fitting that rises when applying deep learning on smaller datasets. To address the limited data size and more generally the difficulty of capturing naturalistic dataset from people of special groups, we investigate the suitability of data augmentation techniques for PBD.

Following the progress seen in wearable HAR literature [61] about using data augmentation to improve the model performance, we use the following augmentation methods that least influence the temporal information of movement data.

- Reversing, which is to reuse data in a temporally reversed direction. This method is used as some activities can be thought of mirror reflections, *e.g.* stand to sit and sit to stand.

- Jittering [61], which is to simulate the signal noise that may exist during data capturing. One way to use jittering in our thesis is to create the normal Gaussian noise with three standard deviations of 0.05, 0.1, 0.15 and globally add them to the original data respectively, to create three extra training sets.

- Cropping [61], which is to simulate unexpected data loss. One way to apply cropping in our thesis is to randomly set data at random timesteps for random joint (angles) to 0 with selection probabilities of 5%, 10% and 15% respectively, to create another three training sets.

Note, the three methods do not change the temporal consistency (in the forward or backward direction) of data to a noticeable degree. Therefore, the labels assigned to the samples at their original temporal locations stay unchanged.

In this chapter, we compare these data augmentation methods on the model performance. The method that performs better will then be adopted for the experiments conducted in other chapters.

4.2 Comparison of Vanilla Neural Networks

In this section, we first present the implementation and training details of neural networks involved in this first comparison experiment. Then, we report the results achieved by comparing stacked-LSTM and Dual-stream LSTM networks, which have shown better performances in previous wearable HAR literature, with the other vanilla neural networks as described in Chapter 3.

4.2.1 Implementation Details

To enable a reasonable comparison, given the same training and testing sets, we run a simple grid search on the main hyperparameters (*e.g.,* number of layers, number of hidden units, and kernel size) for each compared neural network.

Here, we take the stacked-LSTM as an example to show the general process. When comparing the number of layers, the number of hidden units in each layer is set to 32 while the number of layers is set to 3 when comparing the number of hidden units. Each LSTM layer is followed by a Dropout layer with a probability of 0.5.

Results of the searching process for the stacked-LSTM are shown in Figure 4.3. As we can see from the figure, increasing the number of network layers (from 3 layers) or hidden units (from 32 units) leads to a decrease in performance, possibly because they introduce more trainable parameters that lead to over-fitting given the limited size of training data.

For the Dual-stream LSTM, three LSTM layers are used in each stream while the number of hidden units of each layer in the movement stream and sEMG stream is set to 24 and 8 respectively, and each LSTM layer is also followed by a Dropout layer with a probability of 0.5. The weights for loss updating of both streams are set to be equal.

All the neural networks used in our experiments employ the Adam optimizer [102] to update the weight, and the learning rate is set to be $1e-3$. For all the neural network methods, the mini-batch size is set to 20. We implement all models with the TensorFlow deep learning library. The hardware used is a PC with Intel i7 8700K CPU and Nvidia RTX 1080 Ti GPU, while the average training time of the stacked-LSTM is around 20ms per iteration/epoch. For comparison, we use

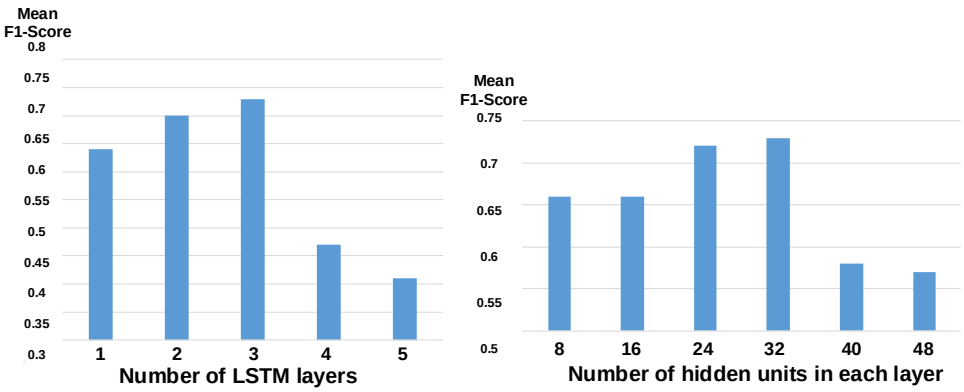


Figure 4.3: Results of the search on the hyperparameters (number of layers and number of hidden units in each layer) of stacked-LSTM.

CNN, convolutional LSTM network (Conv-LSTM), and bidirectional LSTM network (bi-LSTM) [62, 14, 13] described in Chapter 3.

In addition to the above neural network methods, we added to the comparison set the method using Random Forest (RF) as it was used in [12, 29] to model guarding behavior (one category of protective behavior). For the RF method, angle and velocity features (*i.e.,* the range of the joint angles, the mean of joint acceleration value, and the mean of rectified sEMG value) are extracted from the 3s frames, thus we refer to this method as RF-frame in our experiment. It should be noted that, differently from [12, 29], we perform the modeling across different activity types, rather than a model for each activity type. This is critical as only in rehabilitation exercise sessions the activity type is known in advance.

If not mentioned, the default segmentation uses window length of 3s long and 75% overlapping together with zero-padding, while the default augmentation method combines jittering and cropping, as learned from the experiments in section 4.3.1. The number of frames after using such a combination of augmentation methods is increased from ∼3k to ∼21k. Vanilla majority voting described in Chapter 3 are applied to generate the binary ground truth labels (protective vs. non-protective based on a 50% threshold) in this first part of the study for all the compared methods.

4.2.2 Results

The results obtained in the first comparison experiment are reported in Table 4.1. We can see that the stacked-LSTM achieves the best average Macro F1 scores of 0.82, 0.74 in LOSO and LSIO cross validations, respectively, while the Dual-stream LSTM achieves the best average Macro F1 score of 0.74 in LSSO cross validation.

We performed a Friedman test to compare the LOSO results (macro F1 scores) between these methods. The results show statistically significant difference in performances between the methods: $\mathcal{X}^2(5) = 30.474, p < 0.001$. Further, post-hoc Wilcoxon Signed Rank test with Bonferroni corrections (see Table 4.1) show that the stacked-LSTM performs significantly better than the RF-frame ($p = 0.025$) and CNN ($p < 0.001$). It also shows that Dual-stream LSTM, bi-LSTM, and ConvLSTM are not significantly different from stacked-LSTM (at significance level $p = 0.05$). Furthermore, Dual-stream LSTM ($p = 0.006$) and bi-LSTM ($p = 0.032$) are significantly better than CNN. Conv-LSTM does not significantly differ in performance with any of the other methods.

Previous literature on PBD (e.g., [29, 12]) shown the importance of feature selection for the modeling within a specific activity type using the traditional machine learning technique like Random Forest. The performance of such a method dropped to a macro F1 score of 0.67 when modeling within data that comprise different activity types. In addition, a more comprehensive approach using handcrafted feature is seen in [19] that conducted pain level recognition using the same dataset, where interesting performances were only achieved using different sets of

Table 4.1: Comparison Results using the Leave-Some-Subjects-Out (LSSO), Leave-One-Subject-Out (LOSO) and Leave-Some-Instances-Out (LSIO) cross-validation Methods. F_m=Macro F1 score, Re=Recall, Pre=Precision. 95% confidence intervals are added to the LOSO results.

Method	LSSO				LOSO					LSIO			
	Acc	F_m	Re	Pre	Acc	F_m	Re	Pre	p-value against stacked-LSTM (< 0.05)	Acc	F_m	Re	Pre
RF-frame	0.62	0.55	0.57	0.6	0.73±0.095	0.67±0.113	0.67	0.74	0.025	0.59	0.54	0.55	0.56
CNN	0.63	0.54	0.56	0.59	0.78±0.081	0.70±0.081	0.69	0.80	<0.001	0.67	0.61	0.61	0.67
ConvLSTM	0.62	0.61	0.61	0.61	0.81±0.075	0.77±0.100	0.76	0.80	0.172	0.66	0.65	0.67	0.66
bi-LSTM	0.71	0.69	0.69	0.70	0.81±0.055	0.79±0.065	0.79	0.80	>0.05	0.73	0.72	0.73	0.72
Dual-stream LSTM	**0.75**	**0.74**	**0.75**	**0.74**	0.82±0.052	0.80±0.060	0.80	0.79	>0.05	0.73	0.72	0.72	0.72
Stacked-LSTM	0.74	0.73	0.74	0.73	**0.87±0.049**	**0.82±0.069**	**0.83**	**0.81**	-	**0.75**	**0.74**	**0.75**	**0.74**

engineered feature per activity type. Similarly, in [103], the author demonstrates that relevance of feature sets are critically different towards different activity types, thus the method needs to select the proper feature set to function well for each activity type. On the basis of these findings showing that the hand-crafted features are activity-dependent when using standard machine learning techniques (e.g., SVM and RF), our investigation shifted to understanding if the use of deep learning and low-level features would allow for building activity-independent PBD models.

These results suggest that stacked-LSTM does indeed provide overall better performance, and that recurrent models like LSTM-involved networks are better at processing movement and sEMG data for PBD. Interestingly, the Conv-LSTM performs slightly better than CNN, possibly because it is also designed to better capture the temporal information that characterize movement and sEMG data using its recurrent layers with LSTM.

For the 18 folds in LOSO cross-validation where testing subjects are people with CP, we further compute a two-way mixed, absolute agreement, intra-class correlation (ICC) to compare the level of agreement between the ground truth (majority-voted from labels of expert raters) and the output of stacked-LSTM with the level of agreement between the four expert raters. The ICC is a standard method for computing inter-rater agreement [104]. The absolute agreement ICC, which we use, measures strict agreement, rather than the more liberal similarity between rank order of the alternative 'consensus ICC' [105]. A two-way mixed model is used, as the existing raters are the only interested.

We find ICC = 0.215 (single measures) and 0.523 (average measures) with $p = 4.3 \times 10^{-130}$ between the raters, and ICC=0.568 (single measures) and 0.724 (average measures) with $p = 3.1 \times 10^{-159}$ between the stacked-LSTM and the ground truth obtained from the labels of these raters. This finding suggests that stacked-LSTM reaches a moderate level of agreement (with ICC between 0.5 and 0.75 [106]) with the average expert rater on PBD across different types of activity. The agreement is also higher than that between the raters, although this may be explained by the fact that unlike the raters, whose ratings were based on their independent experiences

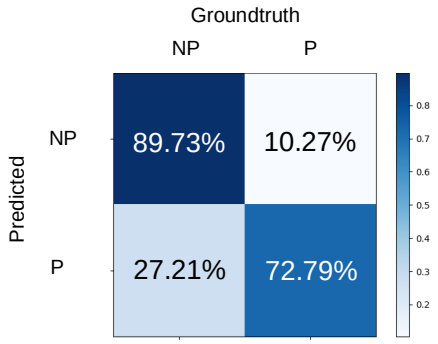


Figure 4.4: A confusion matrix of the performance of stacked-LSTM in LOSO cross validation. NP=non-protective; P=protective.

and background (even if they did have discussions to resolve rating disagreements), the model's training is solely based on the average rater's labelling.

The confusion matrix for the result achieved by stacked-LSTM in LOSO cross-validation is given in Figure 4.4. We can notice from the figure that protective behavior is detected in some healthy participants' data. From an inspection of the recorded videos for these participants as well as checking with previous raters and looking at movement animations of them, we identify various reasons for possible misclassifications: i) some healthy participants were not familiar with the activity or instructions from the experimenter and so hesitated during the execution of the movement (*i.e.,* often looking up at the experimenter while awaiting a confirmation of their execution); and ii) some were not able to conduct specific activities normally like reaching forward due to other physical conditions, *e.g.* obesity, rather than CP.

4.3 Evaluation of Data Preprocessing Methods

The results reported in the previous section show that activity-independent PBD becomes more feasible when using LSTM-based architectures, and can be carried out continuously within each type of activity instances. While neural networks achieve interesting performances in this task, the process of selecting the practical parameters of data preprocessing methods were not explored in more depth yet.

In this section, we analyze three critical aspects of data preprocessing (data augmentation, segmentation, and ground truth definition) to better understand how

they may affect PBD within activity types that build on the ones presented in the EmoPain dataset.

We adopt the stacked-LSTM (3 layers each with 32 hidden units) with the default segmentation (3s long, 75% overlapping and zero-padding) and augmentation (jittering and cropping) methods reported above as the baseline approach while systematically varying each of these methods and parameters.

4.3.1 Comparison of Augmentation Methods

In Chapter 3, we have described three data augmentation methods, namely reversing, jittering, and cropping. Here, we conduct a comparison experiment between them to show the effectiveness and failure of using any of these data augmentation in PBD.

For the augmentation with jittering method alone, to maintain a similar size of the training data with combined augmentation, we create six extra sets of data by applying standard deviations of 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3 to the original training set. We increase the deviation from a smaller value of 0.05 with a maximum of 0.3 as to avoid disturbing data too much.

For the same purpose, to use the augmentation with cropping method alone, selection probabilities of 5%, 10%, 15%, 20%, 25%, 30% are used. It should be noted that the augmentation is only applied to the training data. The testing data remain untouched.

In addition to each of the three augmentation methods, a combination of the jittering and cropping methods is also assessed, as they similarly introduce noises to data without changing its temporal order. Performances are also measured on the original data without augmentation. This led to five test approaches, with the results reported in Table 4.2.

A Friedman test was carried out to compare the five test approaches on the LOSO Macro F1 scores. The results showed significant difference in performance between the these methods ($\mathcal{X}^2(4) = 22.196, p < 0.001$). The p-values from the post-hoc Wilcoxon Signed Rank test with Bonferroni corrections are also reported.

Although with a larger training set than that without augmentation, the augmented training set with reversing method leads to the worst performance and is the

Table 4.2: PBD performances (Mac.F1) and p-values of the post-hoc Wilcoxon Signed Rank test with Bonferroni corrections using the LOSO results under different Data augmentation methods. 95% confidence intervals are added to the LOSO results.

Augmentation method	Training Size	LOSO	LSSO	LSIO	p-value against Jittering + Cropping (< 0.05)
Original	∼3k	0.66±0.110	0.55	0.62	<0.001
Reversing	∼6k	0.40±0.086	0.52	0.53	<0.001
Jittering	∼21k	0.69±0.107	0.63	0.67	0.019
Cropping	∼21k	0.66±0.109	0.68	0.68	0.004
Jittering+cropping	∼21k	0.82±0.069	0.73	0.72	-

only augmentation method (of the four compared) that has lower performance than the baseline without augmentation. This is possibly due to the fact that the reversing method alters the temporal dynamics that characterize how protective behavior is exhibited during an activity. Although all activities included in the dataset are cyclic, *e.g.* 'stand-to-sit vs. sit-to-stand' or 'reach-forward (and returning)', the expression of protective behavior is quite different between such pairs. For instance, in sitting down people with CP tend to bend their trunk at the beginning to reach for the seat for support before descending, whereas in standing up, they avoid bending the trunk due to the fear of pain and mainly push up using their legs and arms.

In comparison with the reversing method, jittering or cropping augmentation do not noticeably affect the temporal order of the data. Further, they may help simulate real-life situations of signal noise and accidental data loss, beneficial for developing a model to be deployed in daily life. Given the similar size of training data, the combination of jittering and cropping methods lead to better performances than using each of them alone. This could be due to the fact that, when each of such augmentation method is used alone to create the same training size as the combined method, the required higher standard deviation in jittering or selection probability in cropping could disturb the training.

4.3.2 Comparison of Padding Methods

In Chapter 3, we have introduced three padding methods, namely zero-padding, last-padding and next-padding. Differently from zero-padding, in the last-padding approach the last sample of the current activity instance is used to pad the window that slides out of the instance; whereas in next-padding, the samples at the following

Table 4.3: PBD performances (Mac.F1) and p-values of the post-hoc Wilcoxon Signed Rank test with Bonferroni corrections using the LOSO results under three padding methods. 95% confidence intervals are added to the LOSO results.

Padding method	LOSO	LSSO	LSIO	p-value against Next-padding (< 0.05)	p-value against zero-padding (< 0.05)
Last-padding	0.72±0.095	0.69	0.66	0.138	0.028
Next-padding	0.79±0.077	0.69	0.66	-	0.478
zero-padding	0.82±0.069	0.73	0.72	0.478	-

temporal positions are used. Here, we compare these different padding methods to understand their impact on the model performance and provide insights that could be helpful to future studies on related movement behaviors detection and PBD.

For the LOSO Macro F1 scores, a Friedman test is carried out to understand if the difference in performance among the three padding methods are statistically significant. Results are summarized in Table 4.3. The results show an effect of padding method on PBD performance ($\mathcal{X}^2(2) = 8.853, p = 0.012$).

Further post-hoc Wilcoxon Signed Rank test with Bonferroni corrections show that last-padding leads to significantly worse performance than zero-padding ($p = 0.028$). This could be because by padding with the last sample, it seems that the subject is maintaining the last position and 'unable' or 'unwilling' to move further, and so appearing as being protective. As zero could be interpreted as a special null value by the model, the zero-padding method may not suffer from this problem.

A competitive performance is achieved with next-padding, with no statistically significant difference to zero-padding. Beyond the tuning of the network hyper-parameters with zero-padding, the slightly lower performance with next-padding could be due to the fact that many CP participants put clear pauses between each activity. The significance of the breaks in padding is that they may seem like freezing behavior. In the context of daily functional activities, we expect that people would be more fluid in their transitions from one activity to another, leading to improved performance with next-padding. However, as such breaks may actually occur in everyday functioning for people with CP as they tend to prepare themselves before starting another activity due to the fear of movement, the last-padding in this context may correctly bias the model toward protective behavior, for the activity prior to

a given break, suggesting that it possibly could also become an adequate method for this case. Nevertheless, when the modeling is conducted on the continuous data stream of varying activity types and transitions, the impact of these padding methods is no longer considered.

4.3.3 Analysis on Sliding-Window Length

For the continuous PBD that we conduct either per activity instance (Chapter 4, Chapter 5) or on the long data sequence of various activity types (Chapter 6), another parameter used in data preparation, *i.e.,* segmentation length, can impact the model performance.

Andreas *et al.* [18] suggested that the window length needs to be adjusted to different types of activity, while the overlapping ratio should be a trade-off between the computation load and the segmentation accuracy. For the task of behavior detection across different activity types, we ask what is the interplay between the segmentation length and the model performance for each activity type, as well as for data comprising all activity types. By looking into this question, we aim to provide insights about how such parameter may be selected for future datasets on similar scenarios. Still, further evaluations would be needed to confirm such understanding as a priori when new PBD related datasets become available.

The boxplots in Figure 4.5 (left) show the distribution of the duration of each activity instance in the EmoPain dataset across different trials, as well as participants. The figure suggests that there are notable differences in duration between activities and even between instances within the same activity, possibly caused by different physical and psychological capabilities of participants. The large variation observed in reach-forward is partially due to differences in capabilities of people to return to the standing position.

We examine the PBD performance with different activity types based on different window lengths in an independent analysis on window length. In addition, we conduct another experiment with all activity types pulled together to better understand the general effect of window lengths on activity-independent PBD performance. The stacked-LSTM (3 LSTM layers each with 32 hidden units) is used together with

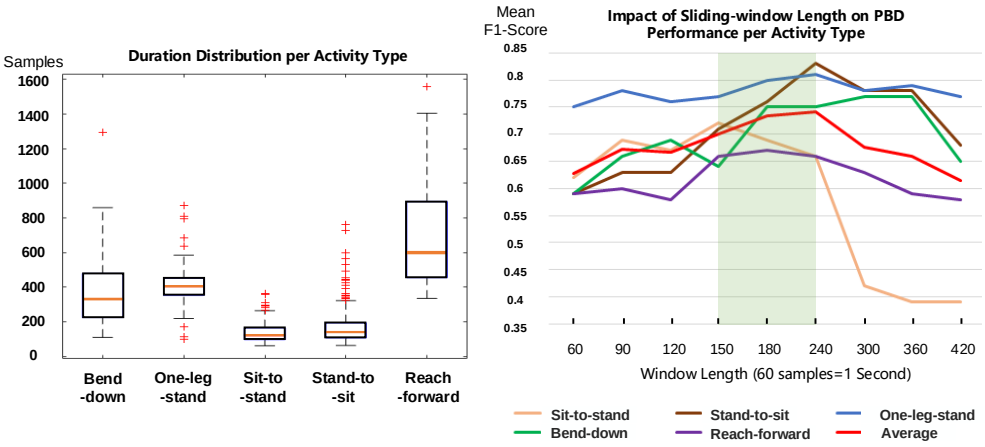


Figure 4.5: (Left): the duration distribution of activity instances in the EmoPain dataset, where 60 samples=1 second. (Right): the impact of sliding-window length on PBD performance per activity type.

our default segmentation (75% overlapping and zero-padding) and augmentation (jittering and cropping) methods.

Impact of Sliding-Window Length on PBD Performance per Activity. For the first set of experiments with a separate model for each activity type, we explore the segmentation with window lengths ranging from 1 to 7s. Considering the size of the dataset and the frame rate of 60Hz, we do not explore larger window lengths.

It should be noted that even though the durations of sit-to-stand and stand-to-sit are similar, we treat them as different activity types. This is because, in real life, they are not generally performed consecutively but interleaved with other activities. In addition, their execution is not the reverse of each other, especially in people with CP as discussed earlier.

The Macro F1 score for each window length are plotted in Figure 4.5 (right) for each activity type, with a red line showing the average performance computed over the five activity types per window length.

A repeated-measure ANOVA is run to understand the effect of the nine window lengths (independent variables) and five activity types (independent variables) on PBD performance (Macro F1 scores, the dependent variable) based on the folds of LSSO cross-validation. The results show an effect of window length ($F =$

$5.212, p = 0.001, \mu^2 = 0.173$) and of window length and activity type interaction ($F = 3.188, p = 0.01, \mu^2 = 0.338$) on PBD performance.

Post-hoc t-test shows that the window lengths in the range from 2.5s to 4s show significantly better F1 scores ($p < 0.05$) than other lengths outside this range, except for 5s. However, the detection at 5s only shows significant difference with 7s ($p = 0.01$), and is approaching significantly lower performance than 4s ($p = 0.056$). The post-hoc t-test for the interaction between window length and activity type is not statistically significant, possibly due to the limited points for each activity (in each of the 6 validation folds); still, a few observations should be made from these results on the basis of Figure 4.5.

- Although both stand-to-sit and sit-to-stand have a short duration, their detection performances differ when given window lengths more than 2.5s, with sit-to-stand reaching the best performance at 2.5s and stand-to-sit reaching the highest performance at 4s. Such differences could be due to the zero-padding used in this study: padding with zeros given larger window lengths may improve or at least maintain the detection of non-protective behavior for stand-to-sit, as a person generally feels safe after reaching the chair and then relaxes; however, when a person stands up from a chair, the protective behavior (*e.g.*, guarding) often persists at the standing posture due to the absence of support (muscle tension remains despite no need and trunk remains slightly flex [19]), therefore zero-padding at the activity completion could conflict with the interpretation of such behavior.

- Despite the fact that the best performance for one-leg-stand is at window length of 4s, this activity is less affected by different window lengths. This could be explained by the fact that while this activity is transient (consisting of simply raising and lowering the leg), it is also sustained because the participant tends to hold the position (possibly oscillating the leg up and down); as a result, performance remains high across short and long windows. In a real situation, leg raises (or balancing on one leg) happens during walking or climbing stairs, thus such short events could be more of interest.

- Detection on bend-down and reach-forward instead benefits from longer window lengths, possibly because the bending movement that characterizes them is common to many other activities (*e.g.,* CP participants tend to bend the trunk first in sitting down to search for support and normal standing up involves a bend as well to facilitate) and so the system needs more information to know how to interpret bending movement correctly.

Given the analysis above, we shortlist window lengths of 2.5s, 3s and 4s for the activity-independent PBD exploration reported in the next.

Impact of Sliding-Window Length on PBD Performance across Activities. With all the activity instances pulled together for training and testing, we conduct LOSO experiments with the three window lengths (2.5s, 3s and 4s) summarized from the previous experiment.

The results are reported in Table 4.4. A high performance is achieved for all three window lengths (independent variables), but a Friedman test shows statistically significant difference in performance (LOSO Macro F1 scores) between the three window lengths: $\mathcal{X}^2(2) = 8.914, p = 0.012$. Post-hoc Wilcoxon Signed Rank test with Bonferroni corrections on the Macro F1 scores show that the 3s window leads to significantly better performances than the window of 4 seconds ($p = 0.017$) but its performance shows only marginal significance in comparison with the 2.5s window ($p = 0.093$). No statistical differences were found between the performances achieved with the 4s and 2.5s windows.

Looking further at the results (Macro F1 scores) across the 30 subjects, presented in Figure 4.6 (number 1 to 12 represent healthy participants, 13 to 30 represent CP

Table 4.4: PBD performances (Mac.F1) under three sliding-window lengths across all activities. 95% confidence intervals are added to the LOSO results.

Validation Method	Activity Type	2.5s	3s	4s
LSSO	Bend-down	0.64	0.75	0.75
	One-leg-stand	0.77	0.8	0.81
	Sit-to-stand	0.72	0.69	0.66
	Stand-to-sit	0.71	0.76	0.83
	Reach-forward	0.66	0.67	0.67
LOSO	All activities	0.78±0.084	0.82±0.069	0.73±0.077

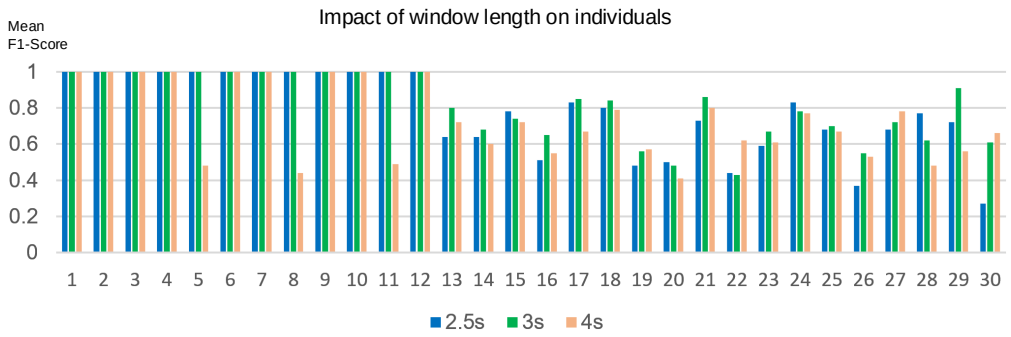


Figure 4.6: Impact of sliding-window length on different subjects. 1-12: healthy participants, 13-30: CP participants.

participants), we can notice some effects of window length.

- The detection performances on most control subjects are 100% accurate across all three window lengths; this could be due to an imbalanced distribution in the training set in which non-protective data take a larger proportion; and the protective movements exhibited by participants with CP suffer more from the padding effect introduced by changing window lengths.

- The model results for people with CP vary with window lengths without a clear pattern, particularly for subjects 13, 16, 17, 22, 26, 28, 29, and 30; this emphasizes the impacts of individual variations on the temporal properties of the data, as shown by the boxplots in Figure 4.5 (left); this may be related to the wide range of protective movement strategies and length of each activity among patients with CP. This may suggest that personalized training may reach better results when sufficient data per person are available.

Overall, the statistical analyses we conduct in the two sets of experiment reported above suggest the following.

- Longer window lengths (>2s) are desirable for activity-independent PBD (at a frame rate of 60Hz), suggesting that the window shall contain sufficient information to differentiate between movements required to conduct an activity and movements associated with protective behavior.

- Window lengths longer than the duration of most types of activities suffer from the padding effect and reduction in number of frames, resulting in worse performance.

Given the representativeness of the EmoPain dataset in terms of movements that it contains and the variability of participants with CP it covers, we expect that principles learned from our study would also apply to other datasets that involve data building on the five basic activities in this study. Naturally, further studies may repeat our experiments to confirm this, when new datasets become available.

4.4 Summary

This chapter investigated the possibility of using deep learning to improve PBD across activity types and continuously within each activity instance by using IMUs and sEMG data. In our approach to addressing this problem, we explored both convolutional and recurrent neural networks, and a traditional approach with RF. In summary, the best detection result was obtained with a stacked-LSTM network, with accuracy and Macro F1 score of 0.87 and 0.82 respectively in LOSO cross-validation.

Analyses on the parameters relevant to our approach were conducted to understand how they affect PBD and could inform PBD in future datasets.

First, we evaluated different approaches to padding in the segmentation of data streams. The results suggest that it is valuable to use a method that does not introduce confounding behavior (*i.e.,* data that could be interpreted as protective behavior) in creating the data frames. In our case, the best method was the zero-padding (the other two we explored were the Last-padding and the Next-padding), and the second best was the Next-padding, suggesting that PBD could also work in full continuous detection without pre-segmentation of activity. However, the zero padding may have its own limitations when protective behavior occurs at the beginning or at the end of an activity. Hence, in the case of activity-dependent modeling, one should consider if the activity type is likely to generate protective behavior in preparation for the exercise or at completion of the movement.

Second, we also compared different data augmentation methods. Our findings suggest that it is important to avoid the use of augmentation methods that noticeably

affect the temporal order of data in a frame. In our experiments, the reversing augmentation method (which we compared with jittering and cropping methods as well as no augmentation at all) may alter the temporal dynamics that characterize how protective behavior is presented during an activity, leading to worse performance than when no augmentation was done.

Third, we explored the effect of the window length used for data segmentation, and we found that the PBD performance generally increased with growth of the window length until a certain peak, beyond which performance appeared to drop. This could be due to the fact that shorter lengths provide insufficient information to understand the movement dynamics and hence distinguish protective behavior from normal expected movements. Meanwhile, larger window lengths may suffer because there is more padding, relative to data present in the windows. Although we found the optimal window length to vary with activity type, our findings suggest that good performances across activity types can be achieved using any window length within a small range of values of 2.5s to 4s, based on our setting. Our statistical analysis results and observations on PBD suggest that the specific range will depend on both the amount of diversity of targeted activities (rather than the specific dataset used) and the duration of each of these activities.

These three sets of insights that emerged from our work in this chapter, based on the EmoPain dataset (and so representative in terms of everyday activities, protective behavior, and the CP population), contribute a set of criteria to select possible optimal parameter settings for future PBD datasets. Naturally, we acknowledge that further testing on other datasets would be necessary to fully verify these findings.

The work presented in this chapter was carried out in 2018-19 and published first in the conference of Ubicomp/ISWC'19 [9] with its extended version published in ACM HEALTH [8]. At the point of completing this thesis, the work presented in this chapter has received 19 citations (excluding the self-reference by my own works). Five of them are review papers taking our work as example to show advances in pain-related technology [107, 108], behavior sensing from body movement [109], wearable research [110], and healthcare management [111]. Li *et al.* [112] built

on our work by proposing a network comprising LSTM layers as the encoder and dense layers as decoder for pain intensity estimation as well as PBD. With a hold-out validation setting, they achieved improved activity-independent PBD performance than the method of using stacked-LSTM alone (mac.F1 of 0.93 vs. 0,92). Yuan *et al.* [113] proposed to use LSTM layers to build an autoencoder structure and connect it with an attention learning module for PBD, achieving improved performances (mac.F1 of 0.59 vs. 0.48) as well. To alleviate the manual efforts made in data partitioning, *e.g.,* pre-segmentation of activity instances in our case, [114] proposed an unsupervised method to incorporate maximum mean discrepancy for the learning of sample (dis)similarities for automatic time-series partitioning.

Chapter 5

Capturing Variety with Attention to Improve Performance

The study presented in the previous chapter provides us with a set of criteria on how sequential data could be better augmented and segmented to enable the training of deep learning models for activity-independent protective behavior detection (PBD). In this chapter, under a similar experimental setting, we investigate how to further improve activity-independent PBD in pre-segmented activity instances. To do so, we look into the definition of pain behaviors [21, 23, 30, 31] and descriptions provided by physiotherapists in [19] that we discussed in the background chapter.

The pain literature [21, 23, 30, 31] provide evidence that fears of injury, pain, and anxiety in chronic pain (CP) cause the individual to engage bodily parts in ways that are not biomechanically necessary or efficient, but may create a sensation of control and assist to alleviate fear. People with CP break their movement in parts, shifting their attention to one part of the body at the time to increase the sense of control. They also recruit specific body parts to help avoid or minimize the use of the ones perceived at risk. From [19] we learned that, in designing interventions to improve movement-related self-efficacy of people with CP, expert observers point out how specific body parts are particularly important to detect the presence or absence of protective behavior.

So far, such attention information was not leveraged in PBD. By applying vanilla neural networks globally on the concatenated multidimensional data, the

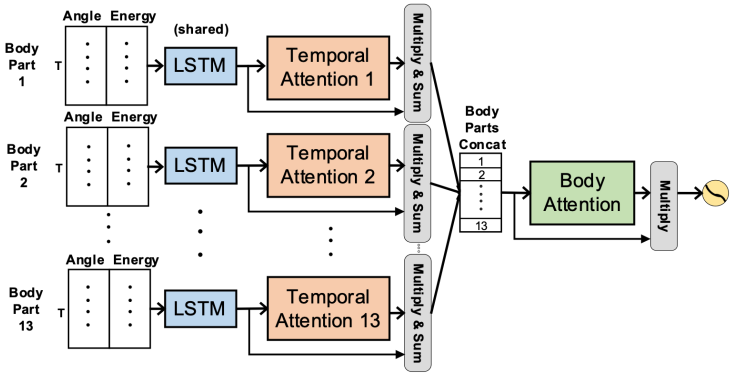
ability of a model to learn local movement dynamics at different body parts is limited. This may hinder the model performance, given that the informative local movement cues are not well learned and redundant information of the idiosyncratic variability in the movement of less relevant body parts may introduce noise. Furthermore, studies in HAR have seen the success of using attention mechanism for achieving better performances and providing insights about the input data, *i.e.*, suggesting the data collected from the specific part of the body provide more informative clues for model's decision-making. As a result, for PBD, this chapter explores how to use attention mechanism to guide the network design to improve the performance as well as to provide data-driven evidences about the movement pattern of people with CP that were only seen in previous qualitative pain literature. Additionally, it leads to develop a new intervention as the body-part attention-driven sonification to help people increase awareness of their use of protective behavior [115].

We propose that for activity-independent PBD, both temporal and bodily attention mechanisms could be useful to capture the attention shifting observed by physiotherapists when describing protective behavior in people with CP. The contribution made in this chapter is as follows.

- We propose an end-to-end neural network architecture called Body Attention Network (BANet). BANet is able to self-learn when (temporal attention) and what (bodily attention) subsets of the anatomical joints contribute most to the detection of protective behavior. Here, we focus only on MoCap data comprising streams of joint angle, but the network can be easily adapted to data of joint positions or even data collected from multiple sensor types (*e.g.*, MoCap plus EMG data).

- Through a range of experiments on the EmoPain dataset, we demonstrate that our method can achieve state-of-the-art results, if not slightly higher, with fewer trainable parameters for the detection of protective behavior.

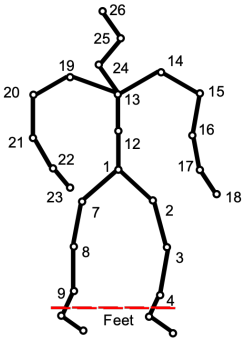
- With visualization and statistical analysis of both temporal and bodily attention weights (learned model weights), we discuss how such mechanisms could capture the dynamic piecemeal breaking of movement expected in people with CP.



(a)

Angle Description

Angle Name	Respective Anatomical points
1 Crown-Hip-LeftFoot	26-1-4
2 Crown-Hip-RightFoot	26-1-9
3 Spine-Hip-LeftLeg	12-1-3
4 Spine-Hip-RightLeg	12-1-8
5 LeftKnee	2-3-4
6 RightKnee	7-8-9
7 LeftElbow	15-16-17
8 RightElbow	20-21-22
9 LeftShoulder	24-14-15
10 Rightshoulder	24-19-20
11 Leftshoulder-LeftUpperLeg-LeftLowerLeg	16-14-2
12 RightShoulder-RightUpperLeg-RightLowerLeg	21-19-7
13 Neck	12-24-26



(b)

Figure 5.1: (a) Overview of the BANet, where each body part is described by the joint angle plus energy features. (b) The 13 joint angles that used as the input for BANet, where data collected from the participants' feet are noisy and hence not used in this work.

5.1 The Body Attention Network

In this section, we first present the BANet architecture. Then we describe the attention mechanisms designed considering the characteristic of protective behavior and targeting the issues we found in the previous HAR literature.

An overview of BANet and 13 joint angles are shown in Figure 5.1. The input to the BANet is a 2×13 low-level movement matrix (comprising angles and energies from 13 body parts), for each sample/timestep in a movement frame. A shared vanilla LSTM subnetwork is used to extract the temporal information separately from each of the 13 body parts, *i.e.,* given a data frame, the output of such temporal

decoder is a matrix of hidden states $\mathbf{H}_T = [h_1^{c,k}, ..., h_t^{c,k}]$ with $h_t^{c,k} = [h_t^{1,k}, ..., h_t^{C,k}]$, where $c \in \{1, 2, ..., 13\}$ for the 13 body parts, $t = 1, 2, ...T$ for the data frame with temporal length of T, and $k = 1, 2, ..., K$ for the number of hidden units used in the temporal encoder.

5.1.1 Temporal and Bodily Attention Learning

The attention mechanism of BANet is implemented with two stages: a temporal attention module is placed first, separately for each body part, followed at a higher level of the model by a global bodily attention module.

Unlike the attention-based architecture seen above for wearable HAR, we propose to put the processing backbone, the LSTM subnetwork, at the beginning of the model to provide the two attention mechanisms a higher-level knowledge about the temporal dynamics of the movement of each body part given a certain duration. We also propose to learn the temporal saliency of the movement of each body before we go to understanding the importance of different body parts. In this subsection, we describe in detail the two attention modules below.

Temporal Attention Module. To learn the temporal attention weight a_t^C for $\mathbf{H}_t^{C,K}$ across all the timesteps t of the current input frame, we use a 1×1 convolutional layer and a softmax layer as

$$a_t^C = \text{softmax}(\mathbf{W}_a * \mathbf{H}_t^{C,K}), \tag{5.1}$$

with

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{i=1}^{N} \exp(x_i)}, \tag{5.2}$$

where $\mathbf{W}_a$ is a learnable weight matrix, and $*$ is the convolution operation. The computation for temporal attention is illustrated in Figure 5.2 (above).

Unlike the fully-connected layer, the 1×1 convolution layer acts as a linear embedding which limits irrelevant connections among the input matrix (in our case is the dense connection of samples within a frame). Thus, the 1×1 convolution layer can help minimize the number of trainable parameters. We would further experiment

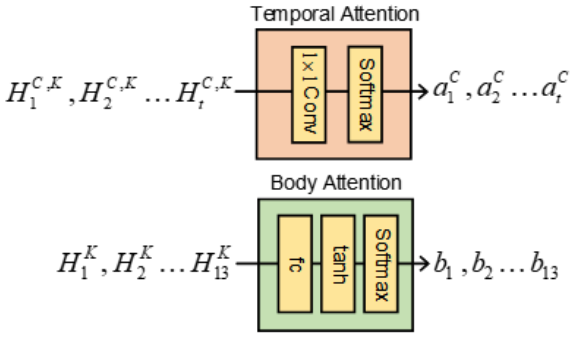


Figure 5.2: The temporal attention block (above) and the bodily attention block (below) that we used in the proposed BANet.

with a variant using fully connected layer for temporal attention computation to justify this.

The temporal attention module further includes a merge of the attention weights with the original output of the temporal decoder at each body part through a multiplication followed by a sum-up over samples as

$$\mathbf{H}_C^K = \sum_{t=1}^{T} a_t^C H_t^{C,K}. \tag{5.3}$$

The output of the temporal attention module is a matrix of the weighted-sum of the temporal information from each of the 13 body parts, which can be written as $\mathbf{H}_c^k = \{[h_1^1, ..., h_1^K], ..., [h_{13}^1, ..., h_{13}^K]\}$.

Bodily Attention Module. So far, the network has processed the data frame separately per body part, where the informative temporal subset of the movement of each body part is learned.

To learn the subset of the body parts that play a key role in the detection of protective behavior during a given movement segment, here we describe the bodily attention module. In this module, the attention to the body parts is computed upon the knowledge across the whole body, in which way each body part is considered in the context with the other body parts to learn what is important at that segment of time. We use two fully-connected layers with tanh and softmax activations to compute the bodily attention weight b_c as

$$b_c = \text{softmax}(\tanh(\mathbf{W}_\beta \mathbf{H}_c^K)), \tag{5.4}$$

where $\mathbf{W}_\beta$ is a learnable weight matrix. The bodily attention module is completed by merging the bodily attention weight with the original output of the temporal attention module as.

$$\text{atten}\mathbf{H}_C^K = b_c \odot \mathbf{H}_C^K. \tag{5.5}$$

Such attention-over-attention structure of BANet finally produces a $K \times 13$ matrix atten$\mathbf{H}_C^K$ which encodes the importance of each body parts at important moments (samples) for the input segment. With such output, the detection is finally completed with a fully-connected layer using softmax activation.

5.2 Experiment Setup

In this section, we briefly present the data preparations and experimental settings.

5.2.1 Data Preparation

Unlike the study we present in the last chapter, here we only use the movement data (and not the sEMG data) of the EmoPain dataset. That is, we want to understand how the model learns the movement cues that are relevant and compare it with how physiotherapists had described the protective movement behaviors.

In total, there are 46 activity instances, where each instance is around 10 minutes long and contains sequences of sit-to-stand, stand-to-sit, bending, reaching forward and one-leg-stand activities. Following the low-level features described in Chapter 3, each sample is characterized by 13 joint angles, as well as the energies of these. The energy is the square of the respective angular velocities.

To create the training and test data for the experiments, we adopt the segmentation with a sliding window length of 3 seconds and overlapping ratio of 75% within each activity type in the movement data. 0-padding is used when the window slides beyond the end of a given activity type. This amounts to a total of 2,569 frames. The groundtruth for each frame is set based on majority-voting, where a frame is labelled as protective if at least 50% of the samples within it had been rated as protective by

each of at least 2 out of the 4 raters, and non-protective otherwise.

For the training of BANet and of other architectures adopted for comparison, we apply two augmentation approaches, namely jittering and cropping, as also mentioned in Chapter 3. The use of the two approaches leads to 18,653 segments, where 11,373 segments are labelled as non-protective (from both healthy participants and participants with CP) and 7,280 segments are labelled as protective (only from participants with CP).

5.2.2 Implementation Details

The BANet is implemented with TensorFlow deep learning library. For the LSTM subnetwork acting as the temporal encoder, we use a 3-layer LSTM network with 8 hidden units in each layer. Dropout with probability of 0.5 is used after each LSTM layer. For the full architecture, weights are updated with Adam optimizer, with a learning rate of 0.003 and batch size of 40. The validation method used in this study is the standard leave-one-subject-out (LOSO) cross-validation across the 30 subjects. We report the Macro F1 score with 95% confidence interval as the metric. Statistical tests, particularly Friedman test and post-hoc Wilcoxon Signed Rank tests with Bonferroni corrections, are used to compare the performances of different architectures.

We compare BANet with vanilla neural networks described in the earlier chapters: i) Convolutional LSTM (Conv-LSTM), with convolution kernel size of 1×10, max pooling size of 1×2, 10 filters, 28 LSTM hidden units and batch size of 50; ii) Bi-directional LSTM (bi-LSTM), with 14 LSTM hidden units followed by a Dropout with probability of 0.5, and batch size of 40; iii) stacked-LSTM, a vanilla 3-layer LSTM network with each layer of 28 hidden units followed by Dropout with probability of 0.5, and the batch size is set to 20. For all the neural networks, the Adam optimizer is used with a learning rate of 0.003.

We also compare it with several variants of the current BANet architecture to better understand how the two attention mechanisms, in terms of their order and mode of use, affect PBD performance.

First, we create a variant of the BANet with a fully-connected layer used in the

temporal attention computation (referred to as BANet-dense) replacing the default 1×1 convolution layer.

In addition, we compare with the variant (referred to as BANet-compat, for BANet compatibility version) where the computation of bodily attention is done at an input level instead of at feature fusion level with the same attention algorithms presented in the last section. Specifically, for BANet-compat, at each timestep, the bodily attention weights were computed for the 13 body parts. After multiplication with the original data per timestep, all the timesteps are concatenated for the temporal information extraction and temporal attention computation as stated above. Thereon, the output to be classified has the same size of k×13 as the BANet (k is the number of hidden units of the LSTM encoder).

Finally, to show the impact of the two attention modules that we use together, we provide the results of BANet-body where only the bodily attention is implemented at the input level, and BANet-time where only the temporal attention is computed.

5.3 Result

Results for the comparison experiments are shown in Table 5.1. As we can see, the proposed BANet achieves the best results (accuracy of 0.87, Macro F1-score of 0.84), with a smaller parameter size of 8,131 in comparison to other tested LSTM-based architectures (parameter size ranging from 14,000 to 40,000).

The parameter reduction is obtained in BANet through the use of: i) the temporal information extraction strategy targeting each body part, which processes data of a smaller dimension and allows the respective shared LSTM layer to have smaller number of hidden units; ii) a 1×1 convolution layer instead of fully-connected layer for computing the temporal attention, with the former being a critical advantage due to the many timesteps (180 timesteps) of the input to this layer.

The second best is achieved with BANet-body which shows the importance of focusing on a subset of joint angles (rather than all) for the detection of protective behavior. Instead, the BANet-time that only learns the temporal attention separately for each joint angle does not achieve high accuracy results. This is expected and is

Table 5.1: Results (Mac.F1 with 95% confidence intervals) and p-values of the post-hoc Wilcoxon Signed Rank test with Bonferroni corrections using LOSO results of the comparison experiment. The method of the best macro f1 score is in bold.

Methods	Accuracy	Macro F1	p-value against BANet (< 0.05)	Parameter size
Conv-LSTM	0.81±0.087	0.74±0.105	0.027	40,940
bi-LSTM	0.85±0.058	0.80±0.074	0.175	14,282
stacked-LSTM	0.86±0.055	0.81±0.072	0.22	18,986
BANet-compat	0.66±0.125	0.57±0.137	<0.001	12,204
BANet-dense	0.82±0.065	0.79±0.074	0.058	71,430
BANet-time	0.81±0.069	0.76±0.085	<0.001	7,767
BANet-body	0.87±0.050	0.83±0.070	0.167	8,023
BANet	**0.87±0.049**	**0.84±0.065**	**-**	**8,131**

due to the lack, in this network, of global processing over all body parts.

The next best result is achieved by the stacked-LSTM (accuracy of 0.8534, Macro F1-score of 0.812). Although the result is very similar to BANet's (see also their confusion matrices in Table 5.2), stacked-LSTM requires a larger number of parameters (18,986).

On the other hand, except for that the BANet-compat is only a representative of the architectures used in [15, 16, 17], the results imply that encoding the importance of body joints at a single timestep is not valuable to the detection of protective behavior, but should be delayed to a higher-level processing stage using data input of a certain temporal length.

Comparison of BANet with vanilla LSTM-based variants. Here, we first verify if our BANet performs statistically better than the other three vanilla LSTM-based variants. The Friedman test shows statistically significant differences in the perfor-

Table 5.2: The confusion matrices for BANet and stacked-LSTM.

BANet		Non-protective	Protective
Groundtruth	Non-protective	1491 (92.84%)	115 (7.16%)
	Protective	331 (31.83%)	709 (68.17%)
stacked-LSTM		Non-protective	Protective
Groundtruth	Non-protective	1451 (90.35%)	155 (9.65%)
	Protective	322 (30.96%)	718 (69.04%)

mances across LOSO folds ($\mathcal{X}^2(3) = 8.544, p = 0.036$). Post-hoc Wilcoxon Signed Rank tests with Bonferroni corrections show that BANet is only statistically significantly better than Conv-LSTM, which yet has the largest parameter size of 40,940 than all other compared vanilla neural networks. Although the significance does not hold in comparison with bi-LSTM and stacked-LSTM, our BANet has noticeably smaller parameter size, more practical for real-life deployment.

Furthermore, the availability of temporal and body weights allow developing new types of intervention. For example, the attention weights produced by BANet were used to drive a new body-movement sonification approach to help people with CP become aware of their protective behavior and capabilities [115].

Comparison of BANet variants. We then explore if our BANet performs statistically better than its different variants. The Friedman test shows statistically significant differences in the performances across LOSO folds ($\mathcal{X}^2(4) = 56.109, p < 0.001$). Post-hoc Wilcoxon Signed Rank tests with Bonferroni corrections show that significance does not hold with respect to the BANet-body (p=0.167) and BANet-dense (p=0.058). However, BANet-dense uses a pretty large parameter size of 71,430 than BANet. When the BANet-body itself is compared with the other architectures, significant differences are found with BANet-time ($p < 0.001$) and BANet-compat ($p < 0.001$). This suggests that the impact introduced by the bodily attention module is more significant than temporal attention.

5.3.1 Analysis on Attention Weights

In this section, we analyze trends in attention weights of BANet to understand to what extent the two attention mechanisms capture aspects of protective movement strategies highlighted in the pain behavior literature [21, 23, 30, 31]. Besides improving model performance, the weights computed by the attention modules toward the input of 13 local joint angles may help verify if the BANet learns information that reflect physiotherapists' description of protective behavior.

Analysis of Bodily Attention Weights. Figure 5.3 shows boxplots of the distributions of bodily attention weights learned from test segments over all the 30 folds of

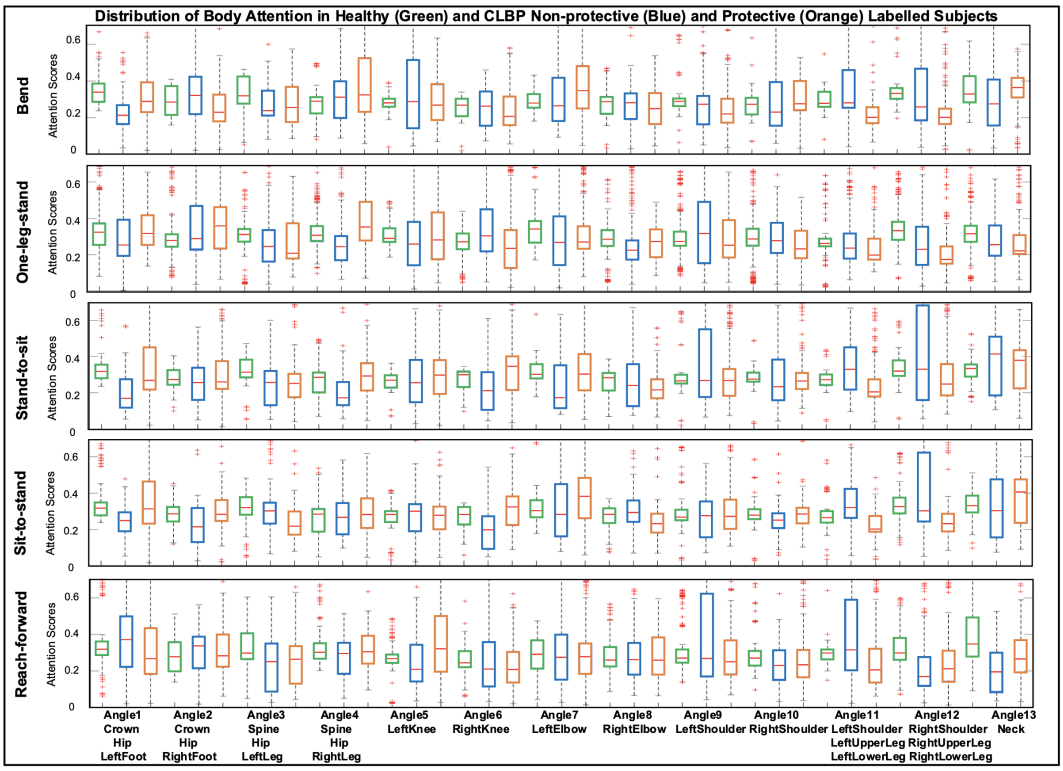


Figure 5.3: Boxplots for the distribution of bodily attention weights computed by BANet for each testing data of a joint angle, organized by activity type.

participants per joint angle, organized by activity type. It is interesting to see that the boxplots for healthy participants (green) are quite narrow compared to those of participants with CP (blue and orange). The boxplot ranges of the healthy participants for most body parts is centered around a medium value. This suggests that data of the healthy participants collected from most body parts tend to receive similar attention within a certain type of activity. By contrast, such is not the case for the boxplots of the patients, either for those showing protective or non-protective behaviors. This reflects findings in previous pain literature, pointing to the larger variability between people with CP in terms of movement strategies (beyond idiosyncrasy) adopted to perform a particular activity.

We then combine the weights of CP participants showing protective and non-protective behaviors, and conduct an independent t-test (two-sample t-test) analysis to compare the size of the boxplots (*i.e.*, using the difference between the 0.75 and 0.25 quantiles of each box as dependent variable) between the healthy and CP

Table 5.3: Results of the independent t-test for comparing the size of boxplots between the healthy and CP participants (showing protective or non-protective behaviors). DF denotes the degree of freedom.

Activity Type	Healthy vs. CP				
	Healthy Mean	CP Mean	t-value	DF	p-value< 0.01
Bend	0.0237	0.0456	-6.0356	24	3.1216e-6
One Leg Stand	0.0214	0.0469	-7.1571	24	2.1353e-7
Reach Forward	0.027	0.0485	-5.1580	24	2.7867e-5
Sit to Stand	0.0206	0.0415	-5.4574	24	1.3115e-5
Stand to Sit	0.019	0.0447	-7.2632	24	1.6702e-7

participants (with participant type as independent variable) across all body parts (represented by the respective joint angle). Such analysis is repeated for each activity type. The results show that the box sizes of the participants with CP (showing normal or protective behavior) are significantly larger than that of the healthy participants (see Table 5.3 for p-values per activity type). One should consider that, given the number of comparison (5 activities) performed, a Bonferroni correction should be considered ($p < \alpha/5$ for significance).

Analysis of Temporal Attention Weights. Figure 5.4 shows heatmaps of the temporal attention weights per joint angle for one healthy participant and one participant with CP for the activity of stand-to-sit. As shown, the temporal attention paid to different body parts (represented by respective joint angles) of the healthy participant look more homogeneous than that for the participant with CP. We further create the heatmaps of each participant per each activity type, as shown in Figure 5.5. In general, a higher homogeneity of temporal attention weight assigned to each body part across time is seen in healthy participants for all the five activity types but not in participants with CP.

To enable a statistical analysis of the maps and a comparison between healthy and CP participants' temporal attention variations, we first apply min-max normalization to the temporal attention weights computed for each body part of each participant. Then, we compute the entropy for temporal attention weights at each body part over time for each participant to represent the level of variation. The independent t-test shows that the temporal attention paid to each body part of participants with CP is significantly less homogenous (with higher entropy) than that of

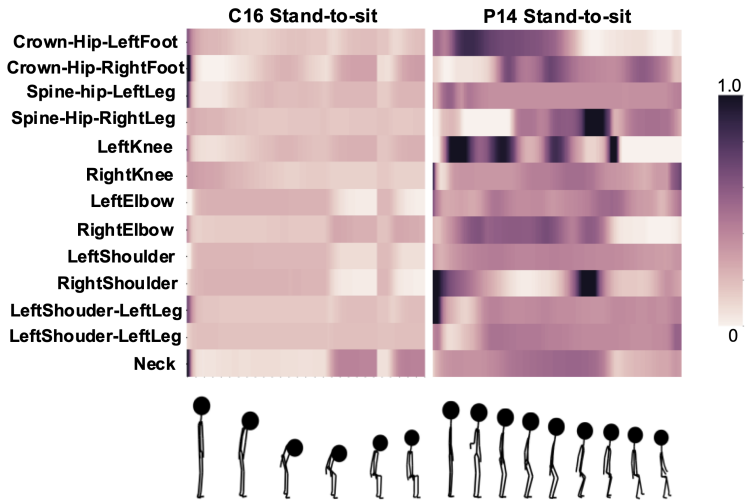


Figure 5.4: Heatmaps of the temporal attention weights computed in BANet for testing instances of healthy subject number 16 and patient number 14 with their respective movement data (stick figures).

the healthy participants (see Table 5.4 for p-values for each activity). It should be noted that, given the number of comparison (5 activities) performed, a Bonferroni correction should be considered ($p < \alpha/5$ for significance).

The above observations and statistical analyses on bodily and temporal attention weights can be related back to the pain literature in two ways.

First, even when people were not asked to perform a movement according to ideal trajectories, healthy subjects tend to perform simple everyday movements in a quite similar way [116] as suggested by the size of the boxplots of the bodily attention weights. Only a few healthy participants' boxplots are slightly wider, especially in bend and reach-forward. This could be because these two activities are biomechanically demanding, and (as revealed in the previous chapter) some healthy

Table 5.4: Results of the independent t-test for comparing the entropy of temporal attention weights between the healthy and CP participants (showing protective or non-protective behaviors). DF denotes the degree of freedom.

Activity Type	Healthy vs. CP				
	Healthy Mean	CP Mean	t-value	DF	p-value< 0.01
Bend	179.3564	230.7798	-7.9902	24	3.2248e-8
One Leg Stand	700.0913	906.2388	-3.6921	24	0.0011
Reach Forward	373.0753	411.4715	-3.0893	24	0.0050
Sit to Stand	192.0729	229.1187	-3.3378	24	0.0027
Stand to Sit	130.0677	290.2926	-38.2443	24	5.0895e-23

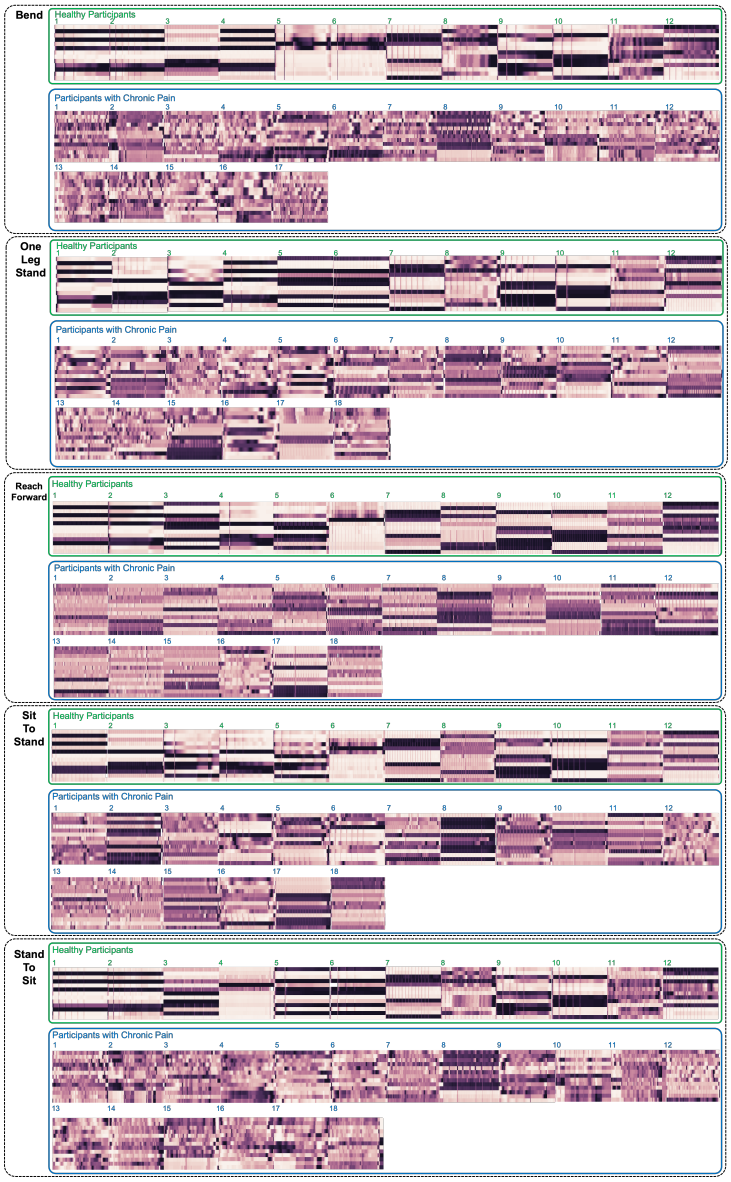


Figure 5.5: Heatmaps of the temporal attention weights computed in BANet for each participant, organized by activity type (zoom in for better reading).

people hesitated in performing these as they were not sure of the instructions (upon an analysis of the original on-site videos).

The wider boxplots of the people with CP instead reflect the literature on CP [21, 23, 30, 31, 19], suggesting larger variety in movement strategies according to how their physical and psychological capabilities affect what the person perceives as safe or dangerous or what part of the body is perceived as more vulnerable. For instance, in stand-to-sit, a person may reduce weight on the legs by twisting the trunk

and use the chair as a support, such as in Figure 5.4-P14.

Second, the limited width of the healthy participants' boxplots together with their more homogeneous temporal weights across body parts and time suggest that the BANet considers the different body parts of more-or-less similarly important in the detection process. This is in line with the pain literature suggesting that healthy people have a synergetic way of using different body parts when performing the movement [12]. For example, see the temporal attention weights of healthy participant number 16 in the stand-to-sit (Figure 5.4).

However, this homogeneity in body and time is not seen for CP participants. As discussed in [38, 19], people suffering from CP tend to engage different body parts at different stages of the movement rather than in a synergetic way, despite making the movement more difficult to execute. This self-induced difficulty is also often the cause for perceived poor self-efficacy and for increase of pain.

For example, let us analyze in more detail P14's darker (and so higher) temporal attention weights during stand-to-sit (Figure 5.4 (b)). P14's engagement of the leg and shoulder at the initiation of the sit-down suggests hesitation (as indicated by physiotherapists in [12, 19]). We know from video analysis that this initial hesitation is followed by a horizontal twist of the shoulder (which is captured by the right shoulder's score) followed by the bending of the neck to check for the chair position, then still the twisting of the shoulder (captured by the left shoulder score) to use the arm (left elbow bent beside the trunk) for support on the chair to minimize the load on trunk and on legs. The healthy participant C16 also uses the arms, but behind the body (rather than on the side) to reach for the chair together with the trunk, which are not used as support for legs or back.

5.3.2 Extra Evaluation of BANet on HAR Datasets

Given the interesting results obtained with BANet for PBD, another work [20] has shown the generalizability of BANet in the context of wearable HAR. They tested our proposed BANet against previous state-of-the-art methods, including attenLSTM [15], DeepSense [73], SADeepSense [117], and their proposed GlobalFusion method, on several wearable HAR benchmark datasets. Unlike BANet that only considers the

Table 5.5: The performances (macro F1 scores with 95% confidence intervals) of BANet and previous state-of-the-art methods reported in [20], using several wearable HAR and abnormal behavior detection datasets.

Method	Dataset		
	RealWorld-HAR	DSADS	DG
DeepSense	0.73±0.081	0.86±0.047	0.64±0.106
SADeepSense	0.73±0.061	0.86±0.035	0.60±0.051
attnLSTM	0.69±0.106	0.87±0.064	0.58±0.046
GlobalFusion	0.84±0.055	0.94±0.035	0.69±0.094
BANet (ours)	0.75±0.066	0.87±0.049	0.62±0.106

attention learned at different body parts and temporal segments, GlobalFusion looks into the sensing quality of different modalities, *e.g.*, accelerometer vs. gyroscope.

On the three wearable HAR and abnormal behavior detection datasets used in their study, namely RealWorld-HAR [118], DSADS [119], and DG [71], according to their reported performances, our BANet achieved the second-best (2 times) and third-best (1 time) performances of macro F1 scores (see Table 5.5). While the design of our BANet is generalizable for processing movement data collected from multiple body parts, such competitive performances are very encouraging as they demonstrate that our method could be useful to the broader community working on body movements.

In this subsection, we further test on another typical benchmark wearable HAR dataset, namely Skoda [67], to evaluate the performance of our BANet beyond PBD. We also compare with other methods that have been tested on this dataset, including ConvLSTM [65], LSTM-S [13], Ensemble of LSTM [60], Att.Model [16], and attenLSTM [15].

For BANet, the input is directly data collected from the 10 accelerometers attached to the right arm of the participant without using low-level features, with one LSTM layer comprising 64 hidden units used as the temporal encoder. The data is down-sampled to 33Hz, with 80% of data per class used for training and the rest used for validation (10%) and testing (10%). The results are reported in Table 5.6.

As shown, our proposed BANet achieves the highest performance on this wearable HAR dataset in comparison with other methods with or without using attention mechanisms. The competitive performances achieved so far by BANet on

Table 5.6: The results of our BANet and other compared methods on Skoda dataset for human activity recognition.

Method	Macro F1 score
LSTM-S [13]	0.92
ConvLSTM [65]	0.91
Ensemble of LSTM (M=20-CE(20)) [60]	0.93
Att.Model [16]	0.91
AttenLSTM [15]	0.94
BANet (ours)	**0.96**

a series of HAR datasets suggest that the model we develop for PBD could be also useful for other movement-based tasks.

5.4 Summary

This chapter investigated the use of both temporal and bodily attention mechanisms combining LSTM layers to improve the detection of activity-independent protective behavior within pre-segmented activity instances. In comparison to the state-of-the-art [15, 16, 17], our architecture delayed the attention processes to the second and third levels of the architecture to enable primary learning of low-level features as the movement was processed. In doing so, both attention mechanisms worked on a higher-level representation of the movement. The results showed that such an approach led to a substantial improvement (Macro F1 score increases from 0.572 to 0.844). Further, it showed results slightly higher than other LSTM-based architectures (without significance against two of them), with a critical decrease in number of trainable parameters (from 40,940 to 8,131).

The results also suggested that bodily attention mechanism played a more important role than the temporal attention mechanism (Macro F1 score of 0.831 vs. 0.758 respectively). Still, the combination of the two mechanisms led to a better performance (Macro F1 score of 0.844). This suggested that the temporal attention mechanism may capture more detailed local temporal dynamics missed by the bodily attention one. In addition, such temporal dynamics may also be critical in discriminating between strategies.

We also discussed some examples of how the two types of attention mechanism

weights captured aspects of movements by people with CP discussed in the pain literature. In addition, these results also suggest that such systems, deployed in everyday rehabilitation activity, could be used to provide more personalized feedback to patients by building functions based on attention weights.

In fact, BANet has recently been used to develop a sonification software driven by the attentional weights to let patients explore how they move [115]. Studies with patients have not been yet run. As the behavioral study on CP is still developing and mainly based on lab studies [44, 45], BANet could also contribute to the qualitative pain literature by enabling a data-driven understanding of protective behavior in everyday life.

We concluded by demonstrating the competitive if not better performances of the proposed BANet on a series of HAR datasets (as seen in the HAR study [20] that compared with our method, and the experiment reported in our thesis), in comparison with other methods that had been specifically developed in the context of HAR. This suggests that the method we developed for PBD has a certain level of generalizability to carry out movement-based tasks from behavior detection to activity recognition.

This work was done during 2019, and is published in a workshop of conference ACII'19 [10]. By the point of completing this thesis, the work presented in this chapter has received 23 citations (excluding the self-reference by my own works). Aside from the study [20] that compared with our work for activity recognition, our attention-based method has inspired the study [120] on the design of their spatial attention mechanism in the development of an attention-based graph convolutional network for bodily emotion recognition. Aside from the bodily attention mechanism proposed in our work, the use of a 1×1 convolutional layer for temporal attention computation is also used by [121] in designing their attention-based LSTM network for group behavior detection in human-robot interaction.

Chapter 6

Improving Protective Behavior Detection in Continuous Data

The studies presented above demonstrate how to leverage deep learning for better protective behavior detection (PBD) on instances of various activity types with suitable data preprocessing methods (*i.e.*, segmentation and augmentation) and an attention-based deep learning model. However, interesting PBD results are only achieved within pre-segmented activity instances.

Pre-segmentation was used in our initial studies (Chapter 4 and Chapter 5) to help focus on understanding the feasibility of using deep learning for activity-independent PBD. In order to move one step closer to real-life PBD systems, pre-segmentation cannot be used anymore. This is because: i) a real-time system needs to function without knowing in advance what type of activity is about to occur; ii) in real life, activities are not well separated from each other, but they generally merge into one another. In short, rather than waiting for a full activity instance to be recognized, it would be useful to provide feedback when the activity is being performed as soon as protective behavior is being detected. As a result, in this chapter, we study continuous PBD without pre-segmentation that should be fully agnostic to the type of activity being performed.

In this chapter, we aim to address such a limitation by leveraging recognition of the context, namely the continuous recognition of the activity (HAR). In the literature review of Chapter 2, we showed how context recognition has proved to

support the targeted task [85, 88, 90]. However, the same question has never been explored in the context of bodily expressions and surely not for the challenge of continuous PBD. Hence, the research question we investigate in this chapter is: how can contextual information be leverage to enable fully continuous PBD across sequences of activities with their transitions?

Here we consider two levels of contextual information in this study. The first type is the inherent configuration of the body, represented by data collected from sensors attached to different bodily positions. In Chapter 5, we showed the importance that local bodily movement dynamics have in PBD, leading to the use of a network with bodily attention mechanism only that performs similarly to the integrated BANet.

The second is the type of activity being performed, which also strongly builds on the first level of body configuration. In Chapter 4, our analysis suggested that the network needs to have sufficiently long windows to gather information about the activity that is performed in order to further understand if the movements executed are abnormal or not. However, all the networks and learning model explored in previous chapters were not specifically enabled to leverage such a context. Therefore, we explore approaches to more directly leverage the contextual information in this chapter. The contribution made in this chapter is five-fold.

- For the first time, continuous detection of protective behavior is studied across full data sequences of CP patients. Previously, continuous PBD was only established on pre-segmented activity instances.

- A novel hierarchical HAR-PBD architecture is designed to leverage activity recognition to enable detection of protective behavior (*i.e.*, movement behavior driven by emotional variables) in continuous data sequences. Protective behavior was investigated in the past without leveraging its activity background.

- Graph convolution (GC) [122] and long short-term memory (LSTM) [97] layers are combined to model the configuration of body-worn inertial measurement units (IMUs) for PBD, while in the past only convolutional neural networks (CNNs)

[123] and LSTMs were applied. Although the concept of combining GC and LSTM exists in computer vision, it is used for the first time to show the advantage of graph representation in the context of emotional behavior across activities.

- A loss function referred to as CFCC loss is used to alleviate class imbalances of continuous data. Investigations of its effect on HAR and on PBD are reported.
- Comprehensive experiments and analyses using data collected from both CP and healthy participants. Various training strategies of the proposed hierarchical architecture are explored, and an analysis of simulating fewer IMUs demonstrates the applicability and efficacy of our method on smaller sensor sets.

6.1 Challenges in Continuous Data

The EmoPain dataset contains full-body movement data continuously captured from chronic pain (CP) and healthy participants during sequences of movements reflecting everyday activities. We refer to these as activities-of-interest (AOIs) since they were chosen by physiotherapists as particularly demanding for people with CP and likely to trigger protective behavior.

While this dataset was not collected in the wild, participants performed each activity without how-to instruction, and transitions between AOIs further created noise typical of in-the-wild data collection. During transition periods, participants could rest according to their needs or enjoy casual movements such as stretching, walking, and self-preparation. An illustration of a complete data sequence of one CP participant with the activity and behavior annotations is shown in Fig 6.1.

The proportion distribution of protective behavior samples within each activity instance across all participants with CP is presented in Figure 6.2. As shown, most protective samples are labeled within the AOIs, while only a few transition samples are labelled as protective behavior. This is because many of the transition periods include relaxing movements that people adopt on their own as they took a break from the instructions.

The boxplots also present noticeable individual differences within a same activity type, *e.g.*, some people show more protective behavior during sit-to-stand

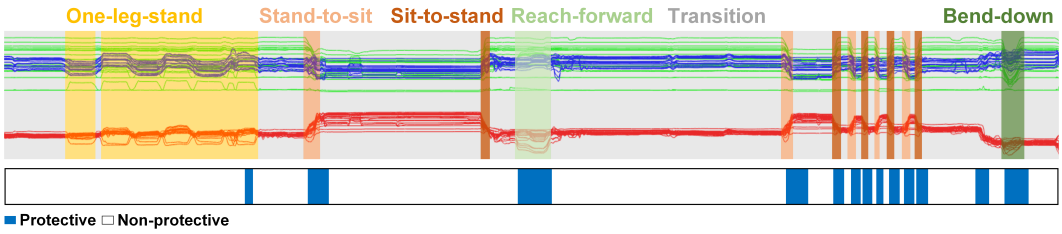


Figure 6.1: An example of the full data sequence from a CP participant, comprising AOIs and transitions. Lines are red, green, and blue for the x, y, and z coordinates data, respectively. Protective behavior labels (majority-voted) are shown below the sequence.

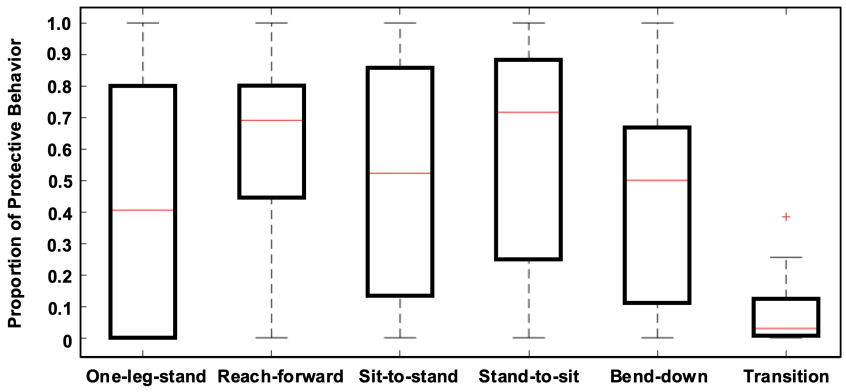


Figure 6.2: The proportion of protective behavior in each activity type across all the participants with CP.

than others. One of the challenges we aim to address in this chapter is the varying contexts of protective behavior in the continuous data, which is the different types of activity background.

For the continuous processing of a sequence, the presence of different activity types alters the way protective behavior is presented, making it more difficult for the model to learn to detect it. Inspired by the studies reported in Chapter 2 that adopt context recognition to aid the task-of-interest [85, 88, 90], we also explore how to improve the PBD in continuous data with a scheme to automatically recognize its context of activity background.

Another challenge we shall handle is the class imbalance, which is also the direct picture of a long data sequence comprising AOIs or events-of-interest. For data in EmoPain dataset, protective behavior samples only take up 21.09% in average across all the CP participants. Class imbalance also exists among different activity

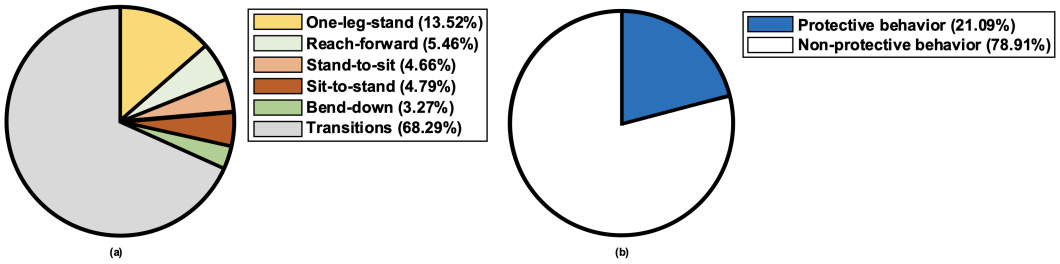


Figure 6.3: The average distribution of (a) activity classes in the entire dataset and (a) protective behavior across all the CP participants.

types, as shown in Figure 6.3. Resting and preparation periods in transition for a new activity are typical for people with CP, and may indeed reflect a large part of real-life data sequences. Toward this, we propose to use a loss function referred to as CFCC loss to alleviate class imbalances of continuous data during the model training.

6.2 Method

An overview of our proposed architecture is presented in Figure 6.4. Both HAR and PBD modules receive the same consecutive frames. These are extracted with a sliding-window from the data sequence collected with 18 IMUs. For HAR module, the activity type label (5 AOIs plus transition) is used for training, whereas for PBD the protective behavior label (absence and presence) is used. In addition, the first module (HAR) aims to recognize the type of activity being performed and pass such information to the second module (PBD) that detects the presence or absence of protective behavior.

For our main experiments, the HAR module is pre-trained with activity labels on the same folds of data during each round of leave-one-subject-out validation (LOSO) used for PBD. The weights of the HAR achieving the highest activity recognition accuracy is saved. The HAR module is *frozen* with such pre-trained weight loaded when used in the hierarchical architecture. Therein, the activity classification output is concatenated with the same piece of input frame and passed to train and test the PBD module using labels of protective behavior.

We use this frozen (optimized) HAR module to better understand the benefit of using the proposed hierarchical HAR-PBD architecture. Further analyses using a

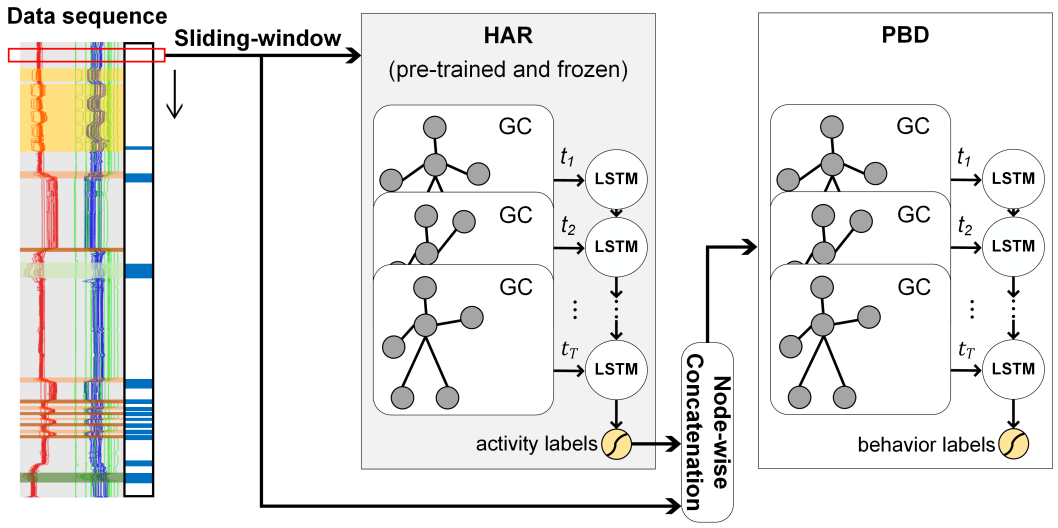


Figure 6.4: The proposed hierarchical HAR-PBD architecture, comprising the human activity recognition (HAR) module and protective behavior detection (PBD) module. By default, using the same data input, the HAR module is pre-trained with activity labels and frozen with weights loaded during training of the PBD module.

non-frozen HAR module are reported at the end of the chapter. To the best of our knowledge, this is the first implementation to leverage HAR to explicitly inform another concurrent task on the same data.

Both modules in our proposed architecture use a similar network comprising graph convolution (GC) and LSTM layers. The GC layer is used to model the body configuration information collected from 18 IMUs. Meanwhile, LSTM is used to learn the temporal dynamics across graphs corresponding to the body movement at different timesteps, critical for both HAR and PBD (*e.g.* hesitation slows down movements, and fear of pain or perceived pain lead to difference in timing of body-part engagement for the same activity).

6.2.1 The GC-LSTM Network for HAR and PBD Modules

In the previous chapters, we reviewed and explored deep learning methods proposed at earlier stages for activity recognition, *e.g.* vanilla neural networks and architectures designed considering the spatial configuration of the sensor/joint network. Performance improvements achieved by these methods suggest that body configuration information is important for activity recognition as well as PBD.

For both HAR and PBD modules in our proposed architecture, a network integrating GC and LSTM layers is used, referred to as HAR/PBD GC-LSTM. There are three considerations for the design of HAR/PBD GC-LSTM.

- The limited size of the EmoPain dataset in comparison with popular visual HAR benchmarks [6, 70] that have been used to evaluate GCNs, making it difficult to adopt more complex existing implementations.
- The need to verify if the graph representation is indeed capable of improving PBD, which requires using GCN as a way to learn data representations and removing unnecessary designs, *e.g.* embedding GCN into LSTM.
- The need to connect the HAR module with the PBD module, which requires the GC-LSTM network to tolerate the fusion of activity information and movement data at input level.

In this work, we focus on a conceptually simple implementation that builds parallel connection between GC and LSTM layers as the basic component in our architecture. Such design is helpful to verify the advantage of using a graph representation to model data from multiple IMUs in the context of HAR and PBD. Explorations of GC-LSTM variants may further improve performances, but are out of the scope of this thesis, since they are merely the backbone in our model.

Graph Input. As we described in Chapter 3, at each timestep, the EmoPain dataset provides 3D coordinates of 22 body joints that were calculated from the raw data stored in a Biovision Hierarchy (BVH) format.

Graph Notation. A body-like graph is built to arrange each of the 22 joints to be a node connected naturally in the graph to the other joints, as shown in Fig 6.5. We denote the graph as $\mathcal{G} = (V, E)$, with a node set $V\{t,i\} = \{v_{ti} \mid t = 1,...,T; i = 1,...,N\}$ representing the N nodes of a graph at timestep t within a graph sequence of length T, and an edge set E representing the edges connecting the nodes in this graph. Since in this work independent LSTM layers are used to learn the temporal dynamics across graphs at different timesteps, the inter-skeleton edge (usually represents the

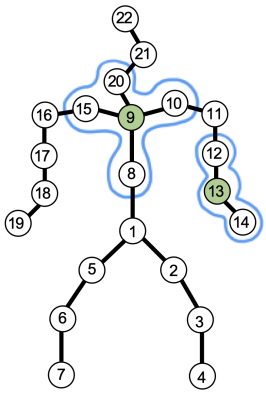


Figure 6.5: The built graph input at a single timestep, where each node represents a human body joint. The blue contour marks the neighbor set (receptive field) of the centered node in green.

temporal dynamics) connecting consecutive graphs is not leveraged. Therefore, only the intra-skeleton edge (representing the connection of body joints) is considered with $E\{i,j\} = \{(v_{ti}, v_{tj}) \mid (i,j) \in B\}$, where B is the set of naturally connected nodes (joints) of the human body graph.

An adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$ is used to identify the edge E between nodes, where $A_{i,j} = 1$ for the connected i-th and j-th nodes, and 0 for disconnected ones. $\mathbf{A}$ stays the same for all the tasks in this work. In other words, the basic configuration of a graph is independent of time and participants, while the relationship between different body parts in different activities is learned during training. The identity matrix is $\mathbf{I}_N \in \{1\}^{N\times N}$, a diagonal matrix that represents the self-connection of each node in the graph. With the adjacency matrix $\mathbf{A}$ and identity matrix $\mathbf{I}_N$, the body configuration is represented by matrices and can be processed by neural networks.

The feature of each node in a graph at timestep t is stored in a feature matrix $\mathbf{X}_t^{\mathcal{G}} \in \mathbb{R}^{N\times \mathbf{3}}$. The raw feature of each node is the coordinates of the respective body joint, denoted as $\mathbf{X}_{v_{ti}}^{\mathcal{G}} = [x_{ti}, y_{ti}, z_{ti}]$. The neighbor set of a node v_{ti} is denoted as $\mathcal{N}(v_{ti}) = \{v_{tj} \mid d(v_{ti}, v_{tj}) \leq D\}$, with the distance function $d(v_{ti}, v_{tj})$ accounting for the number of edges in the shortest path traveling from v_{ti} to v_{tj} and threshold D defining the size of the neighbor set. Following previous studies [77, 78, 79, 80, 81, 33], we set $D = 1$ to adopt the 1-neighbor set of each node.

Graph Convolution. Basically, a GC comprises two parts, one defines the way to

sample data from the input graph and the other concerns assigning learnable weight to the sampled data. It should be noted that a higher-level knowledge about the subset of body parts relevant to specific activities is not manually provided in the network. Therefore, only low-level rules like sampling and weighting are defined in the GC, which allows the network to develop its understanding about the movement.

In our case, the GC needs to conduct sampling on the full-body graph comprising 22 nodes. Following the derivation of GCN presented in [77], the GC used in this work can be written in detail as

$$f_{out}^{GC}(\upsilon_{ti}) = \sum_{\upsilon_{tj} \in \mathcal{N}(\upsilon_{ti})} \frac{1}{Z_{ti}(\upsilon_{tj})} f_{in}^{GC}(P^{GC}(\upsilon_{ti}, \upsilon_{tj})) \cdot \mathbf{w}^{GC}(l_{ti}(\upsilon_{tj})), \tag{6.1}$$

where $P^{GC}(\upsilon_{ti}, \upsilon_{tj}) = \upsilon_{tj}$ is the graph-adapted sampling function with $d(\upsilon_{ti}, \upsilon_{tj}) \leq 1$, $\mathbf{w}^{GC}(\upsilon_{ti}, \upsilon_{tj}) = \mathbf{w}'(l_{ti}(\upsilon_{tj}))$ is the graph-adapted weight function with $l_{ti}(\upsilon_{tj}) = d(\upsilon_{ti}, \upsilon_{tj})$, $\mathbf{w}'$ is the trainable weight matrix, f_{in}^{GC} is the input feature of the sampled node set at current layer while f_{out}^{GC} is the output feature of the respective centered node υ_{ti}, and $Z_{ti}(\upsilon_{tj}) = \mathbf{card}(\{\upsilon_{tk} \mid l_{ti}(\upsilon_{tk}) = l_{ti}(\upsilon_{tj})\})$ is a normalization term representing the cardinality of the partitioned subsets in the neighbor set. The 1-neighbor set $\mathcal{N}(\upsilon_{ti}) = \{\upsilon_{tj} \mid d(\upsilon_{ti}, \upsilon_{tj}) \leq 1\}$ is applied to be the receptive field of each node υ_{ti}, as depicted by the blue contour in Fig 6.5.

Within the weight function, the partition function $l_{ti} : \mathcal{N}(\upsilon_{ti}) \rightarrow \{0, \ldots, K-1\}$ can be used under different strategies, while in our work the distance-partitioning strategy [77] is adopted that divides the 1-neighbor set $\mathcal{N}(\upsilon_{ti})$ into two subsets, namely the centered node υ_{ti} and the remaining neighbor nodes $\upsilon_{tj} \mid d(\upsilon_{ti}, \upsilon_{tj}) \leq 1$. As a result, we have $K = 2$ subsets thus $l_{ti}(\upsilon_{tj}) = d(\upsilon_{ti}, \upsilon_{tj})$. By using the distance-partitioning strategy, $Z_{ti}(\upsilon_{tj})$ equals to the number of all the neighboring nodes υ_{tj} within the same neighbor set because they are within the same subset as well.

Using the adjacency matrix $\mathbf{A}$ and identity matrix $\mathbf{I}_N$, we follow the forward-passing formula presented in [122] to implement the GC used in this work as

$$\mathbf{f}_{out}^{GC} = \hat{\Lambda}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\Lambda}^{-\frac{1}{2}} \mathbf{f}_{in}^{GC} \mathbf{W}, \tag{6.2}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ represents the inter- and self-connection of each node, and $\hat{\Lambda}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$ is a diagonal degree matrix of $\hat{\mathbf{A}}$. Since $\hat{\Lambda}$ is a positive diagonal matrix, the entries of its reciprocal square root $\hat{\Lambda}^{-\frac{1}{2}}$ are the reciprocals of the positive square roots of the respective entries of $\hat{\Lambda}$. Each diagonal value in the degree matrix $\hat{\Lambda}$ counts the number of edges connecting the respective node in the graph described by $\hat{\mathbf{A}}$. Such transformation from $\mathbf{A}$ to $\hat{\mathbf{A}}$ is in accord with our choice of distance-partitioning [77], where each neighbor set is divided into two subsets for weight assignment, namely the center node ($\mathbf{I}_N$) and the neighbor nodes ($\mathbf{A}$). $\mathbf{f}_{in}^{GC}$ is the input feature matrix, and $\mathbf{f}_{in}^{GC} = \mathbf{X}_t^{\mathcal{G}}$ at the first layer of input level. $\mathbf{W}$ is the layer-wise weight matrix.

Connecting Graph Convolution with LSTM. For each module, the input to a single unit of the first LSTM layer is the concatenation of the GC output from all the nodes in the graph $\mathcal{G}$ at timestep t, denoted by $\mathbf{f}_{out}^{GC}(\mathbf{X}_t^{\mathcal{G}}) = [f_{out}^{GC}(v_{t1}), \ldots, f_{out}^{GC}(v_{tN})]^\top$. For the adopted forward-processing LSTM layer, the computation at each LSTM unit is repeated to process the information across graphs from the first timestep to the last. Such conceptually-simple design involving the GC only as a way to learn representations enables us to empirically study its impact on PBD performances.

In comparison, another study embedded GC within the LSTM unit [79]. While this may improve the performance, it becomes more difficult to differentiate the advantage of each component. Additionally, some works proposed to improve performances by using extra computational blocks (*e.g.* fully-connected layers or attention mechanisms [78, 124]) between GC and LSTM layers, which in turn add more trainable parameters to the network that could lead to over-fitting on smaller datasets like ours. Nevertheless, we believe the improvement of the backbone would increase the performance of the entire architecture. We leave this to future works.

6.2.2 Hierarchical Connection of HAR and PBD Modules

Up until this point, the GC-LSTM network used in each module of our proposed architecture has been defined. Here, we describe how to connect HAR and PBD.

In each module, a fully-connected softmax layer is added to the GC-LSTM

network for classification. Let the probability toward each class of the current input frame to be $P = [p_1, \ldots, p_K]$ with K denoting the number of classes, and the one-hot prediction to be Y. K is 6, including the 5 AOIs and transition class for the HAR module, and is 2 for protective and non-protective behavior of the PBD module.

In our proposed architecture, to provide activity-informed input from HAR to PBD, a node-wise concatenation is used where the predicted activity label Y^{HAR} is added to the input matrix $\mathbf{X}^{\mathcal{G}}_{v_{ti}} = [x_{ti}, y_{ti}, z_{ti}]$ of each node of the graph input for PBD (see Fig 6.4). Namely, for the PBD module, activity-informed input feature matrix at a node v_{ti} of a single graph is $\mathbf{X}^{\mathcal{G},PBD}_{v_{ti}} = [\mathbf{X}^{\mathcal{G}}_{v_{ti}}, Y^{HAR}]$. Since the raw graph input fed to the PBD module is joined by the output of the HAR module, we call such a **hierarchical connection** between the two.

6.2.3 Addressing Class Imbalances with CFCC Loss

A problem with datasets targeting real-life situations is class imbalance (*e.g.* datasets for HAR [67, 65, 66]). In the case of the EmoPain dataset, protective behavior is sparsely spread within the AOIs of a movement sequence, while it is generally absent during transitions (see Fig 6.2). Specifically, on average the AOIs represent only 31.71% of a participant's data sequence, with the rest being transition activities. Furthermore, on average, samples labelled as protective represent only 21.09% of a patient's data sequence, with the rest labelled as non-protective (see Fig 6.3).

Typical approaches used to address class imbalance include: i) data re-sampling for each class, where samples are either duplicated from the less-represented class or randomly sampled from the majority class [125]; ii) loss re-weighting, *e.g.* setting higher weights for the loss computed from less-represented class and lower weights for the loss computed from majority class [126]. Unfortunately, these methods require interferences with data samples directly that could harm the training of a model [127], *e.g.* misclassifying samples of the majority class to be the less-represented class given the hard manual inference.

In our work, we propose to use a loss function that directly alleviates class imbalance during training. Normally, for the supervised learning conducted in our

modules, the following categorical cross-entropy loss (CCE) [128] is used

$$\mathcal{L}_{categorical}(P,Y) = -Y\log(P), \tag{6.3}$$

where $P = [p_1,\ldots,p_K]$ is the predicted probability distribution of an input frame over the K classes, and Y is the respective one-hot categorical ground truth label with $Y(k) = 1$ only for the ground truth class k.

During training, the loss computed for each frame is added up to be the total loss for the model to reduce. Such function tends to bias the model to put more attention on decreasing the loss in the majority class and ignore the (mis)classification of the less-represented classes (*e.g.* the AOI classes in the HAR task or the protective behavior class in the PBD task).

To address this, we took inspiration from the research on automatic object detection. In the object detection, a binary-class imbalance usually exists given the smaller area covered by the object-of-interest and the larger objectless background.

Two practical approaches proposed to solve such an issue are found, namely the focal loss [129] and the class-balanced term [127]. Based on binary cross-entropy loss [128], focal loss applies a **sample-wise** factor function to adjust the loss weight for a sample based on its classification difficulty (judged by the predicted probability toward the ground truth class). The focal loss (FL) together with binary cross-entropy loss (CE) can be written as

$$\mathcal{L}_{FL}(p,y) = (1-p_{GT})^{\gamma}\mathcal{L}_{binary}(p,y) = -(1-p_{GT})^{\gamma}(y\log(p)+(1-y)\log(1-p)), \tag{6.4}$$

where p is the predicted probability toward the positive class of the current data sample, y is the binary ground truth indicator with 1 for the positive class and 0 for the negative class, $p_{GT} = yp+(1-y)(1-p)$ is the predicted probability toward the ground truth class. As we can see, the factor $(1-p_{GT})^{\gamma}$ with tunable hyperparameter $\gamma \geq 0$ is added to the original binary cross-entropy loss. The intuition is to reduce the loss computed from data samples that are well-classified, while the threshold for judging this needs to be tuned given different datasets and is controlled by γ. The increase of γ will reduce the threshold, then data samples with comparatively

lower classification probabilities toward the ground truth class would be treated as the well-classified.

In [127], the authors further revised the vanilla cross-entropy loss by adding a **class-wise** loss weight to each class based on the so-called effective number of samples within it. For class c, the effective number of samples is denoted as $E_{n_c} = \frac{1-\beta^{n_c}}{1-\beta}$, with a hyperparameter $\beta \in [0,1)$ controlling how fast the effective samples number E_{n_c} grows when the actual number of samples n_c increases. Practically, $\beta = \frac{n_c-1}{n_c}$. The class-balanced term is the reciprocal of E_{n_c}, written as

$$\frac{1}{E_{n_c}} = \frac{1-\beta}{1-\beta^{n_c}}. \tag{6.5}$$

Unlike the binary imbalance caused by the area of object and its useless background, in the HAR module, class imbalances exist among the 6 categories of activity, while in PBD both protective and non-protective classes share the same importance. Therefore, to adapt the focal loss and class-balanced term to scenarios of HAR and PBD, we replace the CE with CCE and combine the Equation 6.3-6.5 as

$$\mathcal{L}_{CFCC}(P,Y) = -\frac{1-\beta}{1-\beta^{n_k}}(1-YP)^{\gamma} Y \log(P), \tag{6.6}$$

where n_k is the number of frames of the ground truth class k for the current input frame. This revised function, referred to as **C**lass-balanced **F**ocal **C**ategorical **C**ross-entropy (CFCC) loss, will be used in our study.

To the best of our knowledge, this is the first time for such a combination to be used for the computation of multi-class categorical cross-entropy loss in HAR and PBD. With CFCC loss, we aim to alleviate class imbalances during training and also to understand its impact in comparison with the other component of our architecture.

6.3 Experiment Setup

In this section, we provide more details about the data preparation, validation method, metrics, and model implementations.

6.3.1 Data Preparation

Here, we briefly present the data preparations as follows.

Continuous Data Segmentation with Sliding-Window. Using a sliding window of 3s long and 50% overlapping ratio, each data sequence of a complete trial of a participant is segmented into consecutive frames from the start of the first AOI to the end of the last AOI/transition activity. The window length and overlapping parameters are based on the evaluation studies reported in Chapter 4.
At timestep t, we have an input graph $\mathcal{G}_t = (V_t, E_t)$, represented by the input data matrix $\mathbf{X}_t^{\mathcal{G}}$, constant adjacency matrix $\mathbf{A} \in \{0,1\}^{22\times 22}$, and its identity matrix $\mathbf{I}_{22}$, where $\mathbf{X}_{v_{ti}}^{\mathcal{G}} = [x_{ti}, y_{ti}, z_{ti}]$, $v_{ti} \in V_t$. These matrices only represent the graph structure and 3D joint coordinates data of each joint.

Ground Truth for Training. The activity class ground truth, *i.e.,* one-leg-stand, reach-forward, sit-to-stand, stand-to-sit, bend-down, and the transition, of a frame is defined by applying majority-voting to the 180 samples within it.
The protective behavior ground truth of a frame is decided by a binary majority-voting with 50% threshold across the 4 domain-expert raters in accord with Chapter 4 and 5.

Data Augmentation. Following Chapter 4 and Chapter 5, we apply jittering and cropping for data augmentation. For jittering, the normal Gaussian noise is globally applied with standard deviations of 0.05 and 0.1 separately to the original data sequence. For cropping, data samples at random timesteps and joints are set to 0 with selection probabilities of 5% and 10% separately. Each single augmentation method would create two extra augmented data sets, which are only used in the training set. The original number of frames produced with the sliding-window segmentation from all participants is approximately $6,200$, and is increased to around $31K$ after the augmentation.

6.3.2 Validation Method and Metrics

For all the experiments, a LOSO cross validation is applied across the 18 folds of participants with CP. This is because we observed that accuracies acquired at healthy

participants are almost 100%, thus to avoid biasing the average performance we do not build LOSO folds with the healthy participants for testing.

For HAR, we report accuracy and macro F1 score to account for performances of all classes. For PBD, as it is a binary task suffering from class imbalance, we additionally use the protective-class classification output of all folds to plot precision-recall curves (PR curves) and report the area-under-the-curve (PR-AUC) [130].

6.3.3 Model Implementations

A search on number of layers, convolutional kernels, and hidden units for the GC-LSTM network is conducted to identify the suitable hyperparameter set for HAR and PBD modules separately: i) for HAR, we use one GC layer with 26 convolutional kernels, three LSTM layers with 24 hidden units of each, and one fully-connected softmax layer with 6 nodes for output; ii) for PBD, we use three GC layers with 16 convolutional kernels of each, three LSTM layers with 24 hidden units of each, and one fully-connected softmax layer with 2 nodes for output. A dropout layer with probability of 0.5 is added to each GC layer and LSTM layer to alleviate the overfitting risk for all the models.

If not mentioned, the default loss used for all the models is the vanilla categorical cross-entropy loss written in Equation 6.3.

In CFCC loss, the class-balanced term does not vary per sample, instead it is acquired for a class given the number of samples therein, so is computed and fixed before network training. Thereon, we further conduct a hyperparameter search on $\gamma = \{0, 0.5, 1, 1.5, 2, 2.5\}$ for both tasks separately using the respective HAR or PBD module alone. We find $\gamma = 0.5$ to be suitable for HAR, and $\gamma = 2$ for PBD. Given the number of samples per class n_c, we set $\beta = \frac{n_c - 1}{n_c}$.

The Adam algorithm [102] is used as optimizer for all the models, while the learning rate is set to $5e^{-4}$ for the HAR module and $1e^{-3}$ for PBD module, after another search on $lr = \{1e^{-5}, 5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}\}$. The number of epochs is set to 100 for all the models.

During the hyperparameter search, a hold-out validation is adopted where 11 healthy and 17 CP participants are randomly selected to be the training set, with the

rest left out for validation. The validation data is then removed for the respective LOSO experiments. The aim of the hyperparameter search is to determine a proper set of hyperparameters to aid the following experiments, instead of mining the optimal hyperparameters for the dataset.

6.4 Results

The evaluation concerns several components of our proposed hierarchical HAR-PBD architecture, namely the use of graph representation, CFCC loss, and the hierarchical architecture connecting HAR and PBD modules. We conclude by evaluating different training strategies of the hierarchical architecture, and its performances under different sizes of the body graph input.

6.4.1 Contribution of Graph Representation to PBD

The first aim of our evaluation is to understand the contribution of graph representation in comparison with other learning approaches to the PBD performance. Hence, we conduct a set of experiments using the PBD module alone, without the use of the entire hierarchical architecture and CFCC loss.

The evaluation is conducted against the stacked-LSTM and BANet that we explored in the previous chapters, which either take i) joint angles and energies; or ii) 22 pairs of 3D joint coordinates as input. For stacked-LSTM, at each timestep we merely concatenate the coordinates of 22 joints to form the input matrix with a dimension of $22 \times 3 = 66$. Accordingly, the input structure of BANet is adapted for 22 pairs of coordinates, as illustrated in Fig 6.6. The search on number of LSTM layers, hidden units, and learning rates is also conducted for the two comparison

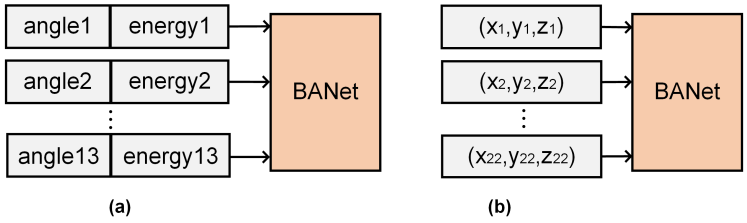


Figure 6.6: Input structures of (a) the original BANet, and (b) the adapted BANet for 22 pairs of 3D joint coordinates.

Table 6.1: PBD results with 95% confidence intervals of different representation learning methods. The best method is marked in bold.

Methods	Acc	Macro F1 score	PR-AUC
Stacked-LSTM (angle+energy)	0.79±0.066	0.61±0.055	0.23
BANet (angle+energy)	0.78±0.067	0.56±0.053	0.24
Stacked-LSTM (coordinate)	0.80±0.046	0.64±0.055	0.32
BANet (coordinate)	0.79±0.066	0.63±0.074	0.27
PBD GC-LSTM	**0.82±0.057**	**0.66±0.061**	**0.44**

models respectively under each input condition.

Differently from their original studies that relied on pre-segmentation of activity instances, both methods are applied here over the full data sequences in a continuous manner. Results are reported in Table 6.1 with PR curves plotted in Fig 6.7.

As shown, the PBD GC-LSTM produces the best accuracy of 0.82, macro F1 score of 0.66, and PR-AUC of 0.44. The actual difference between these compared methods is the way the input data is processed with, *i.e.*, traversal processing (stacked-LSTM), local processing (BANet), and graph representation (PBD GC-LSTM). As such, the results suggest that the graph representation may indeed contribute to improving the continuous detection of protective behavior. Still, the below-chance-level (< 0.5) results of PR-AUC of all methods demonstrate the difficulty of PBD in continuous data sequences. This implies the need to further improve continuous PBD with HAR and CFCC loss.

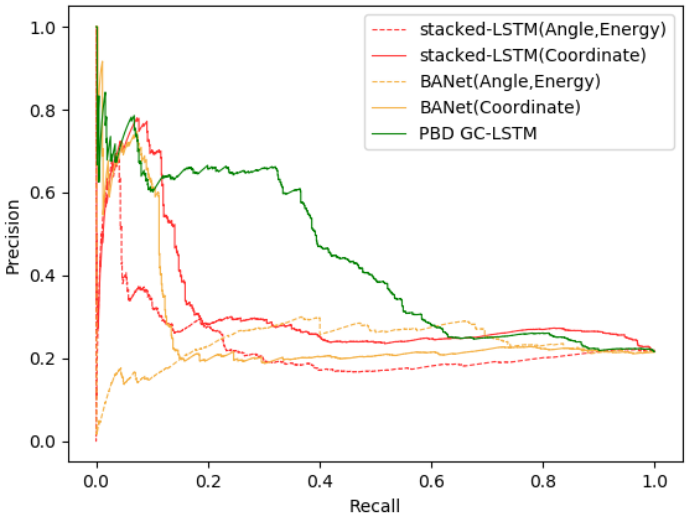


Figure 6.7: PR curves of different representation learning methods.

6.4.2 Contribution of CFCC Loss and HAR

Through an ablation study, we first investigate the contribution of CFCC loss alone in dealing with the imbalanced data for each module of our proposed architecture. We then use our proposed hierarchical architecture to understand the impact of activity-class information produced by the HAR module on PBD performance. In particular, we aim to understand if recognizing the activity background has more impact on improving PBD in continuous data sequences, in comparison with the issue of class imbalances during training.

Contribution of CFCC Loss to Continuous HAR. In our proposed hierarchical HAR-PBD architecture, the HAR GC-LSTM together with CFCC loss was firstly pre-trained on the same set of data using activity labels. Then, the set of weights achieving the best activity recognition performance was saved and frozen during the training of the entire architecture.

For the training and testing of the hierarchical architecture, the HAR output was used as auxiliary information to **contextualize** the PBD. Therefore, the accuracy of the HAR module is important for the PBD module. Here, we analyze the performance of the HAR GC-LSTM alone, with and without using CFCC loss. The results are reported in Table 6.2, with confusion matrices shown in Fig 6.8.

The CFCC loss leads to a higher macro F1 score (0.81 vs. 0.79) in the continuous HAR. Judging from the confusion matrices, CFCC loss reduces the classification bias toward the most represented class (the transition activity), which resulted in a lower accuracy though (0.88 vs. 0.89). These results show the effectiveness of CFCC loss for balancing multi-class categorical loss computation, which was not directly evaluated in the original studies [129, 127]. The computation of CFCC loss is independent of learning models and requires the only prior knowledge of the

Table 6.2: HAR results with 95% confidence intervals of the ablation study. The best method is marked in bold.

Methods	Acc	Macro F1 score
HAR GC-LSTM	0.89±0.028	0.79±0.052
HAR GC-LSTM with CFCC loss	0.88±0.027	**0.81±0.038**

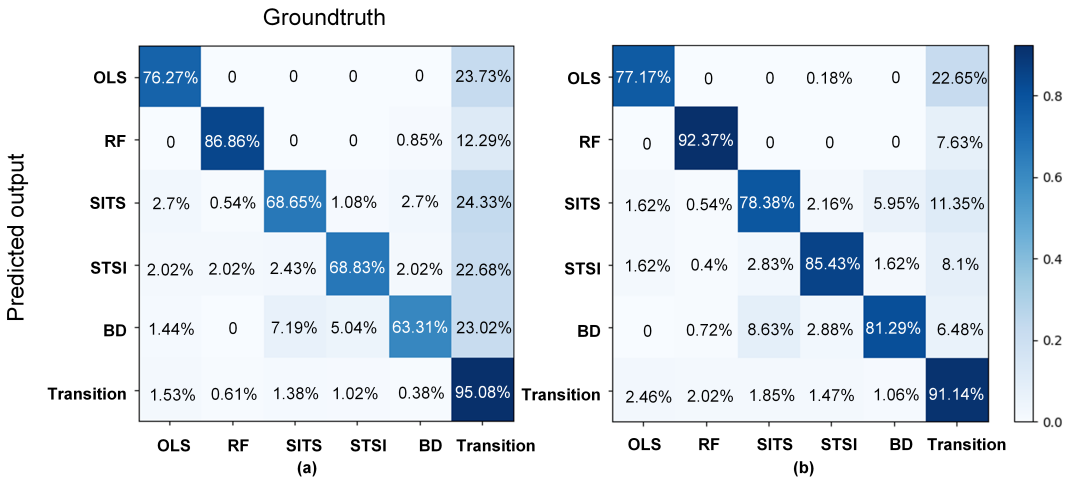


Figure 6.8: Confusion matrices of a) HAR GC-LSTM and b) HAR GC-LSTM with CFCC loss, where the bias toward the majority class of transition is balanced. OLS=one-leg-stand, RF=reach-forward, SITS=sit-to-stand, STSI=stand-to-sit, and BD=bend-down. The improvement on the less-represented class is obvious for the four classes in the middle.

number of samples per class. Therefore, CFCC loss should be useful for HAR tasks on relevant datasets.

Contribution of CFCC Loss to Continuous PBD. Here we investigate the contribution of CFCC loss to continuous PBD using the PBD GC-LSTM. The input to PBD GC-LSTM is the 3D joint coordinates data without activity-class information. As we can see from the results in Table 6.3, the use of CFCC loss leads to 5% improvement in macro F1 score (macro F1 score of 0.71 vs. 0.66). Confusion matrices shown in Fig 6.9 (a)(b) suggest that the CFCC loss does indeed help penalize the bias toward the more frequent class (non-protective class in this case) while improving the recognition of the less-represented one (protective class). However, the PR-AUC of 0.48 is still below chance level, suggesting that addressing class imbalance alone is not sufficient.

Proposed Method: Hierarchical HAR-PBD Architecture with CFCC Loss. For the training and testing of our proposed hierarchical HAR-PBD architecture, the HAR GC-LSTM within it is frozen and loaded with the weights from its pre-training with CFCC loss. This is to keep the HAR performance constant and aid the understanding of the impact of continuously inferred activity information on continuous PBD. The

Table 6.3: PBD results with 95% confidence intervals of the ablation study.

Methods	Acc	Macro F1 score	PR-AUC
PBD GC-LSTM	0.82±0.057	0.66±0.061	0.44
PBD GC-LSTM with CFCC loss	0.83±0.046	0.71±0.059	0.48
Hierarchical HAR-PBD architecture	0.84±0.053	0.73±0.053	0.52
Hierarchical HAR-PBD architecture with CFCC loss	**0.88±0.028**	**0.81±0.030**	**0.60**

results are reported in Table 6.3, with confusion matrix shown in Fig 6.9 (c).

It is interesting to see that our proposed hierarchical HAR-PBD architecture using vanilla categorical cross-entropy loss achieved an improvement of 2% with respect to the PBD GC-LSTM alone using CFCC loss (macro F1 score of 0.73 vs. 0.71). The PR-AUC of 0.52 is also above chance level. Such result shows that the contextual information of activity type contributes to continuous PBD, with our proposed hierarchical HAR-PBD architecture being a practical way for this.

Furthermore, by adding CFCC loss to the PBD module of the hierarchical HAR-PBD architecture, higher macro F1 score of 0.81 and PR-AUC of 0.60 are achieved (confusion matrix shown in Fig 6.9 (d)). The PR curves for the PBD ablation study are plotted in Fig 6.10. These results add to our previous finding and show that using a mechanism (CFCC loss in our case) to address the class imbalance problem led to a further-clear improvement. In general, our experimental results suggest that both the activity type information and CFCC loss are necessary for continuous PBD, despite one being more effective than the other.

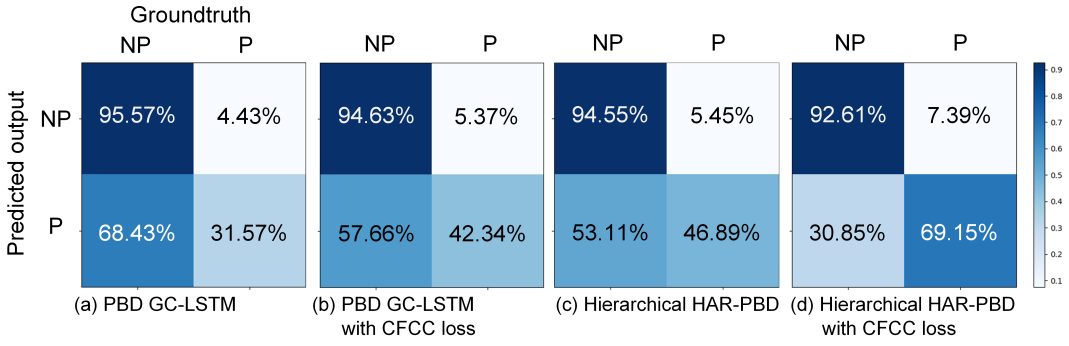


Figure 6.9: Confusion matrices for PBD methods in the ablation study. NP= non-protective, P=protective. The improvement on the protective class is obvious.

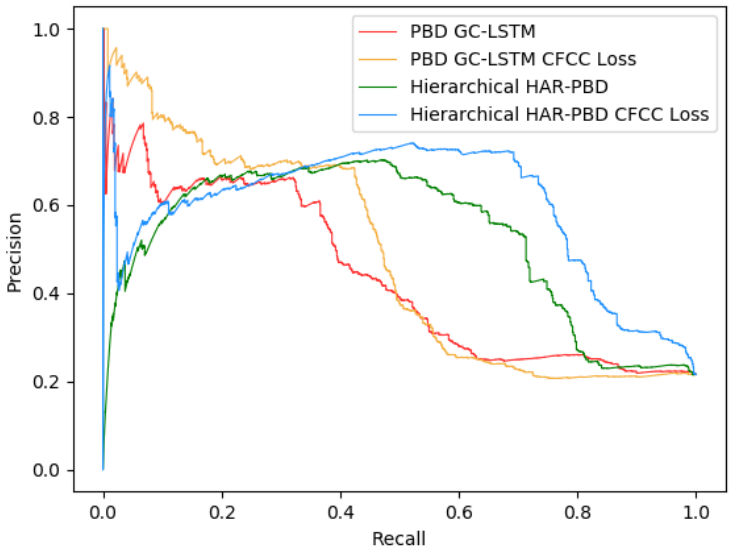


Figure 6.10: PR curves of different PBD methods in the ablation study.

6.4.3 Comparison of Training Strategies

In the previous subsections, the HAR module used in hierarchical HAR-PBD architecture was pre-trained with the same training data using activity labels and frozen to adopt the model of the best activity recognition performance. The aim was to understand the contribution of HAR to PBD across the different configurations.

Here, we further explore the relationship between HAR and PBD modules by exploring joint-training strategies of the hierarchical architecture. In joint-training of the architecture, the HAR module would not be frozen, but the activity labels are still used to update it when the PBD module is trained. Specifically, the protective behavior labels of the same data input together with the output of HAR module are used to train the PBD module.

Thereon, we compare the following four joint-training strategies together with the use of CFCC loss.

i) ***Joint HAR(CFCC)-PBD*** and ***Joint HAR-PBD(CFCC)***, where HAR and PBD modules are initialized and trained together using activity and protective behavior labels respectively, with CFCC loss only added to either the HAR or PBD module;

ii) ***Joint HAR-PBD with CFCC***, where CFCC loss is added to both modules in such joint training;

iii) ***Pre-trained Joint HAR(CFCC)-PBD*** and ***Pre-trained Joint HAR-PBD(CFCC)***,

similar to (i) where the only difference is that the HAR module is first trained alone with activity labels using CFCC loss to achieve the best activity recognition performance and then its training continues with the training of the PBD module;
iv) ***Pre-trained Joint HAR-PBD with CFCC***, where CFCC loss is added to both modules in the joint training of (iii).

For all these joint training strategies, the loss weights are set to $\{1.0, 1.0\}$ for both HAR and PBD modules. If CFCC loss is not mentioned, the loss used for the respective module is the vanilla categorical cross-entropy loss. We also compare them with our default method used in previous subsections, here referred to as ***Pre-trained HAR(Frozen)-PBD(CFCC)***, where the HAR module is first trained alone with activity labels and CFCC loss to achieve the best activity recognition performance per LOSO fold, then it is **frozen** with weights loaded and used in the hierarchical architecture for training and testing of the PBD module. Results are reported in Table 6.4, with the PR curves for PBD results plotted in Fig 6.11.

Without pre-training the HAR module, the best HAR, with macro F1 score of 0.56, and PBD performances, with macro F1 score of 0.74 and PR-AUC of 0.55, are achieved by the joint HAR-PBD(CFCC). However, by adding CFCC loss to the HAR module alone (joint HAR(CFCC)-PBD), the performances are reduced notably for the HAR and slightly for PBD. One explanation could be that the error passed back from the PBD module harmed the HAR performance, especially when such error of PBD was not well handled, *e.g.*, without using CFCC loss. In addition, by adding CFCC loss to both modules (joint HAR-PBD with CFCC), the HAR performance achieved of macro F1 score of 0.54 is comparable to joint HAR-PBD(CFCC) but the

Table 6.4: HAR and PBD results with 95% confidence intervals for different training strategies of the Hierarchical HAR-PBD architecture, the best method is marked in bold.

	HAR		PBD		
Training strategies	Acc	Macro F1 score	Acc	Macro F1 score	PR-AUC
Joint HAR(CFCC)-PBD	0.62±0.083	0.42±0.066	0.85±0.036	0.70±0.063	0.54
Joint HAR-PBD(CFCC)	0.76±0.045	0.56±0.065	0.84±0.047	0.74±0.042	0.55
Joint HAR-PBD with CFCC	0.66±0.063	0.54±0.077	0.81±0.059	0.71±0.057	0.45
Pre-trained Joint HAR(CFCC)-PBD	0.68±0.040	0.55±0.044	0.85±0.036	0.74±0.034	0.58
Pre-trained Joint HAR-PBD(CFCC)	0.84±0.047	0.73±0.073	0.87±0.034	0.79±0.039	0.58
Pre-trained Joint HAR-PBD with CFCC	0.72±0.043	0.64±0.061	0.85±0.030	0.76±0.045	0.55
Pre-trained HAR(Frozen)-PBD(CFCC)	**0.88±0.028**	**0.81±0.030**	**0.88±0.027**	**0.81±0.038**	**0.60**

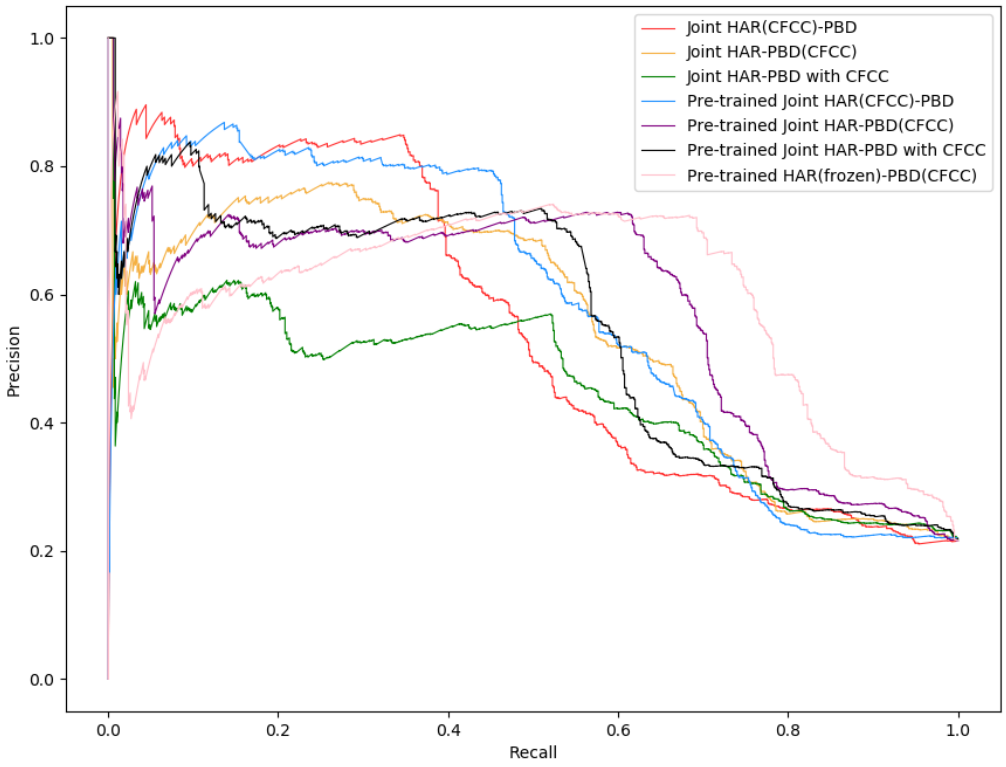


Figure 6.11: PR curves of the hierarchical architecture under different training strategies.

PBD performance is much lower, with macro F1 score of 0.71 and PR-AUC of 0.45. Given the current hierarchical architecture, such results suggest that alleviating class imbalance in PBD has a stronger impact on the overall performance in joint training, while addressing it in HAR would somehow penalize the PBD performance.

Rather than to start joint training from scratch, we further look into the uses of pre-training of the HAR module to reach an initial best activity recognition performance of macro F1 score of 0.81 before joint-training. A similar outcome as above is observed where the best performance is achieved by adding CFCC loss to the PBD alone. Once again, this proved the higher impact of alleviating the class imbalance of PBD, as the error passed back from the PBD module could harm the training of HAR module. In general, the results show that a pre-training of the HAR module improved the final performances of both HAR and PBD modules in comparison to the ones without it.

The performances achieved by the various joint-training strategies of the hierarchical architecture are still lower than the one of freezing the HAR module as used

in previous subsections, for both HAR, with macro F1 score of 0.81, and PBD, with macro F1 score of 0.81 and PR-AUC of 0.60. It should be noted that this method is a **two-stage** process in training and an **end-to-end** process in inference.

Although these results highlight the importance of HAR performance to PBD, they also suggest that the error propagated back from the PBD module in joint-training was not informative to improve the HAR performance. This highlights the need to further investigate the interaction scheme between HAR and PBD modules, beyond straightforward error back-propagation. This is left for future work.

6.4.4 Simulating Fewer IMUs

Until this point, we have assumed all 18 IMUs to be available to enable the input of a full-body graph. In this experiment, we quantify the fluctuation in performance when fewer IMUs are available. We simulate the limited availability of IMUs by removing nodes (containing data of respective joints) from the full-body graph.

According to the study on human observation of protective behavior [19], protective movement strategies are often visible on both sides of the body, even if via different patterns. For example, a twisting of the trunk to reach for a chair may lead to a narrower angle between the arm and the trunk on one side but a larger angle between another arm and the trunk. Therefore, a ***one-side sensor set*** of 14 nodes is created, where nodes number of 2-4 and 10-14 on the left limbs of the full-body graph are removed.

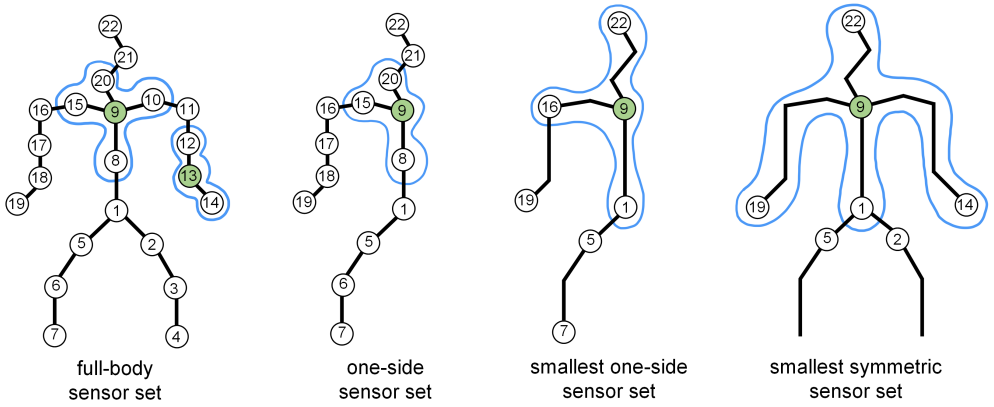


Figure 6.12: Graph structures of the four sensor sets. The blue contour marks the neighbor set of each centered node that colored in green.

Second, to simulate an even more compact sensor set, we further remove nodes number of 6, 8, 15, 17, 18, 20, and 21 from the one-side sensor set, resulting in a ***smaller one-side sensor set*** of 7 nodes. Additionally, from the full-body graph, we symmetrically remove nodes number of 3, 4, 6, 7, 8, 10-13, 15-18, 20, and 21 from both body sides to create a ***smallest symmetric sensor set*** with 7 nodes as well. The graph structures of these sensor sets still reflect human body connections, as shown in Fig 6.12.

The hierarchical HAR-PBD architecture with CFCC loss is used here on the graph input extracted from each sensor set. For a fair comparison, we report the results achieved in another optimization search that used to determine the suitable hyperparameters under each condition. The HAR and PBD results of each sensor set are shown in Fig 6.13, with PR curves for the PBD results plotted in Fig 6.14.

Although the best PBD performance is obtained by using the default graph input of 22 nodes (macro F1 score of 0.81 and PR-AUC of 0.60), competitive results are achieved using the one-side graphs with number of nodes reduced to 14 (macro F1 score of 0.77 and PR-AUC of 0.55) and even 7 (macro F1 score of 0.76 and PR-AUC of 0.53). These results are better than the ones achieved using the hierarchical architecture alone without CFCC loss on the full-body graph (macro F1 score of 0.73 and PR-AUC of 0.52).

On the other hand, given the same number of 7 nodes, the worst performance

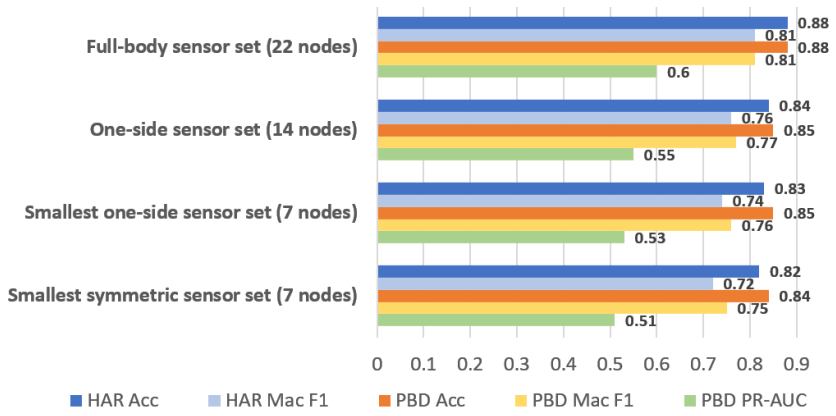


Figure 6.13: HAR and PBD results of the hierarchical HAR-PBD architecture with CFCC loss using input of different sensor sets.

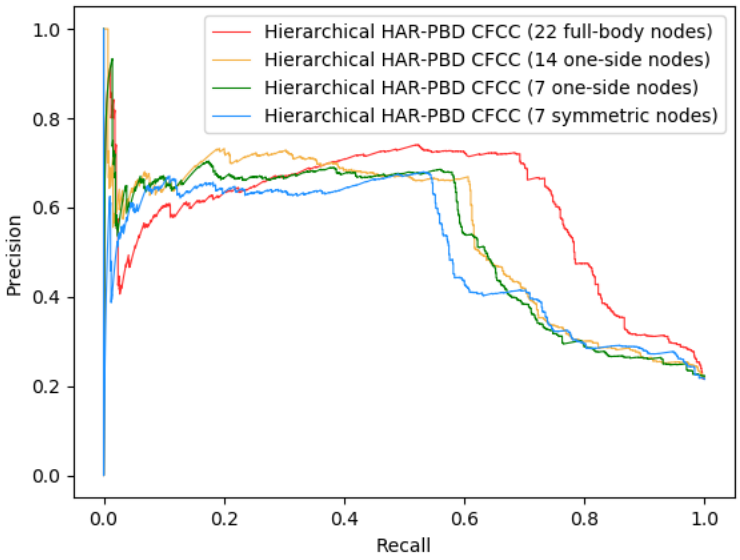


Figure 6.14: PR curves of the hierarchical architecture with CFCC loss using input of different sensor sets.

is achieved by the smallest symmetric sensor set that follows a general practice of retaining nodes on both sides of the body (macro F1 score of 0.75 and PR-AUC of 0.51). This shows the advantage of using a knowledge-driven strategy in guiding the sensor-set reduction, in the context of PBD.

It is empirically verified that the proposed hierarchical HAR-PBD architecture with CFCC loss leads to improvement even with small sensor sets. In order to further improve the PBD performance, efforts could be made on i) designing better graph structure, since in this work we merely employed the human-body connections; ii) further exploring the configurational pattern of body movement in the context of CP rehabilitation, given the performance achieved by one-side sensor sets.

6.5 Error Analysis with Visualization

So far, we see improvements and reductions in performances for both the HAR and PBD modules in the experiments conducted above. Here, to enable an in-depth understanding of the temporal functioning behavior of the two modules in the hierarchical HAR-PBD architecture, a visualized example of the model performances on the data sequence of one CP participant is shown in Fig 6.15. The upper two diagrams are the ground truth and recognition result of the HAR module, respectively.

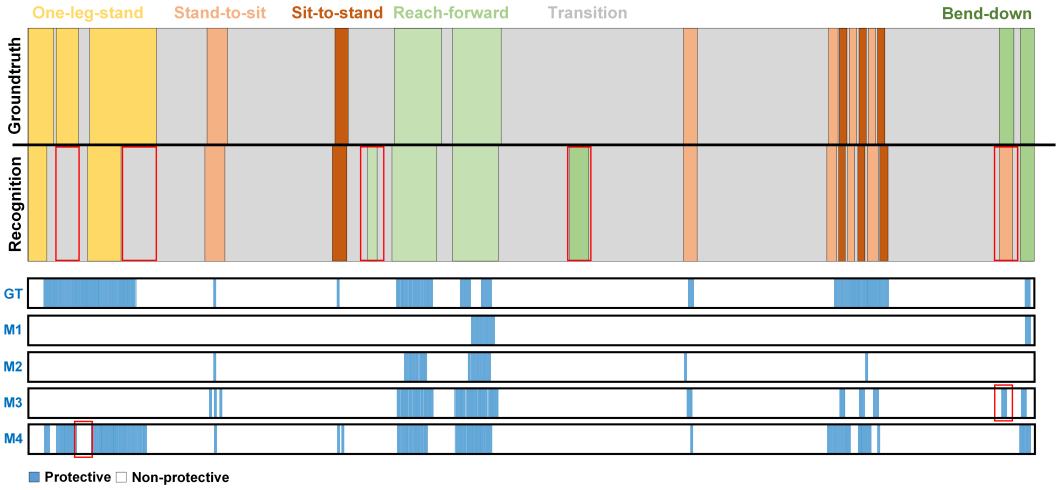


Figure 6.15: An example of the ground truth and results of HAR and PBD modules for data of a CP participant. The upper diagram is showing the ground truth of activity class and the recognition result by HAR GC-LSTM with CFCC loss. At the lower diagram, the first row is presenting the ground truth for PBD. 'M1' to 'M4' are respectively the detection result of i) PBD GC-LSTM; ii) PBD GC-LSTM with CFCC loss; iii) hierarchical HAR-PBD architecture, and iv) hierarchical HAR-PBD architecture with CFCC loss.

As shown, on this long data sequence, our HAR GC-LSTM using CFCC loss achieves good performances without any pre-localization and -segmentation of the AOIs. The lower five diagrams are the ground truth and results of the PBD module achieved by the four different methods, respectively.

In the HAR result (upper part of Fig 6.15), the errors are found to be: i) misclassification of one-leg-stand as transition activity (red rectangles on the left); ii) misclassification of transition activities as reach-forward and bend-down (red rectangles in the middle); iii) misclassification of bend-down as stand-to-sit (red rectangle in the right).

We notice that most misclassified activities were possibly due to their similarity in execution, given the use of protective behavior by this CP participant. For instance, the analysis of the on-site recorded video shows that the participant was unable/unwilling to raise the leg up during one-leg-stand, which is similar to the transition activity of standing still. During bend-down, the participant was not to bend the trunk but the leg and reached both arms to the ground, which is similar to the activity of stand-to-sit.

We now compare the four PBD approaches (see M1-M4 in the lower part of Fig 6.15). Without the activity-class information and CFCC loss, the baseline method of PBD GC-LSTM (M1) misclassified most frames as the majority class of non-protective behavior, which takes up around 78.91% in the training data. More protective behavior frames are correctly detected by using CFCC loss (M2), possibly owing to its ability to drive the model to focus more on the less-represented class, *i.e.,* the protective behavior class in our case. For this CP participant, M3 is shown to be more effective than M2 in terms of PBD during stand-to-sit, sit-to-stand, and bend-down. This could be mainly owed to the activity-type information on these frames provided by the HAR module. The hierarchical HAR-PBD architecture with CFCC loss (M4) leads to the best result, especially for PBD during one-leg-stand.

In the PBD result of the hierarchical architecture without CFCC loss (M3), the misclassified area marked by a red rectangle on the right side of the figure seems to be affected by the misclassification of bend-down as stand-to-sit in the HAR module. Such error is corrected by using CFCC loss (M4), possibly because it forces the model to adaptively down-weight the frames of majority class, *i.e.,* the non-protective behavior class in our case. However, for the same approach (M4), the error marked by a red rectangle on the left side is likely to have been affected by the misclassification of one-leg-stand as transition activity by the HAR module.

These results suggest that i) misclassifications by the HAR module have a negative impact on PBD performance; ii) and this problem could be minimized by addressing the class imbalance with CFCC loss in the PBD module. These support our concept of approaching continuous PBD by addressing the two technical issues together, namely the contextual information of activity types and the imbalanced presence of protective behavior in training.

6.6 Summary

In this chapter, we targeted PBD in continuous movement data as the milestone for using deep learning for CP rehabilitation. We proposed a hierarchical HAR-PBD architecture with GCN to recognize the varying context of activity to aid the

simultaneous detection of protective behavior. An adapted CFCC loss was also used to alleviate class imbalances existed in continuous data during training.

Our evaluation with data from the EmoPain dataset suggested that the activity type information is effective to aid PBD in continuous data, leading to a notable improvement over the baseline (macro F1 score of 0.73 and PR-AUC of 0.52 vs. macro F1 score of 0.66 and PR-AUC of 0.44), and is more impactful than just solving class imbalances (macro F1 score of 0.71 and PR-AUC of 0.48). The best result was achieved by combining the hierarchical architecture with CFCC loss, with macro F1 score of 0.81 and PR-AUC of 0.60. Additionally, we verified that graph representation improves the PBD performance.

We showed that it is feasible to jointly train the hierarchical HAR-PBD architecture. However, work is needed to gain mutual improvement between HAR and PBD modules. Furthermore, we showed the applicability and efficacy of our method using fewer nodes/joints (macro F1 scores of 0.77 and 0.76 with 14- and 7-node data input, respectively).

This work was done during 2019-2020, and is published in IMWUT [11]. By the point of completing this thesis, the work presented in this chapter has received 6 citations (excluding the self-reference by my own works). In a review about analyzing the gap between emotion and joint action from the perspective of behavioral neuroscience [131], our work was taken as an example to show how advanced machine learning technique has directly incorporated the learning of contextual information to aid the detection of affect-related behavior detection.

Chapter 7

Conclusion and Discussion

The development of deep learning and ubiquitous technology is opening a new era for the provision of healthcare support to people with chronic pain. To contribute to this innovation trend and novel opportunities, this thesis targeted the important first step of developing methods for activity-independent continuous automatic detection of protective behavior with deep learning. We identified and contributed to a series of research questions, which will not only benefit the development of intelligent physical rehabilitation systems for people with chronic pain, but also the broader community working on movement-based tasks.

In this chapter, we summarize the contributions and clarify possible use cases of our works. We then discuss the current limitations of the methods we proposed, and conclude by laying out the avenues for future works.

7.1 Summary of Contributions

The key contributions of studies presented above are as follows.

- In Chapter 4, published in [8, 9], we extended the previous state-of-the-art by showing the feasibility of protective behavior detection using deep learning on instances of various activity types in a continuous manner. Before this work, previous studies had focused on activity-dependent feature engineering to overall estimate the existence of protective behavior per activity instance [19, 56, 12, 29]. Our aim was to explore the possibility to detect where protective behavior occurs exactly within an instance of activity. The detection of where it occurs allows

to develop more specific interventions aimed to address the specific part of the activity that is feared by the person with chronic pain. Our findings brought the field one step closer to being able to continuously detect pain-relevant behavior in everyday life without knowing the type of activity in advance.

In addition, the study aimed at addressing critical data processing aspects that are necessary to apply deep learning. Targeting these points, this work has made the following contributions. For coping with the limited size of movement-related clinical dataset, a range of data augmentation strategies and their combinations were examined. An analysis and discussion of these methods shed light on how each of them could contribute to protective behavior detection beyond the dataset used in this thesis. The impact of data segmentation parameters on detection performance was also analyzed. Despite the fact that the optimal segmentation window length for protective behavior detection varies depending on the activity type, we provided a set of criteria for identifying practical parameters that work across different activity types, demonstrating how our approach could generalize to other datasets for protective behavior detection or analyzing affect-influenced movement behavior in general.

- In Chapter 5, published in [10], we explored the use of bodily and temporal attention mechanisms to better leverage the information that each body part carried at different stages of a movement. We proposed a novel deep learning model performing spontaneous temporal and bodily subsets learning, given the inspirations received from the following chronic pain studies.

  First, pain literature [21, 23, 30, 31] provided evidence that fears of injury, pain, and anxiety in chronic pain (chronic pain) cause the individual to engage bodily parts in ways that are not biomechanically necessary or efficient, but may create a sensation of control and assist to alleviate fear. Second, from [19] we also learned that, in designing interventions to improve movement-related self-efficacy of people with chronic pain, expert observers pointed out how specific body parts are particularly important to detect the presence or absence of protective behavior.

Through a range of experiments, we demonstrated that our method can achieve state-of-the-art results, if not slightly higher, with fewer trainable parameters for the detection of protective behavior. With attention score visualization and analysis, we discussed how such mechanisms could facilitate the better understanding of protective behavior from real-life measurements, rather than just lab-based observations.

A further evaluation on Skoda dataset [67], typically used as a benchmark for human activity recognition (HAR) research, showed the good generalizability of our method (macro F1 score of 0.96 vs. 0.92 [13], 0.91 [65], 0.93 [60], 0.91 [16], and 0.94 [15]) beyond the detection of protective behavior. In addition, another wearable HAR study [20] had also contributed to showing how the BANet could achieve competitive if not better performances in this context against the previous state-of-the-art models.

- In Chapter 6, for the first time, continuous detection of protective behavior was studied using data sequences comprising different activity types without pre-segmentation. This work is published in [11]. Previously, continuous protective behavior detection had been only established on pre-segmented activity instances.

  We proposed a novel hierarchical HAR-PBD architecture to leverage activity recognition to enable the detection of protective behavior in continuous data sequences. Protective behavior had been investigated in the past without leveraging its activity background. Graph convolution (GC) and long short-term memory (LSTM) layers were combined to model the body-worn inertial measurement units (IMUs) data for protective behavior detection, while in the past only convolutional neural networks (CNNs) and LSTMs were applied. It was also adopted for the first time to show the advantage of graph representation in the context of emotional bodily behavior across activities.

  A loss function referred to as CFCC loss was also employed to alleviate class imbalances of the continuous data. We further explored various training strategies of the proposed hierarchical architecture, and conducted an analysis of simulating

fewer IMUs to demonstrate the applicability and efficacy of our method on smaller sensor sets.

7.2 Future Use Cases

While the goal of this thesis is not to build a ubiquitous support system already to use for pain management, our contributions are the key components of such a system, since performance of protective behavior detection is critical for effective support.

For instance, the contextualization provided by the activity recognition module used in Chapter 6 not only leads to improved protective behavior detection performance, but informs assessment of people with chronic pain and customizes timely support for self-management. We discuss here the main use cases and further developments that can exploit our proposed methods to deliver new types of support and interventions in chronic-pain management and beyond.

7.2.1 In-the-Wild Informed Clinical Rehabilitation

Clinicians need to know about patients' difficulties in everyday activities outside the clinic's safe environment [43], and without relying on self-reported behaviors (*e.g.*, diaries), which are commonly used but have low reliability [26] because patients' awareness of habitual protective behavior and their triggers is low [27].

A ubiquitous system capable of identifying activity context and continually monitoring protective behavior could help physiotherapists gain a better understanding of the patient's activity challenges and progress, which typically changes across activities of interest. Connected to GPS and time, the system could further contextualize the activity with factors that introduce stress, *e.g.*, social pressures estimated from user's location. If equipped with attention mechanisms, as we explored in Chapter 5, the system could also provide some insights about, *e.g.*, the movement of specific body parts that tend to be more feared by the person.

7.2.2 Patient-Oriented Ubiquitous Self-Management

Because of the complexity of the actual world (environment, social demands, diversity of tasks and duties, etc.), as well as interference from emotional states, it

is normal to have difficulty translating movement strategies learned in the clinic to everyday life [38].

In [26, 25], a ubiquitous system transforms real-time movements (of specific body parts) into sound (sonification) to increase awareness in people with chronic pain of their physical capabilities. This further facilitates the autonomous use of movement strategies of the user beyond the clinic. If integrated in such a ubiquitous system, our methods (presented in Chapter 6) could help identify when and what kind of support is needed, *e.g.*, when the frequency of protective behavior during specific activities exhibited by specific body parts rises above a certain level; it can instantly provide reminders about breathing and taking breaks etc. Taking breaks and relaxation are important pacing strategies to avoid setbacks and prolonged recovery.

During exercise, the system can also provide dedicated suggestions or exercise plans based on the frequency of protective behavior detected. Actually, a recent effort following such a trend is seen in [115], where a sonification software is developed based on the attentional weights of BANet, the model we presented in Chapter 5.

7.2.3 From Chronic Pain to Next-Stage Movement Sensing

Beyond supporting the management of chronic pain, our proposed methods could be applied in a variety of contexts where ubiquitous activity recognition technology is being leveraged.

For example, ubiquitous technology is opening the new platform to aid workers in factory assembly lines [82], to support them in their workspace activities, *e.g.* to identify and help correct mistakes, to aid training, and establish human-robot collaboration. Thereon, another interesting application could be to promote workers' wellbeing, such as in reducing mental or physical stress. Therein, our methods (presented in Chapter 6) can be integrated into the system to leverage activity recognition for detecting cues of fatigue or pain. Such a system could help identify the need for a break and adjust working timetables. These are essential to minimize development of musculoskeletal conditions, a common problem in manufacturing industries. In similar contexts, the number of sensors could be reduced to fit the specific activities and relevant movements.

Another active area of application is in healthcare beyond the management of chronic pain. For instance, in [5], limb movement was assessed to screen perinatal stroke in infants, while arm movement was analyzed to track everyday rehabilitation progress of stroke patients [1]. For these, integration of our methods demonstrated in Chapter 6 in the system could help establish the link between the type of activity/movement and the behavior category-of-interest (*e.g.* good or poor rehabilitation engagement in [1], and even the level of pain or anxiety in the future). Such activity-aware functions could allow more in-depth understanding of the patient and generate opportunities for richer personalized support.

7.3 Limitations and Future Work

In this section, we discuss the general limitations the studies conducted so far on this topic face and the potential ideas for future works.

7.3.1 The Focus on a Coarse Language of Protective Behavior

In this thesis, we considered detecting protective behavior as a single unique class. However, the work by Keefe *et al.* [21], Sullivan *et al.* [31], and the discussion in [12] show different types of protective behavior with various physical and psychological contexts, which may provide better insights on how to adapt support offered by technology. Hence, a future step for this study could be exploring the possibility to discriminate between different classes of protective behavior.

For example, while guarding a body part appears to be related and appearing more in specific kinds of movement, stiffness of the same body part usually persists across different movements and time. Discriminating guarding against stiffness may help a model decide when to suggest extra exercises to reduce stiffness vs. when to provide support and exposure to a movement in a way that help overcome fear.

From a machine learning perspective, this suggests that two temporal scales should be considered in our modeling, one as in our study to explore the movement during a local period of the day to track the types of activity and the body part of interest for the behavior judgment. A broader timescale could help instead track if the anomalous movement behavior persists over days/weeks/months etc., and a summary

of its existence across a variety of activities. For such, the active development of the research in multiscale machine learning [132, 133, 134, 135, 136] would provide rich knowledge and baselines to help use achieve the goal.

An interesting point from the emerging work of Williams *et al.* [137] is the discovery of a hierarchy of behavior that physiotherapists observed. That is, the flow vs. no-flow at a higher-level being defined by a smooth speed across elements of the movement. From the perspective of protective behavior detection, this raises another question on how to sense such higher-level language describing movements, with more categories from the data aspect or refining the engineering of features etc.

7.3.2 Lacking Multi-Modality of Protective Behavior

In our work, we have particularly focused on movement data, with only the first study (presented in Chapter 6) including the sEMG data. The previous analysis of protective behavior is multimodal as our literature review reveals, for even when the person is not performing the activity in an altered way, anomalous muscles activations can indicate fear of movement [39, 138]. Similarly, previous machine learning studies have shown that adding sEMG data to movement data improves the prediction of pain levels [56, 57].

In Chapter 5 and 6, we decided to focus on the movement data alone for two reasons. In Chapter 5, we desired to understand how the data-driven network could learn cues that reflect visual observations of physiotherapists when describing protective behavior. In Chapter 6, we aimed to understand if a network whose configuration reflects the bodily skeleton structure could improve performance.

The question is now, can the above proposed models be extended to include sEMG data? Particularly, it would be interesting to explore how to embed sEMG in the skeleton-like GCN for movement data. Indeed, I have already started to explore this question through my co-supervision of an M.Sc. thesis [139] in machine learning, from considering simple integration of extra nodes of the sEMG data to the existing GCN according to the physical connection of the muscles and the skeleton joints, to using more complex fusion mechanisms. The results show interesting improvement, namely macro F1 scores of 0.83 and 0.88 for the late fusion and central fusion

methods, respectively, vs. 0.81 of our baseline method using movement data alone, and a paper is in preparation.

Protective behavior and in general bodily movements go beyond merely connecting to unique muscle activities. Relevant physiological phenomena such as respiration may help understand the bodily movement. Indeed, in chronic pain, due to anxiety, shallow respiration makes movement more difficult, which is also linked to the increase in muscle tensions. Tensions in facial muscles can also help control other parts of the body, even if they are not directly useful. According to the emerging work of Williams *et al.* [137], breathing patterns and facial expressions are used widely by physiotherapists to inform their estimation of the patients' status and needs, which for us could be incorporated as extra modalities in the next step.

In short, while the majority of physiotherapists expressed enthusiasm in the literature for using technology-enabled platforms for future patient-physio interaction [25, 28], enough room is available for technological advancements, such as including richer modalities (*e.g.*, physiological signals sensing heart rate and skin response) during data collection and modeling, and enabling oral interaction with the patient for better engagement.

7.3.3 The Lack of Data

On one hand, the size of existing datasets is very limited for deep learning and ubiquitous computing research in chronic pain management. On the other hand, data collected in a research facility provide only a limited understanding of the movement behavior and capabilities of people with chronic pain. Additionally, it is widely acknowledged that collecting data from patients is challenging given increasingly strict data protection regulations and privacy issues.

These are the major problems we face as moving into real-world applications. In order to fully leverage the existing data, this thesis followed a traditional practice of adding noise for the purpose of data augmentation [61]. In the broader deep learning community, researchers in other application areas have started to use more advanced modeling techniques to solve the issue of limited data size, with methods proposed like transfer learning via large-scale pretraining phases [140] and zero- and

few-shot learning [141, 142].

Therefore, while we are planning for the collection of multimodal datasets with a larger size that considers various living scenarios, studies for the next step may first explore how may we alleviate the lack of data and possibly lack of annotation from the modeling perspective. Additionally, new challenges for modeling will also arise given the arrival of new datasets, where different numbers/types of sensors may be used for data collection and data may come from different environments. To make the model work across these different settings, the promising approaches could be domain adaptation methods [143, 144, 145] and models that consider the varying quality and even missing of sensors [146, 147].

7.3.4 The Dependence on Manual Annotation

Methods proposed in the above chapters are all supervised-learning methods that rely on manual annotations, particularly the *ground truth* majority-voted from domain-expert ratings of protective behavior. While in our emerging work reported in Appendix A we have explored the method for better fitting with multiple annotators without using the single ground truth, it remains an open question about how to enable the model to achieve the performance of human experts without full annotations or with a few annotations. This is essential, as providing annotation to a future large in-the-wild dataset could be extremely cumbersome even if doable. To address this question, we may proceed from the modeling side with methods like few-short learning [141] and weakly-supervised learning [148].

7.3.5 The Use of a Large IMUs Network

For most experiments reported in this thesis, a set of 18 IMUs was assumed to be available to provide data of the full-body graph (22 joints). So many IMUs are not usually directly taped to the body, and we do not expect this to be the case when the system is deployed. In fact, ubiquitous motion capture suits that facilitate sensor wearability, *e.g.* the MetaMotion IGS-190 [92] (used for the EmoPain dataset) and Xsens MVN [149], have been around for a long time. Both systems are integrated, wireless, and consider users' comfort so that are technically suitable for the sensing

needed in daily life.

However, such motion capture systems are still expensive even though the IMU sensors are becoming cheaper, more accurate, and wearable (*e.g.*, invisible, washable, or transferable between clothes [83]). Although it is out of the scope of this thesis to develop cheaper suits or examine how to integrate sensors into patients' clothes, it remains an open area for hardware developers and fashion designers to propose better solutions. Additionally, the progress in ubiquitous computing (as our work) may inspire further advances in hardware development, a very active area where we saw *e.g.* the integration of multiple sensors in sports garments. Studies with clinicians and patients show that such advancement is very desirable to help manage the conditions [44, 45]. Hopefully, our research may further augment the future wearable devices with capabilities of protective behavior detection and extend applications to the rehabilitation and clinical contexts.

The original aim of this thesis is, with a large set of sensors, to understand what is feasible and then explore how to improve it. Thereon, several recent studies have aimed to combine sparse IMUs or just accelerometers (less than 6 sensors) and visual information to reconstruct full-body motions in the wild[150, 151]. Given the highest performance of protective behavior detection is achieved by using full-body graphs as seen in Chapter 6, we can follow these works to acquire full-body movement data in less constrained settings using a smaller sensor set.

Bibliography

[1] Shane Halloran, Lin Tang, Yu Guan, Jian Qing Shi, and Janet Eyre. Remote monitoring of stroke patients' rehabilitation using wearable accelerometers. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC)*, pages 72–77, 2019.

[2] Deepti Aggarwal, Bernd Ploderer, Thuong Hoang, Frank Vetere, and Mark Bradford. Physiotherapy over a distance: The use of wearable technology for video consultations in hospital settings. *ACM Transactions on Computing for Healthcare*, 1(4):1–29, 2020.

[3] Dingwen Li, Jay Vaidya, Michael Wang, Ben Bush, Chenyang Lu, Marin Kollef, and Thomas Bailey. Feasibility study of monitoring deterioration of outpatients using multimodal data collected by wearables. *ACM Transactions on Computing for Healthcare*, 1(1):1–22, 2020.

[4] Luca Canzian and Mirco Musolesi. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 1293–1304, 2015.

[5] Yan Gao, Yang Long, Yu Guan, Anna Basu, Jessica Baggaley, and Thomas Ploetz. Towards reliable, automated general movement assessment for perinatal stroke screening in infants using wearable accelerometers. *Proceedings of the ACM on Interactive, Mobile, Wearable and ubiquitous technologies (IMWUT)*, 3(1):1–22, 2019.

[6] Catherine B Johannes, T Kim Le, Xiaolei Zhou, Joseph A Johnston, and Robert H Dworkin. The prevalence of chronic pain in united states adults: results of an internet-based survey. *The Journal of Pain*, 11(11):1230–1239, 2010.

[7] Harald Breivik, Beverly Collett, Vittorio Ventafridda, Rob Cohen, and Derek Gallacher. Survey of chronic pain in europe: prevalence, impact on daily life, and treatment. *European journal of pain*, 10(4):287–333, 2006.

[8] Chongyang Wang, Temitayo A Olugbade, Akhil Mathur, Amanda C De C Williams, Nicholas D Lane, and Nadia Bianchi-Berthouze. Chronic pain protective behavior detection with deep learning. *ACM Transactions on Computing for Healthcare*, 2(3):1–24, 2021.

[9] Chongyang Wang, Temitayo A Olugbade, Akhil Mathur, Amanda C De C. Williams, Nicholas D Lane, and Nadia Bianchi-Berthouze. Recurrent network based automatic detection of chronic pain protective behavior using mocap and semg data. In *Proceedings of the 23rd international symposium on wearable computers*, pages 225–230, 2019.

[10] Chongyang Wang, Min Peng, Temitayo A Olugbade, Nicholas D Lane, Amanda C De C Williams, and Nadia Bianchi-Berthouze. Learning temporal and bodily attention in protective movement behavior detection. In *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 324–330. IEEE, 2019.

[11] Chongyang Wang, Yuan Gao, Akhil Mathur, Amanda C De C. Williams, Nicholas D Lane, and Nadia Bianchi-Berthouze. Leveraging activity recognition to enable protective behavior detection in continuous data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–27, 2021.

[12] Min SH Aung, Sebastian Kaltwang, Bernardino Romera-Paredes, Brais Martinez, Aneesha Singh, Matteo Cella, Michel Valstar, Hongying Meng, Andrew

Kemp, Moshen Shafizadeh, et al. The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal emopain dataset. *IEEE transactions on affective computing*, 7(4):435–451, 2015.

[13] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1533–1540, 2016.

[14] Francisco Javier Ordóñez Morales and Daniel Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 92–99, 2016.

[15] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM international symposium on wearable computers*, pages 56–63, 2018.

[16] Vishvak S Murahari and Thomas Plötz. On attention models for human activity recognition. In *Proceedings of the 2018 ACM international symposium on wearable computers*, pages 100–103, 2018.

[17] Shuochao Yao, Yiran Zhao, Shaohan Hu, and Tarek Abdelzaher. Quality-deepsense: Quality-aware deep learning framework for internet of things applications with sensor-temporal attention. In *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning*, pages 42–47, 2018.

[18] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, 2014.

[19] Temitayo A Olugbade, Nadia Bianchi-Berthouze, Nicolai Marquardt, and Amanda C de C Williams. Human observer and automatic assessment of movement related self-efficacy in chronic pain: from exercise to functional activity. *IEEE Transactions on Affective Computing*, 11(2):214–229, 2018.

[20] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. Globalfusion: A global attentional deep learning framework for multisensor information fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–27, 2020.

[21] Francis J Keefe and Andrew R Block. Development of an observation method for assessing pain behavior in chronic low back pain patients. *Behavior therapy*, 1982.

[22] Ali Asghari and Michael K Nicholas. Pain self-efficacy beliefs and pain behaviour. a prospective study. *Pain*, 94(1):85–100, 2001.

[23] Johan WS Vlaeyen and Steven J Linton. Fear-avoidance and its consequences in chronic musculoskeletal pain: a state of the art. *Pain*, 85(3):317–332, 2000.

[24] Aneesha Singh, Annina Klapper, Jinni Jia, Antonio Fidalgo, Ana Tajadura-Jiménez, Natalie Kanakam, Nadia Bianchi-Berthouze, and Amanda Williams. Motivating people with chronic pain to do physical activity: opportunities for technology design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2803–2812, 2014.

[25] Aneesha Singh, Stefano Piana, Davide Pollarolo, Gualtiero Volpe, Giovanna Varni, Ana Tajadura-Jimenez, Amanda CdeC Williams, Antonio Camurri, and Nadia Bianchi-Berthouze. Go-with-the-flow: tracking, analysis and sonification of movement and breathing to build confidence in activity despite chronic pain. *Human–Computer Interaction*, 31(3-4):335–383, 2016.

[26] Aneesha Singh, Nadia Bianchi-Berthouze, and Amanda CdeC Williams. Supporting everyday function in chronic pain using wearable technology. In

Proceedings of the 2017 CHI Conference on human factors in computing systems, pages 3903–3915, 2017.

[27] Sergio Felipe, Aneesha Singh, Caroline Bradley, Amanda CdeC Williams, and Nadia Bianchi-Berthouze. Roles for personal informatics in chronic pain. In *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 161–168. IEEE, 2015.

[28] Tali Swann-Sternberg, Aneesha Singh, Nadia Bianchi-Berthouze, and Amanda Williams. User needs for technology supporting physical activity in chronic pain. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 2441–2446. ACM, 2012.

[29] MS Hane Aung, Nadia Bianchi-Berthouze, Paul Watson, and AC de C Williams. Automatic recognition of fear-avoidance behavior in chronic pain physical rehabilitation. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 158–161, 2014.

[30] Johan WS Vlaeyen, Stephen Morley, and Geert Crombez. The experimental analysis of the interruptive, interfering, and identity-distorting effects of chronic pain. *Behaviour research and therapy*, 86:23–34, 2016.

[31] Michael JL Sullivan, Pascal Thibault, André Savard, Richard Catchlove, John Kozey, and William D Stanish. The influence of communication goals and physical demands on different dimensions of pain behavior. *Pain*, 125(3):270–277, 2006.

[32] Jesús Joel Rivas, Felipe Orihuela-Espina, Luis Enrique Sucar, Amanda Williams, and Nadia Bianchi-Berthouze. Automatic recognition of multiple affective states in virtual rehabilitation by exploiting the dependency relationships. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–7. IEEE, 2019.

[33] Uttaran Bhattacharya, Trisha Mittal, Rohan Chandra, Tanmay Randhavane, Aniket Bera, and Dinesh Manocha. Step: Spatial temporal graph convolutional

networks for emotion perception from gaits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1342–1350, New York, USA, 2020. AAAI.

[34] ACII. Pd&emopain workshop, 2019. http://www.emo-pain.ac.uk/PDEmoPain19/, Sidst set 21/05/2022.

[35] FG. Emopain challenge, 2020. https://wangchongyang.ai/EmoPainChallenge2020/, Sidst set 21/05/2022.

[36] ACII. Emopain challenge, 2021. http://www.casapaganini.it/entimement/workshops/2021/Workshop2021_Home.php, Sidst set 21/05/2022.

[37] Irene Tracey and M Catherine Bushnell. How neuroimaging studies have challenged us to rethink: is chronic pain a disease? *The journal of pain*, 10(11):1113–1120, 2009.

[38] Temitayo A Olugbade, Aneesha Singh, Nadia Bianchi-Berthouze, Nicolai Marquardt, Min SH Aung, and Amanda C De C Williams. How can affect be detected and represented in technological support for physical rehabilitation? *ACM Transactions on Computer-Human Interaction (TOCHI)*, 26(1):1–29, 2019.

[39] Paul J Watson, C Kerry Booker, and Chris J Main. Evidence for the role of psychological factors in abnormal paraspinal activity in patients with chronic low back pain. *Journal of Musculoskeletal Pain*, 5(4):41–56, 1997.

[40] Joan M Romano, Mark P Jensen, Judith A Turner, Amy B Good, and Hyman Hops. Chronic pain patient-partner interactions: Further support for a behavioral model of chronic pain. *Behavior Therapy*, 31(3):415–440, 2000.

[41] UK Pain Messages. The pain consortium. In *Pain News*, pages 21—-22, 2016.

[42] Robert N Jamison. Are we really ready for telehealth cognitive behavioral therapy for pain? *Pain*, 158(4):539–540, 2017.

[43] Karon F Cook, Francis Keefe, Mark P Jensen, Toni S Roddey, Leigh F Callahan, Dennis Revicki, Alyssa M Bamer, Jiseon Kim, Hyewon Chung, Rana Salem, et al. Development and validation of a new self-report measure of pain behaviors. *PAIN®*, 154(12):2867–2876, 2013.

[44] Enrica Papi, Athina Belsi, and Alison H McGregor. A knee monitoring device and the preferences of patients living with osteoarthritis: a qualitative study. *BMJ open*, 5(9):e007980, 2015.

[45] Enrica Papi, Ged M Murtagh, and Alison H McGregor. Wearable technologies in osteoarthritis: a qualitative study of clinicians' preferences. *BMJ open*, 6(1):e009544, 2016.

[46] Wilbert E Fordyce, David Lansky, Donald A Calsyn, John L Shelton, Walter C Stolov, and Daniel L Rock. Pain measurement and pain behavior. *Pain*, 18(1):53–69, 1984.

[47] Beatrice De Gelder. Why bodies? twelve reasons for including bodily expressions in affective neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3475–3484, 2009.

[48] Andrea Kleinsmith and Nadia Bianchi-Berthouze. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing*, 4(1):15–33, 2012.

[49] Andrea Kleinsmith, Nadia Bianchi-Berthouze, and Anthony Steed. Automatic recognition of non-acted affective postures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 41(4):1027–1038, 2011.

[50] Nikolaos Savva, Alfonsina Scarinzi, and Nadia Bianchi-Berthouze. Continuous recognition of player's affective body expression as dynamic quality of aesthetic experience. *IEEE Transactions on Computational Intelligence and AI in games*, 4(3):199–212, 2012.

[51] Fahd Albinali, Matthew S Goodwin, and Stephen S Intille. Recognizing stereotypical motor movements in the laboratory and classroom: a case study with children on the autism spectrum. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 71–80, 2009.

[52] Jyoti Joshi, Roland Goecke, Gordon Parker, and Michael Breakspear. Can body expressions contribute to automatic depression analysis? In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE, 2013.

[53] David K Ahern, Michael J Follick, James R Council, Nancy Laser-Wolston, and Henry Litchman. Comparison of lumbar paravertebral emg patterns in chronic low back pain patients and non-patient controls. *Pain*, 34(2):153–160, 1988.

[54] Helena Grip, Fredrik Ohberg, Urban Wiklund, Ylva Sterner, J Stefan Karlsson, and Björn Gerdle. Classification of neck movement patterns related to whiplash-associated disorders using neural networks. *IEEE transactions on information technology in biomedicine*, 7(4):412–418, 2003.

[55] James P Dickey, Michael R Pierrynowski, Drew A Bednar, and Simon X Yang. Relationship between pain and vertebral motion in chronic low-back pain subjects. *Clinical Biomechanics*, 17(5):345–352, 2002.

[56] Temitayo A Olugbade, MS Hane Aung, Nadia Bianchi-Berthouze, Nicolai Marquardt, and Amanda C Williams. Bi-modal detection of painful reaching for chronic pain rehabilitation systems. In *Proceedings of the 16th international conference on multimodal interaction*, pages 455–458, 2014.

[57] Temitayo A Olugbade, Nadia Bianchi-Berthouze, Nicolai Marquardt, and Amanda C Williams. Pain level recognition using kinematics and muscle activity for physical rehabilitation in chronic pain. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 243–249. IEEE, 2015.

[58] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th international conference on mobile computing, applications and services*, pages 197–205. IEEE, 2014.

[59] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[60] Yu Guan and Thomas Plötz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–28, 2017.

[61] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. *19th ACM International Conference on Multimodal Interaction (ICMI)*, 1(1):216–220, 2017.

[62] Nastaran Mohammadian Rad and Cesare Furlanello. Applying deep learning to stereotypical motor movement detection in autism spectrum disorders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 1235–1242. IEEE, 2016.

[63] Nastaran Mohammadian Rad, Seyed Mostafa Kia, Calogero Zarbo, Twan van Laarhoven, Giuseppe Jurman, Paola Venuti, Elena Marchiori, and Cesare Furlanello. Deep learning for automatic stereotypical motor movement detection using wearable sensors in autism spectrum disorders. *Signal Processing*, 144:180–191, 2018.

[64] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[65] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*, pages 233–240. IEEE, 2010.

[66] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pages 108–109. IEEE, 2012.

[67] Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, and Gerhard Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2):42–50, 2008.

[68] Michelle Karg, Ali-Akbar Samadani, Rob Gorbet, Kolja Kühnlenz, Jesse Hoey, and Dana Kulić. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Transactions on Affective Computing*, 4(4):341–359, 2013.

[69] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016.

[70] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[71] Marc Bachlin, Daniel Roggen, Gerhard Troster, Meir Plotnik, Noit Inbar, Inbal Meidan, Talia Herman, Marina Brozgol, Eliya Shaviv, Nir Giladi, et al. Potentials of enhanced context awareness in wearable assistants for parkinson's disease patients with the freezing of gait syndrome. In *2009 International Symposium on Wearable Computers*, pages 123–130. IEEE, 2009.

[72] Matthew S Goodwin, Marzieh Haghighi, Qu Tang, Murat Akcakaya, Deniz Erdogmus, and Stephen Intille. Moving towards a real-time system for automatically recognizing stereotypical motor movements in individuals on the autism spectrum using wireless accelerometry. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 861–872, 2014.

[73] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek 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*, pages 351–360, 2017.

[74] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 127–140, 2015.

[75] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, pages 2224–2232, 2015.

[76] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[77] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[78] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[79] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1227–1236, 2019.

[80] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3595–3603, 2019.

[81] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7912–7921, 2019.

[82] Xia Qingxin, Atsushi Wada, Joseph Korpela, Takuya Maekawa, and Yasuo Namioka. Unsupervised factory activity recognition with wearable sensors using process instruction information. *Proceedings of the ACM on Interactive, Mobile, Wearable and ubiquitous technologies (IMWUT)*, 3(2):1–23, 2019.

[83] Sung-Won Kang, Hyeob Choi, Hyung-Il Park, Byoung-Gun Choi, Hyobin Im, Dongjun Shin, Young-Giu Jung, Jun-Young Lee, Hong-Won Park, Sukyung Park, et al. The development of an imu integrated clothes for postural monitoring using conductive yarn and interconnecting technology. *Sensors*, 17(11):2560, 2017.

[84] JHM Bergmann and AH McGregor. Body-worn sensor design: what do patients and clinicians want? *Annals of biomedical engineering*, 39(9):2299–2312, 2011.

[85] Jiyoung Lee, Seungryong Kim, Sunok Kim, Jungin Park, and Kwanghoon Sohn. Context-aware emotion recognition networks. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision, pages 10143–10152, 2019.

[86] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE multimedia*, 19(03):34–41, 2012.

[87] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[88] Qingqiu Huang, Yu Xiong, and Dahua Lin. Unifying identification and context learning for person recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2217–2225, 2018.

[89] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4804–4813, 2015.

[90] Wallace Lawson, Laura Hiatt, and J Trafton. Leveraging cognitive context for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 381–386, 2014.

[91] J Gregory Trafton, Laura M Hiatt, Anthony M Harrison, Franklin P Tamborello, Sangeet S Khemlani, and Alan C Schultz. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2013.

[92] The MetaMotion IGS-190 motion capture suit. Metamotion. In *https://metamotion.com/gypsy/gypsy-gyro.htm*. online, 2021.

[93] N. Lawrence. Matlab motion capture toolbox. In *http://inverseprobability.com/mocap/*. online, 2009.

[94] Paul J Watson, C Kerry Booker, and Chris J Main. Evidence for the role of psychological factors in abnormal paraspinal activity in patients with chronic low back pain. *Journal of Musculoskeletal Pain*, 5(4):41–56, 1997.

[95] Chongyang Wang. Animation of c16d, 2022. https://youtu.be/c4VuXHDIn58, Sidst set 21/05/2022.

[96] Chongyang Wang. Animation of p14n, 2022. https://youtu.be/Q1VIzYQZ1Co, Sidst set 21/05/2022.

[97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[98] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

[99] Diarmuid Denneny, Annina Frijdal, Nadia Bianchi-Berthouze, Jim Greenwood, Rebecca McLoughlin, Katrine Petersen, Aneesha Singh, and Amanda C de C Williams. The application of psychologically informed practice: observations of experienced physiotherapists working with people with chronic pain. *Physiotherapy*, 106:163–173, 2020.

[100] Sojeong Ha and Seungjin Choi. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. *International Joint Conference on Neural Networks (IJCNN)*, 1(1):381–388, 2016.

[101] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. *IEEE conference on computer vision and pattern recognition (CVPR)*, 1(1):4305–4314, 2015.

[102] Diederik P Kingma and Jimmy. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1(1):1, 2014.

[103] Temitayo A Olugbade. *Automatic Monitoring of Physical Activity Related Affective States for Chronic Pain Rehabilitation*. PhD thesis, UCL (University College London), 2018.

[104] Kenneth O McGraw and Seok P Wong. Forming inferences about some intraclass correlation coefficients. *Psychological methods*, 1(1):1, 1996.

[105] Kevin A Hallgren. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology*, 8(1):1, 2012.

[106] Terry K Koo and Mae Y Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine*, 15(2):155–163, 2016.

[107] David Naranjo-Hernández, Javier Reina-Tosina, and Laura M Roa. Sensor technologies to manage the physiological traits of chronic pain: A review. *Sensors*, 20(2):365, 2020.

[108] Rasha M Al-Eidan, Hend Al-Khalifa, and AbdulMalik Al-Salman. Deep-learning-based models for pain recognition: A systematic review. *Applied Sciences*, 10(17):5984, 2020.

[109] Sophie Skach, Rebecca Stewart, and Patrick GT Healey. Sensing social behavior with smart trousers. *IEEE Pervasive Computing*, 20(3):30–40, 2021.

[110] Lucy E Dunne and Jamie A Ward. Making sensors, making sense, making stimuli: The state of the art in wearables research from iswc 2019. *IEEE Pervasive Computing*, 19(1):87–91, 2020.

[111] Mir Mohammed Assadullah. Barriers to artificial intelligence adoption in healthcare management: A systematic review. *Available at SSRN 3530598*, 2019.

[112] Yi Li, Shreya Ghosh, and Jyoti Joshi. Plaan: Pain level assessment with anomaly-detection based network. *Journal on Multimodal User Interfaces*, pages 1–14, 2021.

[113] Xinhui Yuan and Marwa Mahmoud. Alanet: Autoencoder-lstm for pain and protective behaviour detection. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 824–828. IEEE, 2020.

[114] Tahiya Chowdhury, Murtadha Aldeer, Shantanu Laghate, and Jorge Ortiz. Cadence: A practical time-series partitioning algorithm for unlabeled iot sensor streams. *arXiv preprint arXiv:2112.03360*, 2021.

[115] NE Gold, Chongyang Wang, Temitayo Olugbade, N Berthouze, and A Williams. P (l) aying attention: Multi-modal, multi-temporal music control. In *Proceedings-International Conference on New Interfaces for Musical Expression*. NIME, 2020.

[116] Tomasz Sipko, Edmund Glibowski, Katarzyna Barczyk-Pawelec, and Michał Kuczyński. The effect of chronic pain intensity on sit-to-stand strategy in patients with herniated lumbar disks. *Journal of manipulative and physiological therapeutics*, 39(3):169–175, 2016.

[117] Shuochao Yao, Yiran Zhao, Huajie Shao, Dongxin Liu, Shengzhong Liu, Yifan Hao, Ailing Piao, Shaohan Hu, Su Lu, and Tarek F Abdelzaher. Sadeepsense: Self-attention deep learning framework for heterogeneous on-device sensors in internet of things applications. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1243–1251. IEEE, 2019.

[118] Timo Sztyler and Heiner 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)*, pages 1–9. IEEE, 2016.

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

[120] Esam Ghaleb, André Mertens, Stylianos Asteriadis, and Gerhard Weiss. Skeleton-based explainable bodily expressed emotion recognition through graph convolutional networks. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 2021.

[121] Fangkai Yang, Wenjie Yin, Tetsunari Inamura, Mårten Björkman, and Christopher Peters. Group behavior recognition using attention-and graph-based neural networks. In *ECAI 2020*, pages 1626–1633. IOS Press, 2020.

[122] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*, 2017.

[123] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[124] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018.

[125] Chris Drumnond. Class imbalance and cost sensitivity: Why undersampling beats oversampling. In *ICML-KDD 2003 Workshop: Learning from Imbalanced Datasets*, 2003.

[126] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384, 2016.

[127] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1(1):9268–9277, 2019.

[128] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[129] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[130] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[131] MMN Bieńkiewicz, Andrii Smykovskyi, Temitayo Olugbade, Stefan Janaqi, Antonio Camurri, Nadia Bianchi-Berthouze, Mårten Björkman, and Benoît G Bardy. Bridging the gap between emotion and joint action. *Neuroscience & Biobehavioral Reviews*, 2021.

[132] Temitayo Olugbade, Nicolas Gold, Amanda C de C Williams, and Nadia Bianchi-Berthouze. A movement in multiple time neural network for automatic detection of pain behaviour. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, pages 442–445, 2020.

[133] Yuichi Yamashita and Jun Tani. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS computational biology*, 4(11):e1000220, 2008.

[134] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, pages 1863–1871. PMLR, 2014.

[135] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[136] Katherine Metcalf and David Leake. Unsupervised hierarchical temporal abstraction by simultaneously learning expectations and representations. In *IJCAI*, pages 3144–3150, 2019.

[137] Amanda C de C Williams, Raffaele Buono, Temitayo Olugbade, Nicolas Gold, and Nadia Bianchi-Berthouze. Guarding and flow in the movements of people with chronic pain: a qualitative study of physiotherapists' judgements. *Deliverable 4.4 for EU Horizon FET 2020 EnTimeMent Project*, 2022.

[138] Paul J Watson, CK Booker, CJ Main, and ACN Chen. Surface electromyography in the identification of chronic low back pain patients: the development of the flexion relaxation ratio. *Clinical Biomechanics*, 12(3):165–171, 1997.

[139] Guanting Cen. *Exploring Multimodal Fusion for Protective Behavior Detection in Continuous Data*. UCL Master Thesis, 2021.

[140] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019.

[141] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[142] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020.

[143] Akhil Mathur, Nadia Berthouze, and Nicholas D Lane. Unsupervised domain adaptation under label space mismatch for speech classification. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2020, pages 1271–1275. International Speech Communication Association (ISCA), 2020.

[144] Shaoduo Gan, Akhil Mathur, Anton Isopoussu, Fahim Kawsar, Nadia Berthouze, and Nicholas Lane. Fruda: Framework for distributed adversarial domain adaptation. *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[145] Akhil Mathur. *Scaling Machine Learning Systems using Domain Adaptation*. PhD thesis, UCL (University College London), 2020.

[146] Shengzhong Liu, Shuochao Yao, Yifei Huang, Dongxin Liu, Huajie Shao, Yiran Zhao, Jinyang Li, Tianshi Wang, Ruijie Wang, Chaoqi Yang, et al. Handling missing sensors in topology-aware iot applications with gated graph neural network. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–31, 2020.

[147] Jesús Joel Rivas, Felipe Orihuela-Espina, Luis Enrique Sucar, and Nadia Bianchi-Berthouze. Dealing with a missing sensor in a multilabel and multimodal automatic affective states recognition system. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8. IEEE, 2021.

[148] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

[149] The Xsens MVN motion capture suit. Xsens. *https://www.xsens.com/motion-capture*, 2021.

[150] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Transactions on Graphics (ToG)*, 30(3):1–12, 2011.

[151] Timo Von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. Human pose estimation from video and imus. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1533–1547, 2016.

[152] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[153] Simon K Warfield, Kelly H Zou, and William M Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903–921, 2004.

[154] Thomas A Lampert, André Stumpf, and Pierre Gançarski. An empirical study into annotator agreement, ground truth estimation, and algorithm evaluation. *IEEE Transactions on Image Processing*, 25(6):2557–2572, 2016.

[155] Davood Karimi, Haoran Dou, Simon K Warfield, and Ali Gholipour. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, 65:101759, 2020.

[156] Hongying Meng, Andrea Kleinsmith, and Nadia Bianchi-Berthouze. Multi-score learning for affect recognition: the case of body postures. In *International Conference on Affective Computing and Intelligent Interaction*, pages 225–234. Springer, 2011.

[157] Ninghang Hu, Gwenn Englebienne, Zhongyu Lou, and Ben Kröse. Learning to recognize human activities using soft labels. *IEEE transactions on pattern analysis and machine intelligence*, 39(10):1973–1984, 2016.

[158] Chengjiang Long, Gang Hua, and Ashish Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3000–3007, 2013.

[159] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2839–2847, 2015.

[160] Jennifer Healey. Recording affect in the field: Towards methods and metrics for improving ground truth labels. In *International conference on affective computing and intelligent interaction*, pages 107–116. Springer, 2011.

[161] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. Who said what: Modeling individual labelers improves classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[162] Wei Ji, Shuang Yu, Junde Wu, Kai Ma, Cheng Bian, Qi Bi, Jingjing Li, Hanruo Liu, Li Cheng, and Yefeng Zheng. Learning calibrated medical image segmentation via multi-rater agreement modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12341–12351, 2021.

[163] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine learning*, 95(3):291–327, 2014.

[164] Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 932–939. JMLR Workshop and Conference Proceedings, 2010.

[165] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11244–11253, 2019.

[166] Le Zhang, Ryutaro Tanno, Mou-Cheng Xu, Chen Jin, Joseph Jacob, Olga Ciccarelli, Frederik Barkhof, and Daniel C Alexander. Disentangling human error from the ground truth in segmentation of medical images. *arXiv preprint arXiv:2007.15963*, 2020.

[167] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L Ball, et al. Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957*, 2017.

[168] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):1–14, 2017.

[169] A Kendall, V Badrinarayanan, and R Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *British Machine Vision Conference*, 2017.

[170] Lingni Ma, Jörg Stückler, Christian Kerl, and Daniel Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 598–605. IEEE, 2017.

[171] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *arXiv preprint arXiv:2006.09239*, 2020.

[172] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[173] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2931–2940, 2019.

[174] Yichen Shen, Zhilu Zhang, Mert R Sabuncu, and Lin Sun. Real-time uncertainty estimation in computer vision via uncertainty-aware distribution distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 707–716, 2021.

[175] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[176] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[177] Igor Lovchinsky, Alon Daks, Israel Malkin, Pouya Samangouei, Ardavan Saeedi, Yang Liu, Swami Sankaranarayanan, Tomer Gafner, Ben Sternlieb, Patrick Maher, et al. Discrepancy ratio: Evaluating model performance when even experts disagree on the truth. In *International Conference on Learning Representations*, 2019.

[178] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *arXiv preprint arXiv:2006.04388*, 2020.

[179] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

[180] Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, and Heng Huang. Multi-horizon time series forecasting with temporal attention learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2527–2535, 2019.

[181] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2021.

[182] Jordi de La Torre, Domenec Puig, and Aida Valls. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154, 2018.

[183] Jacob Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

[184] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[185] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[186] Chongyang Wang, Yuan Gao, Chenyou Fan, Junjie Hu, Tin Lun Lam, Nicholas D Lane, and Nadia Bianchi-Berthouze. Agreementlearning: An end-to-end framework for learning with multiple annotators without groundtruth. *arXiv preprint arXiv:2109.03596*, 2021.

Appendix A

Learning from Multiple Annotators without Objective Ground Truth

The main study chapters of this thesis demonstrate interesting progresses we have achieved in the detection of protective behavior in pre-segmented activity instances or continuous data of various activities, given the ground truth labels produced in a majority-voting manner.

However, a ground truth produced through majority-voting is not always very representative. In addition, it comes with information loss for the training of a model, given the existing multiple annotations of different experts are ignored. Furthermore, for scenarios where an objective ground truth is missing and the opinions of domain experts play a key role, learning from the majority-voted ground truth may pose a bottleneck on model performance when evaluation is conducted on all the annotators.

To address this limitation, one could let the model learn from all annotators. However, without a proper regularization, the model learning with all the annotations is vulnerable to the disagreements and imbalances present in the annotations. In this appendix chapter, we present our emerging study aiming at targeting these issues.

An illustrative diagram summarizing the existing practices to address this problem and our proposal for learning from multiple annotators is shown in Figure A.1. Our contributions are summarized as follows.

- We propose a novel agreement learning model to directly leverage the agreement information stored in the annotations from multiple annotators to regularize the

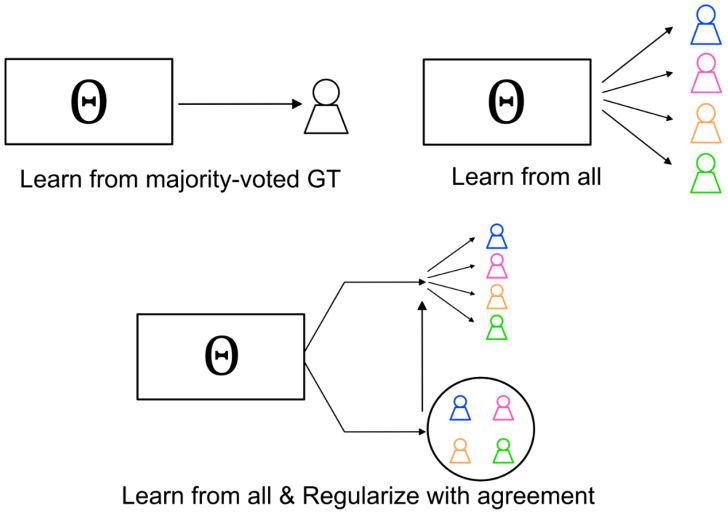


Figure A.1: Unlike the methods that learn from the majority-voted ground truth or all the annotations directly, the proposed model regularizes the classifier that fits with all the annotators with the estimated agreement information between annotators.

behavior of the classifier that learns from them.

- To improve the robustness of our model, we propose a general agreement distribution and an agreement regression loss to model the uncertainty in annotations.

- To regularize the classifier, we propose a regularization function to tune the classifier to better agree with all the annotators.

- Our method noticeably improves existing backbones for better agreement levels with all the annotators on classification tasks in two medical datasets, involving data of body movement sequences that we used in previous chapters and bone X-rays to verify the effectiveness of our method in another domain.

A.1 Motivation

There exist difficulties for model development in applications where the objective ground truth is difficult to establish or ambiguous merely given the input data itself. That is, the decision-making, *i.e.* the detection, classification, and segmentation process, is based on not only the presented data but also the expertise or experiences of the annotator. However, the disagreements existed in the annotations hinder the

definition of a good single ground truth. Therefore, an important part of supervise learning for such applications is to fit the model to the domain experts' annotations.

In supervised learning, the input normally comprises pairs of (x_i, l_i), where x_i and l_i are respectively data and the label of i-th sample. Given the annotations provided by multiple annotators, typical methods aim to provide a single set of ground truth label. Therein, a common practice is to aggregate these multiple annotations with majority voting [152]. However, majority-voting could misrepresent the data instances where the disagreement between different annotators is high. This is particularly harmful for applications where differences in expertise or experiences exist in the annotators.

Except for majority-voting, some have tried to estimate the ground truth label using STAPLE [153] based on Expectation-Maximization (EM) algorithms. Nevertheless, such method is sensitive to the variance in the annotations and the data size [154, 155]. When the number of annotations per x_i is modest, efforts are put into creating models that utilize all the annotations with multi-score learning [156] or soft labels [157]. Recent approaches have instead focused on leveraging or learning the expertise of the annotators while training the model [158, 159, 160, 161, 162, 163, 164, 165, 166]. A basic idea is to refine the classification or segmentation toward the underlying ground truth by modeling the annotators.

In this appendix chapter, we focus on a hard situation when the ground truth is ambiguous to define. On one hand, this could be due to the missing of objective ground truth in a specific scenario. For instance, in the analysis of bodily movement behavior for chronic pain (CP) rehabilitation, the self-awareness of people with CP about their exhibited pain or fear-related behaviors is low, thus physiotherapists play a key role in judging it [27, 25]. However, since the physiotherapists are assessing the behavior on the basis of visual observations, they may disagree on the judgment or ground truth.

On the other hand, the ground truth could be temporarily missing, at a special stage of the task. For example, in abnormality prescreening for bone X-rays, except for abnormalities like fractures and hardware implantation that are obvious and deter-

ministic, other types like degenerative diseases and miscellaneous abnormalities are mainly diagnosed with further medical examinations [167]. That is, at prescreening stage, the opinion of the doctor makes the decision, which could disagree with other doctors or the final medical examination though.

Thereon, unlike the traditional modeling goal that usually requires the existence of a set of ground truth labels to evaluate the performance, the objective of modeling in this work is to improve the overall fitting between the model and the annotators.

A.2 Related Work

In this section, we review more relevant studies that could provide knowledge and inspirations for this work. As mentioned above, some recent studies aim to model the annotation behavior of each labeller to help refine the decision-making of the model to be as close as to the underlying ground truth. Another set of studies we review provide some knowledge about uncertainty modeling, as we foresee the possible existence of a certain level of uncertainty within the diverse annotations. We further review studies that inform alternative ways for model evaluation without requiring the use of a single set of ground truth labels.

A.2.1 Annotator Modeling

When dealing with multiple annotators, there is a group of studies that aim to let the model better approach the underlying ground truth of the data input by modeling the behavior/expertise/reliability of the annotator. The leveraging or learning of annotators' expertise is usually implemented in a two-step or multiphase manner, or integrated to run simultaneously.

For the first category, one way to acquire the expertise is by referring to the prior knowledge about the annotation, *e.g.* the year of experience of each annotator, and the discussion held on the disagreed annotations. With such prior knowledge, studies in [158, 159, 160] propose to distill the annotations, by deciding which annotator to trust on disagreed samples. The expertise, or behavior of an annotator can also be modeled given the annotation and the data, which could be used to weight each annotator in the training of a classification model [161], or adopted to refine the

segmentation learned from multiple annotators [162].

More close to our problem are the ones that simultaneously model the expertise of annotators while training the classifier. Previous efforts are seen on using probabilistic models [163, 164] driven by EM algorithms, and multi-head models that directly model annotators as confusion matrices estimated in comparison with the underlying ground truth [165, 166]. All these methods consider the existence of an underlying ground truth for each x_i, where the annotations are noisy estimations of it. Furthermore, during the evaluation, such approaches usually compare the model with the objective ground truth (*e.g.*, the biopsy result in cancer screening) or the ground truth agreed by extra or left-out annotators. However, when an objective ground truth does not exist, such evaluations are not possible and there is still the need to understand how to learn from the subjective annotations.

While the idea behind these works may indeed work for applications where the distance between each annotator and the underlying ground truth exists and can be estimated in some ways to refine the decision-making of a model, we argue that in some cases it is (at least temporarily) difficult to assume the existence of the underlying ground truth. For instance, in analyzing protective behavior of people with CP, there isn't a gold standard protocol in judging it, and experts may indeed have different opinions sometimes. For another instance, at the prescreening of bone X-ray abnormality, the instant judgement from the doctor on-duty usually drops the decision, although a disagreement could arise from the possible discussion with other doctors (if there are) or the post-hoc medical examination.

Additionally, it could be less reasonable to rank the annotators, since in real-life practice, each of them is able to carry out the work independently. Thereon, we shift the focus of the model from approaching the underlying ground truth to fitting with all the annotators, where each annotator is treated equally.

A.2.2 Uncertainty Modeling

Uncertainty modeling is a popular topic in the computer vision domain, especially for tasks of semantic segmentation and object detection. Therein, methods proposed can be categorized into two groups: i) the Bayesian methods, where parameters of the

posterior distribution (*e.g.* mean and variance) of the uncertainty are estimated with Monte Carlo dropout [168, 169, 170] and parametric learning [142, 171] etc.; and ii) 'non-Bayesian' alternatives, where the distribution of uncertainty is learned with ensemble methods [172], variance propagation [173], and knowledge distillation [174] etc.

Except for their complex and time-consuming training or inference strategies, another characteristic of these methods is the dependence on Gaussian or Dirac delta distributions as the prior assumption.

In this work, we consider the uncertainty during our learning of agreement information per data sample. This uncertainty exists as, at the level of each data sample (*e.g.,* a timestep during movement or a single bone X-ray image), different annotators could be inconsistent with their judgements.

Therefore, instead of letting the model learn to estimate the exact level of agreements between annotators across different samples, we make it understand the distribution of the agreement. Additionally, we design the distributional agreement learning without relying on a specific priori, *e.g.* Gaussian distribution.

A.2.3 Model Evaluation without Ground Truth

In the context of modeling multiple annotations without ground truth, typical measures for evaluation are the metrics of agreements. For example, [49] uses metrics of agreement, *e.g.* Cohen's Kappa [175] and Fleiss' Kappa [176], as the way to compare the agreement level between a system and an annotator and the agreement level between other unseen annotators, in a cross-validation manner. However, this method does not consider how to directly learn from all the annotators, and how to evaluate the performance of the model in this case.

To aid such evaluation when compare a model with all the annotators, [177] proposes a metric named discrepancy ratio. In short, the metric compares performances of the model-annotator vs. the annotator-annotator, where the performance can be computed as discrepancy *e.g.* with absolute error, or as agreement *e.g.* with Cohen's kappa.

As we reasoned earlier, the application scenarios targeted in this work face the

missing of object ground truth not only during training but also in evaluation. Thus, traditional metrics like accuracy and F measurements are no longer useable. For PBD, we may be able to consider such an issue during future dataset development, *e.g.* by asking annotators to reach an agreement during their annotations, which is beyond this work.

Additionally, we aim to let the model learn from all the annotators, thus a metric comparing the model with them should be used during evaluation. Therefore, in this work, we use the Cohen's kappa as the agreement calculator together with the method proposed in [177] to evaluate the performance of the model. We refer to this metric as agreement ratio.

A.3 Method

An overview of our proposed agreement learning model is shown in Fig.A.2. The core of our proposed method is to learn to estimate the agreement level between different annotators based on their raw annotations, and simultaneously utilize the agreement-level estimation to regularize the training of the classification task. Therein, different components of the proposed method concern: the learning of agreement levels between annotators, and regularizing the classifier with such information. In testing or inference, the model estimates annotators' agreement level based on the current data input, which is then used to aid the classification.

In this work, we consider the dataset comprising N samples $\mathbf{X} = \{x_i\}_{i=1,...,N}$, with each sample x_i being a timestep in a body movement data sequence or an image. For each sample x_i, r_i^j denotes the annotation provided by j-th annotator, with $\alpha_i \in [0,1]$ being the agreement level computed between annotators. For a binary task, $r_i^j \in \{0,1\}$. With such dataset $\mathcal{D} = \{x_i, r_i^1, ..., r_i^J\}_{i=1,...,N}$, the proposed method aims to improve the fitting of the model with all the annotators. It should be noted that, for each sample x_i, the method does not expect the annotations to be available from all the J annotators.

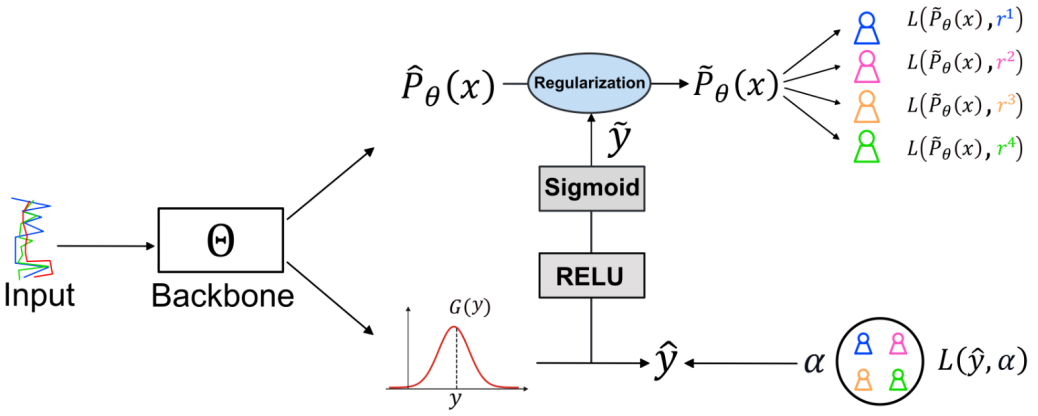


Figure A.2: An overview of the proposed agreement learning model, which comprises i) (above) the classifier stream that fits with all the annotators; and ii) (below) the agreement learning stream that learns to estimate the agreement between annotators and leverage such information to regularize the classifier.

A.3.1 Learning Agreement with Uncertainty Modeling

To enable a robust learning of the agreement information between annotators, we consider modeling the uncertainty that could exist in the annotations. In our scenarios, the uncertainty comes from the annotators' inconsistent judgement exhibited in their annotations across different local data samples, which may not follow specific prior distributions.

Inspired by the study of [178] that proposed to use a general distribution for uncertainty modeling in the bounding box regression of object classification, without relying on any prior distributions, we further propose a general agreement distribution $G(y_i)$ for agreement learning (see the upper part of Figure A.3).

The distribution values (*i.e.*, values along the x-axis of the distribution) are the possible agreement levels y_i between annotators with a range of $[0,1]$, which is further discretized into $\{y_i^0, y_i^1, ...y_i^{n-1}, y_i^n\}$ (*i.e.*, to form the x-axis of the distribution). Here need to note that such range of values does not rely on the number of annotators as $y_i^0 = 0$ represents 'not agreed', and $y_i^n = 1$ represent 'all agreed'. The general agreement distribution has a property $\sum_{k=0}^{n} G(y_i^k) = 1$, which thus can be implemented with a softmax layer with $n+1$ nodes. The number of nodes is a hyperparameter in our method that should be tuned to balance the granularity of possible agreement values between the annotators and the number of trainable

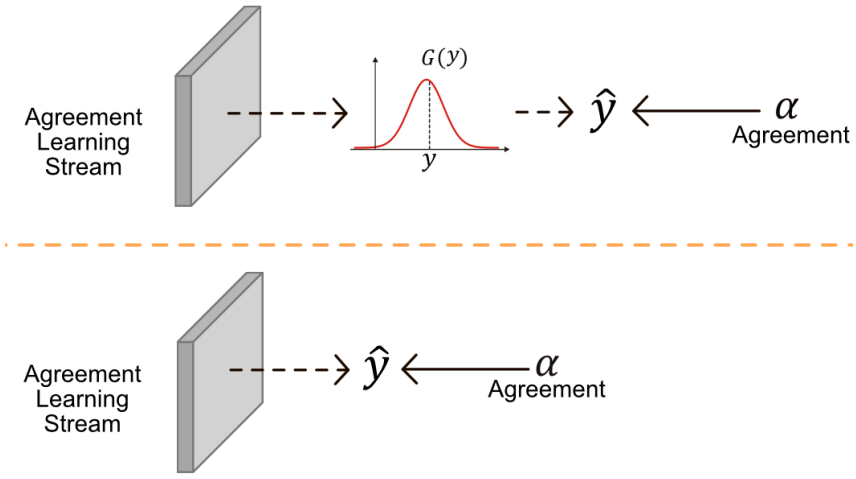


Figure A.3: The learning of the agreement between annotators is modeled with a general agreement distribution using agreement regression loss (above), with the X axis of the distribution being the agreement levels and the Y axis being the respective probabilities. The learning can also be implemented as a linear regression task with RMSE (below).

parameters for training. The predicted agreement $\hat{y}_i$ for regression can be computed as the weighted sum of all the distribution values

$$\hat{y}_i = \sum_{k=0}^{n} G(y_i^k) y_i^k. \tag{A.1}$$

For training the model to predict agreement value $\hat{y}_i$ toward the target agreement α_i, inspired by the effectiveness of quantile regression in understanding the property of conditional distribution [179, 180], we propose a novel Agreement Regression (AR) loss defined by

$$\mathcal{L}_{AR}(\hat{y}_i, \alpha_i) = \max[\alpha_i(\hat{y}_i - \alpha_i), (\alpha_i - 1)(\hat{y}_i - \alpha_i)]. \tag{A.2}$$

Comparing with the original quantile regression loss, the quantile q is replaced with the agreement α_i computed at current input sample x_i. The quantile q is usually fixed for a dataset, as to understand the underlying distribution of the model's output at a given quantile. By replacing q with α_i, together with the design of our general agreement distribution, we optimize the model to focus on the given agreement level dynamically across samples.

In [181], the authors proposed to use the top k values of the distribution and their mean to indicate the shape (flatness) of the distribution, which provides the level of uncertainty in object classification. In our case, all probabilities of the distribution are used to regularize the classifier. While this also informs the shape of the distribution for the perspective of uncertainty modeling, the skewness reflecting the high or low agreement level learned at the current data sample is also revealed. Thereon, two fully-connected layers with RELU and Sigmoid activations respectively are used to process such information and produce the agreement indicator $\tilde{y}_i$ for regularization.

Learning Agreement with Linear Regression. Straightforwardly, we can also formulate the agreement learning as a plain linear regression task, modelled by a fully-connected layer with a Sigmoid activation function (see the lower part of Fig.A.3). Then, the predicted agreement $\hat{y}_i$ is directly taken as the agreement indicator $\tilde{y}_i$ for regularization. Given the predicted agreement $\hat{y}_i$ and target agreement α_i at each input sample x_i, by using Root Mean Squared Error (RMSE), the linear regression loss is computed as

$$\mathcal{L}_{RMSE}(\hat{y}, \alpha) = [\frac{1}{N}\sum_{i}^{N}(\hat{y}_i - \alpha_i)^2]^{\frac{1}{2}}. \tag{A.3}$$

It should be noted that, the proposed AR loss can also be used for this linear regression variant, which may help optimize the underlying distribution toward the given agreement level. In the experiments, an empirical comparison between different variants for agreement learning is conducted.

A.3.2 Regularizing the Classifier with Agreement Information

Since the high-level information implied by the agreement between annotators could provide extra hints in classification tasks, we utilize the agreement indicator $\tilde{y}_i$ to regularize the classifier training toward providing outcomes that better fitting with the annotators.

Given a binary classification task (a multi-class task can be decomposed into several binary ones), at input sample x_i, we denote the original predicted probability toward the positive class of the classifier to be $\hat{p}_\theta(x_i)$. The general idea is that, when the learned agreement indicator is i) at chance level *i.e.* $\tilde{y}_i = 0.5$, $\hat{p}_\theta(x_i)$ shall stay

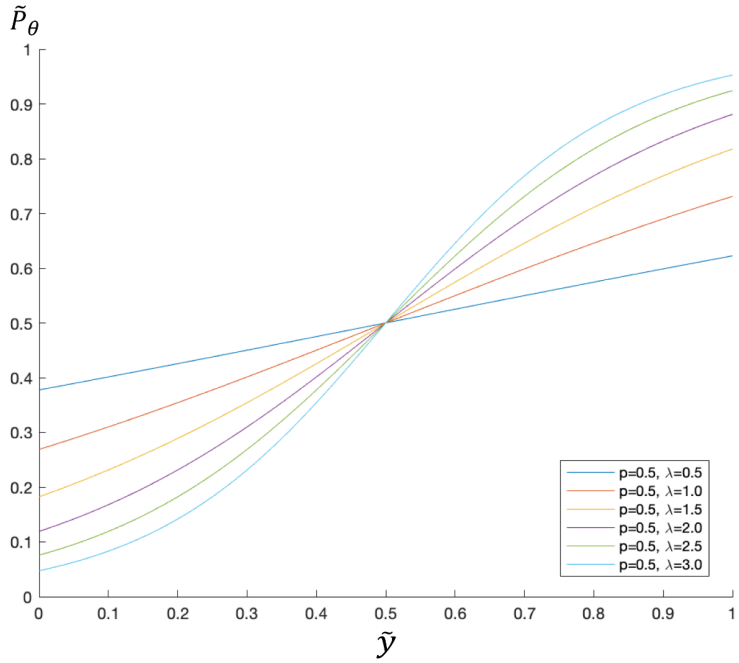


Figure A.4: The property of the regularization function. X and Y axes are the agreement indicator $\tilde{y}_i$ and regularized probability $\tilde{p}_\theta(x_i)$, respectively. $\tilde{p}_\theta(x_i)$ is regularized to the class, for which the $\tilde{y}_i$ is high, with the scale controlled by λ.

unchanged; ii) biased toward the positive or negative class, the value of $\hat{p}_\theta(x_i)$ shall be regularized to be higher or lower accordingly. For these targets, we propose a novel regularization function written as

$$\tilde{p}_\theta(x_i) = \frac{\hat{p}_\theta(x_i)e^{\lambda(\tilde{y}_i-0.5)}}{\hat{p}_\theta(x_i)e^{\lambda(\tilde{y}_i-0.5)} + (1-\hat{p}_\theta(x_i))e^{\lambda(0.5-\tilde{y}_i)}}, \tag{A.4}$$

where $\tilde{p}_\theta(x_i)$ is the regularized probability toward the positive class of the binary task, λ is a hyperparameter controlling the speed at which the raw predicted probability $\hat{p}_\theta(x_i)$ changes toward $\tilde{p}_\theta(x_i)$ when the agreement indicator increases or decreases.

Fig.A.4 shows the property of the function: for the original predicted probability $\hat{p}_\theta(x_i) = 0.5$, the function with larger λ augments the effect of the learned agreement indicator $\tilde{y}_i$ so that the output $\tilde{p}_\theta(x_i)$ is regularized toward the more (dis)agreed; when $\tilde{y}_i$ is at 0.5, where annotators are unable to reach an above-chance opinion about the task, the regularized probability stays unchanged with $\tilde{p}_\theta(x_i) = \hat{p}_\theta(x_i)$.

A.3.3 Alleviating Imbalances when Using Logarithmic Loss

In this subsection, by refining the traditional cross-entropy loss, we first alleviate the influence of class imbalances present in the annotation of each annotator on the classifier stream that learns from multiple annotators. We further look into the use of another loss function that directly designed for the objective of reaching better agreement levels with annotators. By using this loss function, we may also avoid the class-imbalance problem during training.

Annotation Balancing for Each Annotator. For the classifier stream, given the regularized probability $\tilde{p}_\theta(x_i)$ at the current input sample x_i, the classifier is updated using the sum of the loss computed according to the available annotation r_i^j from each annotator.

Due to the various the nature of the task (*i.e.*, positive samples are sparse), the annotation from each annotator could be noticeably imbalanced. To address this problem, we use the Focal Loss (FL) [129], written as

$$\mathcal{L}_{\mathrm{FL}}(p,g) = -|g-p|^{\gamma}(g\log(p)+(1-g)\log(1-p)), \tag{A.5}$$

where p is the predicted probability of the model toward the positive class at the current data sample, $g \in \{0,1\}$ is the binary ground truth, and $\gamma \geq 0$ is the focusing parameter used to control the threshold for judging the well-classified. A larger γ leads to a lower threshold so that more samples would be treated as the well-classified and down weighted. In our scenario, the FL function is integrated into the following loss function to compute the loss for each annotator

$$\mathcal{L}_{\theta,j}(\tilde{\mathbf{P}}_\theta, \mathbf{R}^j) = \frac{1}{\grave{N}^j}\sum_{i=1}^{\grave{N}^j}\mathcal{L}_{FL}(\tilde{p}_\theta(x_i), r_i^j), \tag{A.6}$$

where $\grave{N}^j \leq N$ is the number of samples that have been labelled by j-th annotator, $\tilde{\mathbf{P}}_\theta = \{\tilde{p}_\theta(x_i)\}_{i=1,\ldots,N}$, $\mathbf{R}^j = \{r_i^j\}_{i=1,\ldots,N}$ and $r_i^j = null$ if this annotator did not annotate at $i-$th sample and the loss is not computed here. By default, the losses

computed from all the annotators are averaged to be the final loss of the classifier

$$\bar{\mathcal{L}}_\theta(\tilde{\mathbf{P}}_\theta, \mathbf{R}) = \frac{1}{J}\sum_{j=1}^{J} \mathcal{L}_{\theta,j}(\tilde{\mathbf{P}}_\theta, \mathbf{R}^j). \tag{A.7}$$

Additionally, searching for the γ manually for each annotator could be cumbersome, especially for a dataset labeled by numerous annotators. In this work, in order to save such efforts, we compute γ for each annotator given the number of samples per class of each binary task. The hypothesis is that, for annotations biased more toward one class, γ shall set to be bigger since larger number of samples tend to be well-classified. Following [127], we leverage the effective number of samples to compute each γ_j as

$$\gamma_j = \frac{(1-\beta^{n_k^j})}{(1-\beta^{(\dot{N}^j - n_k^j)})}, \tag{A.8}$$

where n_k^j is the number of samples for the majority class k in the current binary task annotated by annotator j, $\beta = \frac{\dot{N}^j - 1}{\dot{N}^j}$.

Leaning with an Agreement-oriented Loss. In [182], a Weighted Kappa Loss (WKL) was used to compute the agreement-oriented loss between the output of a model and the annotation of an annotator. As developed from the weighted Cohen's Kappa [183], this loss may guide the model to pay attention to the overall agreement instead of the local accuracy. Thus, we may be able to avoid the cumbersome work of alleviating the class imbalances. The loss function can be written as

$$\mathcal{L}_{\text{WKL}} = \log(1-\kappa). \tag{A.9}$$

The linear weighted kappa is used as κ in this equation, where the penalization weight is proportional to the distance between the predicted and the class. We replace the FL loss written in Equation A.5, to compute the weighted kappa loss across samples and annotators using Equation A.6 and Equation A.7. Since the value range of Equation A.9 is $(-\infty, \log 2]$, a Sigmoid function is applied before we sum the loss from each annotator. We compare this loss function to the logarithmic one.

A.4 Experiment Setup

This section describes the datasets we use for evaluation, the implementation details, method for agreement computation, and the metric.

A.4.1 Datasets

Two medical datasets are selected to evaluate the proposed model, involving data of body movement sequences and bone X-rays.

EmoPain. As described in Chapter 3, four experts were recruited to provide the binary annotations of the presence or absence of protective behavior per timestep for each CP participant data sequence. In comparison with the studies presented in previous chapters that adopted majority-voting for ground truth definition, here we use the annotations of all the four annotators in our modeling.

MURA. The MURA dataset [167] comprises 40,561 radiographic images of 7 upper extremity types (*i.e.*, shoulder, humerus, elbow, forearm, wrist, hand, and finger), and is used for the binary classification of abnormality. This dataset is officially split into training (36,808 images), validation (3197 images), and testing (556 images) sets, with no overlap in subjects. The training and validation sets are publicly available, with each image labelled by a radiologist.

While some abnormalities like fractures and hardware implantation are deterministic, the others like degenerative diseases and miscellaneous abnormalities are mostly determined given further examination. Thus, at the prescreening stage, such abnormality classification relies on the expertise of the expert.

For the testing set, the authors of the dataset recognized possible disagreements from other experts during such prescreening process and recruited six additional radiologists for annotation, and defined the *ground truth* with majority-voting among each three randomly-picked radiologists for each sample. In average, the left-out three radiologists achieved Cohen's kappa with such *ground truth* of 0.731, 0.763, and 0.778, respectively.

To simulate the opinions of different experts for data we have access to, three virtual

annotators are purposely created to reach overall Cohen's kappa with the existing annotator of 0.80, 0.75, and 0.70, respectively. Here need to note that, such a process of creating the virtual annotators, by randomly changing the existing annotation, only cares about the overall Cohen's kappa of each simulated annotator with the existing real annotator, and we do not control the variance of annotations per sample.

A.4.2 Implementation Details

For experiments on the EmoPain dataset, the state-of-the-art HAR-PBD network presented in Chapter 6 is adopted as the backbone, and Leave-One-Subject-Out validation is conducted across the 18 participants with CP. The average of the performances achieved on all the folds is reported. The training data is augmented by adding Gaussian noise and cropping, as seen in Chapter 4, Chapter 5, and Chapter 6. The number of bins used in the general agreement distribution is set to 10, *i.e.*, the respective softmax layer for agreement learning has 11 nodes. The λ used in the regularization function is set to 3.0.

For experiments on the MURA dataset, the Dense-169 network [184] pretrained on the ImageNet dataset [185] is used as the backbone. The original validation set is used as the testing set, where the first view (image) from each of the 7 upper extremity types of a subject is used. Images are all resized to be 224×224, while images in the training set are further augmented with random lateral inversions and rotations of up to 30 degrees. The number of bins is set to 5, and the λ is set to 3.0.

For all the experiments, the classifier stream is implemented with a fully-connected layer using a Softmax activation with two output nodes for the binary classification task. Adam [102] is used as the optimizer with an initial learning rate set to lr =1e-4, which is reduced by multiplying 0.1 if the performance is not improving after 10 epochs. The number of maximum epochs is set to 50.

For the classifier stream, the logarithmic loss is adopted by default as used in Equation A.5, A.6, A.7, and A.9, while the WKL loss is used for comparison when mentioned. For the agreement learning stream, the AR loss is used for the distributional variant, while the RMSE is used for the linear regression variant. We implement our method with TensorFlow deep learning library on a PC with a RTX

3080 GPU and 32 GB memory.

A.4.3 Agreement Computation

For a binary task, the agreement level α_i between annotators is computed as

$$\alpha_i = \frac{1}{\grave{J}} \sum_{j=1}^{\grave{J}} w_i^j r_i^j, \tag{A.10}$$

where $\grave{J}$ is the number of annotators that have labelled the $i-$th sample x_i. In this way, $\alpha_i \in [0,1]$ stands for the agreement of annotators toward the positive class of the current binary task. In this work, we assume each sample was labelled by at least one annotator. w_i^j is the weight for the annotation provided by j-th annotator that could be used to show the different levels of expertise of the annotators. The weight can be set manually given prior knowledge about the annotator, or used as a learnable parameter for the model to estimate. In this work, we treat annotators equally by setting w_i^j to 1. We leave the discussion on other situations to future works.

A.4.4 Metric

Following [177], we evaluate the performance of a model by using the agreement ratio defined as

$$\Delta = \frac{C_J^2}{J} \frac{\sum_{j=1}^{J} \text{Sigmoid}(\kappa(\tilde{\mathbf{P}}_\theta, \mathbf{R}^j))}{\sum_{j,j'=1 \& j \neq j'}^{J} \text{Sigmoid}(\kappa(\mathbf{R}^j, \mathbf{R}^{j'}))}, \tag{A.11}$$

where the numerator computes the average agreement for the pairs of predictions of the model and annotations of each annotator, and the denominator computes the average agreement between annotators with C_J^2 denoting the number of different annotator pairs. κ is the Cohen's Kappa. The agreement ratio $\Delta > 0$ is larger than 1 when the model achieves better performance than the average annotator.

A.5 Results

In this section, we present and discuss the results in the evaluation of our proposed method on the EmoPain dataset that we use across this thesis and another medical dataset. We first demonstrate the improvements introduced by our method. Then, we

study the impact of the proposed AR loss.

A.5.1 Logarithmic Loss with Balancing Methods vs. WKL Loss

As shown in the first section of Table A.1, models trained with majority-voted ground truth produce agreement ratios of 1.0417 and 0.7616 with logarithmic loss and annotation balancing (in this case is class balancing for the single majority-voted ground truth) on the EmoPain and MURA datasets, respectively.

As shown in the second section of Table A.1, directly exposing the model to all the annotations is harmful, with performances lower than the models with majority-voting of 0.9733 and 0.7564 achieved with logarithmic loss used alone on the two datasets, respectively. By using the balancing method during training, the performance on the EmoPain dataset is improved to 1.0189 but is still lower than what can be achieved using majority-voted ground truth, while a better performance of 0.7665 on the MURA dataset is achieved. These results show the importance of balancing for the modeling with learn-from-all paradigm.

The performances of the model with majority-voted ground truth (1.0452/0.7638)

Table A.1: The ablation experiment on the EmoPain and MURA datasets. Majority-voting refers to the method using the majority-voted ground truth for training. CE and WKL refer to the logarithmic and weighted kappa loss functions used in the classifier stream, respectively. Linear and Distributional refer to the agreement learning stream with linear regression and general agreement distribution, respectively. The best performance in each model/annotator set is marked in bold for each dataset.

Model/Annotator	CE	WKL	Annotation Balance	Linear	Distributional	EmoPain $\Delta\uparrow$	MURA $\Delta\uparrow$
Majority-voting	√		√			1.0417	0.7616
		√				**1.0452**	**0.7638**
Learn-from-all	√					0.9733	0.7564
	√		√			1.0189	0.7665
		√				**1.0407**	**0.7715**
Agreement Learning (Ours)	√		√	√		1.0477	0.7727
	√		√		√	1.0508	0.7796
		√		√		1.0471	0.7768
		√			√	**1.0547**	**0.7801**
Annotator 1						0.9613	**1.0679**
Annotator 2						1.0231	0.9984
Annotator 3						**1.0447**	0.9743
Annotator 4						0.9732	0.9627

and learn-from-all paradigm (1.0407/0.7667) are further improved by using the WKL loss on the two datasets, respectively. This proves the advantage of using the WKL loss for improving the fitting with multiple annotators, which is designed to optimize a model at the global agreement level with each annotator rather than the local accuracy.

A.5.2 The Impact of Agreement Learning

For both datasets, as shown in the third section of Table A.1, with our proposed agreement learning method using general agreement distribution, the best overall performances of 1.0547 (with WKL loss) and 0.7796 (with logarithmic loss) are achieved on the two datasets, respectively.

For agreement learning, the combination of general agreement distribution and AR loss shows better performance than its variant using linear regression and RMSE on both datasets (1.0477 with logarithmic loss and 0.7768 with WKL loss). Such results could be due to the fact that the agreement indicator produced from the linear regression is directly the estimated agreement value, which could be largely affected by the errors made during agreement learning. In contrast, with general agreement distribution, the information passed to the classifier is first the shape and skewness of the distribution. Thus, it is more tolerant to the errors (if) made by the weighted sum that used for the actual regression in the agreement learning. This advantage can also be taken as a way to capture the uncertainty that may exist in the annotations.

A.5.3 Comparing with the Annotators

In the last section of Table A.1, the annotation of each annotator is used to compute the agreement ratio against the other annotators.

For the EmoPain dataset, the best method in majority-voting (1.0452) and learn-from-all (1.0407) paradigms show very competitive if not better performances than annotator 3 (1.0447) who has the best agreement level with all the other annotators. Thereon, the proposed agreement learning method is able to improve the performance to an even higher agreement ratio of 1.0547 against all the annotators. This performance suggests that, when adopted in real-life, the model is able to analyze the

protective behavior of people with CP, at a performance that is highly in agreement with the human experts. These results once again show that the HAR-PBD backbone, which was proposed in our last study presented in Chapter 6, is able to provide promising results even when the learning scenario becomes more challenging.

However, for the MURA dataset, the best performance so far achieved by the agreement learning model of 0.7801 is still lower than annotator 1. This suggests that, at the current task setting, the model may make around 22% errors more than the average human expert. One reason could be largely due to the challenge of the task. As shown in [167], where the same backbone only achieved a similar if not better performance than the other radiologists for only one (wrist) out of the seven upper extremity types. In this work, the testing set comprises all the extremity types, which makes the experiment even more challenging. In the future, one may explore using better backbones other than the vanilla Dense network to improve such performance.

A.5.4 The Impact of Agreement Regression Loss

The proposed AR loss can be used for both the distributional and linear agreement learning. However, as seen in Table A.2 and Table A.3, the performance of linear agreement learning is better with RMSE rather than with the AR loss. The design of the AR loss assumes the loss computed for a given quantile is in accord with its counterpart of agreement level. Thus, such results may be due to the gap between the quantile of the underlying distribution of the linear regression and the targeted agreement level. Therefore, the resulting estimated agreement indicator using AR loss passed to the classifier may not reflect the actual agreement level. Instead, for linear regression, a vanilla loss like RMSE promises that the regression value is fitting toward the actual agreement level.

By contrast, the proposed general agreement distribution directly adopts the range of agreement levels to be the distribution values, which helps to narrow the gap when AR loss is used. Therein, the agreement indicator is extracted from the shape and skewness of such distribution (probabilities of all distribution values), which could better reflect the agreement level when updated with AR loss. As shown,

Table A.2: The experiment on the EmoPain dataset for analyzing the impact of Agreement Regression (AR) loss on agreement learning. The best performance in each agreement learning type is marked in bold.

Loss for Classifier	Agreement Learning Type	Agreement Learning Loss	$\Delta\uparrow$
CE	Linear	RMSE	**1.0477**
		AR	0.9976
	Distributional	RMSE	1.0289
		AR	**1.0508**
WKL	Linear	RMSE	**1.0471**
		AR	1.035
	Distributional	RMSE	1.0454
		AR	**1.0547**

Table A.3: The experiment on the MURA dataset for analyzing the impact of Agreement Regression (AR) loss on agreement learning. The best performance in each agreement learning type is marked in bold.

Loss for Classifier	Agreement Learning Type	Agreement Learning Loss	$\Delta\uparrow$
CE	Linear	RMSE	**0.7727**
		AR	0.7698
	Distributional	RMSE	0.7729
		AR	**0.7796**
WKL	Linear	RMSE	**0.7768**
		AR	0.7674
	Distributional	RMSE	0.7684
		AR	**0.7801**

the combination of distributional agreement learning and AR loss achieves the best performance in each dataset.

A.6 Summary

In this appendix chapter, we targeted the scenario of learning with multiple annotators where the ground truth is ambiguous to define during training as well testing. Unlike previous studies dealing with multiple annotators that aimed to model the underlying ground truth [161, 163, 164, 165, 166, 162], for this targeted learning scenario, we shift the focus of modeling to better fitting with all the annotators.

Aside from the EmoPain dataset that we used across this thesis, another medical comprising bone X-rays that also falls into this scenario was also adopted for the evaluation. We showed that backbones developed with majority-voted *ground truth* or multiple annotations can be easily improved to achieve better agreement levels

with the annotators, by leveraging the underlying agreement information stored in the annotations. Our experiments also showed that, when the objective is to better fit with annotators, the agreement-oriented loss (*i.e.*, the weighted kappa loss) is better than logarithmic loss (*i.e.*, the cross-entropy loss) during training.

For agreement learning, our experiments demonstrate the advantage of learning with the proposed general agreement distribution and agreement regression loss, in comparison with other possible variants. Future works may extend this work to prove its efficiency in datasets having multiple classes, as only binary tasks were considered in this work. Additionally, the learning of annotator's expertise seen in [165, 166, 162] could be leveraged to weight the agreement computation and learning proposed in our method for cases where annotators are treated differently.

This work was done in a visit to Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS) in 2021, with a paper in preparation and its preprint available at [186].