UNIVERSITY OF GENOVA

DITEN - DEPARTMENT OF ELECTRICAL, ELECTRONICS AND TELECOMMUNICATION ENGINEERING AND NAVAL ARCHITECTURE
PAVIS - PATTERN ANALYSIS AND COMPUTER VISION, ISTITUTO ITALIANO DI TECNOLOGIA

PHD IN SCIENCE AND TECHNOLOGY FOR ELECTRONIC AND TELECOMMUNICATION ENGINEERING
CURRICULUM: COMPUTATIONAL VISION, RECOGNITION AND MACHINE LEARNING

Unsupervised Human Action Recognition using 3D Skeleton Poses

PhD Thesis submitted for the degree of *Doctor of Philosophy*
(XXXV cycle)

PhD Candidate: **Giancarlo Paoletti**

Alessio Del Bue	Supervisor
Cigdem Beyan, Jacopo Cavazza	Co-Supervisors
Maurizio Valle	Coordinator of the PhD course

DITEN

iit PAVIS
ISTITUTO ITALIANO DI TECNOLOGIA

March 2023

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified.

Giancarlo Paoletti
March 2023

Abstract

Action recognition, a sub-field of computer vision, has garnered increasing attention in recent years due to its potential applications in addressing a wide range of real-world problems. By analysing individuals' movements and actions, researchers can better understand their underlying motivations, thoughts, and emotions, which has numerous practical applications, including the development of more effective algorithms that can better understand and respond to human behaviour. Some useful applications of action recognition include security surveillance systems, human-robot and human-computer interaction, patient monitoring and assistive technologies, sign language recognition, consumer behaviour analysis, and sports analysis. Despite recent progress, the development of a fully automated human activity recognition system that can accurately classify activities remains challenging due to the complexity of visual data, such as varying camera viewpoints, occlusions, changes in scale and appearance, background clutter, and lighting changes. A skeleton-based approach offers privacy-preserving characteristics and allows the model to focus on the essential characteristics of the body and its movements rather than being influenced by extraneous factors. This can result in a more accurate understanding of human anatomy and movement. Supervised learning approaches are effective in annotating sequences with corresponding actions or activities. However, this process is time-consuming, requires specialised knowledge, and is prone to human error. The problem is further complicated by intra-class and inter-class similarities, making it difficult to distinguish between different actions. As a result, the reliance on annotated data for sequence annotation may compromise the scalability of big data systems. This motivates the need to explore unsupervised methods as an alternative. Unsupervised learning techniques effectively overcome the challenges faced by traditional supervised methods in this research field. These challenges include a lack of labelled data and the high variability of human actions. Despite this, unsupervised learning for HAR remains an emerging sub-field of research, leading to the exploration of new techniques such as clustering, dimensionality reduction, and deep learning. The main focus of this thesis is unsupervised action recognition using 3D skeleton poses as a specific typology of

data, aiming to introduce new algorithms that address the limitations of previous models and provide insight into the usefulness of unsupervised learning. This study presents a subspace clustering algorithm for the classification of trimmed sequences of actions using skeleton joints datasets, introducing new strategies for handling temporal data using covariance matrices. Additionally, a novel unsupervised method using a convolutional autoencoder to learn human action representations is proposed. This approach demonstrates the benefits of combining residual convolutions with spatio-temporal convolutions, resulting in more efficient and memory-effective architectures with the introduction of graph Laplacian regularisation to reconstruct skeleton-based action sequences better. This research also examined the effectiveness of unsupervised methods for human emotion recognition from full-body movement data. However, current unsupervised methods, while designed for high recognition accuracy, do not consider the resilience of the models to perturbed data, which is common in real-world scenarios. Based on these findings, a novel framework was developed, incorporating a transformer encoder-decoder with strong denoising capabilities and additional losses to improve robustness against such data perturbation and alteration.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

1-NN	1-Nearest Neighbour Clustering algorithm
ACC	Classification Accuracy
AE-L	Graph Laplacian-regularised Convolutional-Residual AutoEncoder
AR	Axis Removal
CLN	Clean data
COV	Covariance matrix
GN	Gaussian Noise
GRL	Gradient Reversal Layer
HAR	Human Action Recognition
HER	Human Emotion Recognition
JO	Joint Outlier
JR	Joint Removal
LEP	Linear Evaluation Protocol
LR	Limbs Removal
RM	Reversed Motion

ROT Rotation

SHR Shear

SKELTER SKELeton TransformER

SSVI Self-Supervised Viewpoint Invariance

SUB Subtract

U-HAR Unsupervised Human Action Recognition

U-HER Unsupervised Human Emotion Recognition

Chapter 1

Introduction

The study of action recognition, a sub-task of computer vision, has been a crucial area of research for many years. It is a significant research topic because it offers insight into human behaviour, personality, and psychological states. Analysing individuals' movements and actions researchers can better understand their underlying motivations, thoughts, and emotions by analysing individuals' movements and actions. This knowledge has numerous practical applications, including developing more effective algorithms that can better understand and respond to human behaviour. Overall, the study of action recognition is a critical component of the broader field of computer vision, with the potential to impact many different fields and industries. Recently, the field has garnered increasing attention due to its potential applications in addressing a wide range of real-world problems, including surveillance systems, human-robot and human-computer interaction [57], patient monitoring and assistive technologies [24], sign language [198, 37], computational behavioural science [174, 167, 43], consumer behaviour analysis [138], sports analysis, and many others. Below, some useful applications of action recognition for real-world cases are detailed.

Security surveillance is widely-used to protect individuals, structures, and possessions [132, 101, 69, 155, 91, 176]. A crucial aspect of modern surveillance systems is the ability to recognise actions accurately, reducing the need for human intervention. These systems have the potential to identify and prevent a range of undesirable events, altercations, criminal activities, and so forth. As the global population continues to age, the number of individuals aged 65 or over has reached 700 million in 2019 [199]. It is projected that by 2050, this demographic will make up 16% of the world's population [199]. This presents a growing concern for the care of older individuals.

One potential solution is the development of **assistive technologies**, methods used for elderly care: *i.e.,* robotic agents that can accurately recognise the actions of the elderly and respond to their behaviours accordingly. The market for virtual reality has experienced a significant increase: according to recent studies, the global virtual reality market size is projected to reach 84.09 billion USD by 2028 [72]. Action recognition techniques are crucial for implementing mature **virtual reality systems**, enabling computers to accurately interpret users' body movements and provide appropriate responses and interactions. The implementation and maintenance of social distancing have proven to be effective in controlling the spread of the recent COVID-19 viral outbreak. To ensure the effectiveness of this measure, governments have imposed restrictions on the minimum distance individuals must maintain between one another in public settings. The **visual social distancing** problem [42] is the automatic identification of interpersonal distances from an image and the categorisation of related groups of people. This is critical for conducting non-invasive analyses of individuals' adherence to social distancing guidelines and providing statistics on the safety levels of specific areas where these guidelines are not being followed. It is done by detecting and tracking two or more people (using *e.g.,* skeleton poses to maintain *privacy*), measuring their reciprocal distance and classifying whether they are too close to each other or not.

Human behaviours refer to physical actions associated with emotions, personality, and psychological state [133]. Therefore, to effectively recognise human activities through behaviour, it is necessary to determine the kinetic states of individuals. Some human activities, such as walking and running, are relatively easy to identify from both humans and action classifiers due to their prevalence in daily life for the former and the higher number of dataset samples (*w.r.t.* this particular action) for the latter. More complex and subtle activities are more challenging to recognise. The primary objective is to identify intentional and unintentional gestures that individuals use to communicate. This includes the voluntary selection of gestures to convey a message and the more subtle and often unconscious movements that may reveal underlying emotions or thoughts [27]. In these cases, it may be helpful to decompose the activity into simpler movements that are easier to identify (*e.g.,* segment a long and enriched action, composed of different gestures, into smaller chunks of atomic actions). In detail, gestures are considered to be primitive movements of the body that may correspond to a specific action [233]. Atomic actions are movements that describe a specific motion and may be part of more complex activities [140]. These components form human-to-object or human-to-human interactions, which usually involve multiple individuals or objects [157, 204]. Such interactions will ultimately form events,

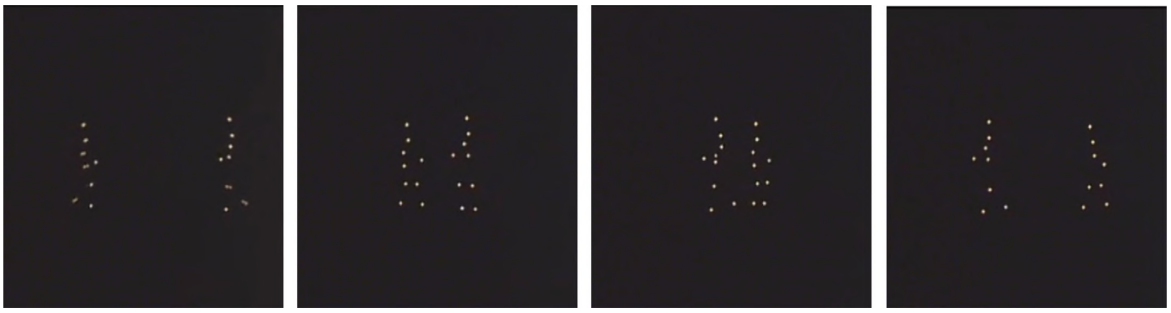


Figure 1.1 The moving lightspot experiment [90], conducted by Swedish perceptual psychologist Gunnar Johansson, aimed to document and explain the phenomenon of human sensitivity to biological motion. In the experiment, actors wore lightbulbs attached to their body parts and joints while performing various actions in the dark (on a black background)[1]. The results of the experiment showed that people were able to recognize the actions of the actors when the lightbulbs were moving (*e.g.,* two people walking towards each other), but not when they were stationary. This groundbreaking experiment inspired new fields of research into human perception, leading up to modern techniques that use multiple cameras to construct a 3D representation of actors' movements.

high-level activities that describe social interactions between individuals and indicate the intention or social role of a person [102].

These areas of study have spurred a significant portion of the computer vision community to research action recognition and modelling. Similar to other areas of computer vision, psychological studies often motivate current approaches. One notable example is Johansson's moving lightspots experiment [90], conducted in the 1970s to study 3D human motion perception from 2D patterns. This experiment, as displayed in Figure 1.1, demonstrated that the number of lightspots and their distribution on the human body could impact motion perception, with an increasing number of lightspots potentially reducing ambiguity in motion understanding. Johansson's study also showed that human vision could detect not only motion directions but also different types of limb motion patterns, including recognition of the activity and velocity of the motion patterns. As reported in the study [90]: "The geometric structures of body motion patterns in man are determined by the construction of their skeletons. From a mechanical point of view, the joints of the human body are endpoints of bones with constant length."

This study has influenced much of the literature on human body pose estimation, and action recognition [177, 33, 209], as the knowledge of the position of multiple body parts allows the machine to learn to distinguish between different action classes.

[1] The full video is available here: https://www.youtube.com/watch?v=1F5ICP9SYLU

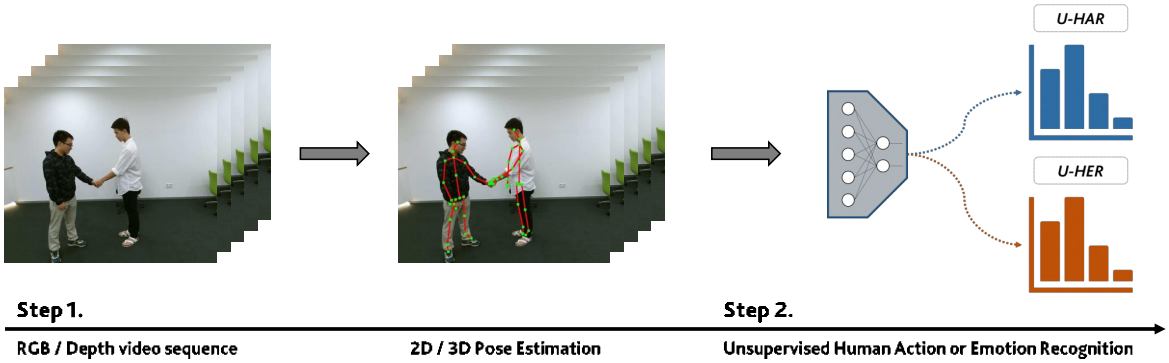


Figure 1.2 A general description of the HAR pipeline. It involves two main steps: skeleton data acquisition and action recognition using computational models. For the first step, input data is represented by RGB or Depth-based video frames (acquired from the respective sensors) which will be processed using an SDK toolkit or Human Pose Estimation architectures [19]. The second step is to devise a model capable of correctly classifying and discriminating the different actions involved for each data sample. The scope of this thesis lies within this last category.

In line with, and inspired by such insights mentioned above, this thesis's main research focus is directed towards recognising actions using a specific typology of data: *3D skeleton poses*. Despite the significant progress made in the last few years, the development of a fully automated human activity recognition system that can accurately classify activities remains a challenging task due to the complexity of the visual data, such as varying camera viewpoints, partial or total occlusions, changes in scale and appearance, background clutter, and abrupt changes in lighting conditions. Consequently, a skeleton-based HAR approach is an exciting paradigm to consider, given its beneficial privacy-preserving characteristics. This helps ensure that the model is not influenced by any potential biases or preconceived notions, achieving a more accurate understanding of human anatomy and movement using skeleton poses. As a result, the model is able to focus on the essential characteristics of the body and its movements rather than being distracted by extraneous factors.

1.1 Rationale

In skeleton-based HAR, action or activity sequences are represented through the multi-dimensional time series of joints located at the intersection of skeletal bones, which are typically tracked in time via motion capture systems, images or depth sensors (as seen in Figure 1.2). Recently, skeleton-based HAR has undergone a paradigm shift, similar to other

fields of pattern recognition, with the replacement of hand-crafted feature representations by data-driven ones and the adoption of an end-to-end classification pipeline.

1.1.1 Supervised learning and its shortcomings

The literature demonstrates the effectiveness of *supervised learning* approaches for both paradigms, where each sequence is manually annotated with the corresponding action/activity [28, 241, 227, 143, 137, 185, 11, 29]. But this comes with the cost of annotating behavioural roles is time-consuming and requires specific knowledge of the event. Each sequence is in fact assumed to be (manually) annotated by the action/activity it involves. Additionally, intra-class and inter-class similarities can make the problem even more challenging. Actions within the same class may be expressed differently by different people, and actions between different classes may be difficult to distinguish due to similar information. Other than being an expensive task, a time-consuming task, and prone to human errors [153], sequence annotations compromise the scalability of the big data regime. This motivates the need to research unsupervised (or self-supervised) methods which do not need to rely heavily on such annotated data, *fuelling the main core of this thesis*.

1.1.2 Unsupervised learning and contributions

Unlike supervised counterparts mentioned in Section 1.1.1, unsupervised learning methods for HAR manage to overcome issues as mentioned above (*e.g.,* lack of labelled data and the high variability of human actions) imposing as well fewer computational and methodological burdens *w.r.t.* supervised methods. Still, it represents an emerging sub-field of research, and this motivates researchers to explore new unsupervised skeleton action recognition techniques, such as clustering, dimensionality reduction, and deep learning. In other words, due to the increasing demand for methods capable of handling and modelling the ever-growing supply of unlabelled data, this thesis aims to introduce in literature new methods capable of mitigating shortcomings of previous models and shed light on the goodness of using such unsupervised algorithms. The contributions related to the PhD research, focused on Human Activity Recognition (HAR) using unsupervised learning techniques, can be described as follows.

Subspace Clustering

This section outlines a subspace clustering algorithm for classifying trimmed sequences of actions using skeleton joints datasets, introducing new strategies for handling temporal data using covariance matrices. Subspace clustering was initially devised in Computer Vision to segment dynamic moving objects [70, 38]. This technique posits that high-dimensional data (in the context of this study, skeletal joints) can be represented as a union of subspaces, each having a lower dimensionality and simpler geometric structure. Each subspace typically corresponds to a class (in this case, an action or activity).

The central concept in subspace clustering is learning encodings that are subsequently utilised to construct an affinity matrix from which the data can be clustered according to the modelled similarities and differences between samples [211]. While this is often achieved through a self-expressive model in which each data point is expressed as a linear combination of the remaining ones, additional constraints such as sparsity have also been adopted [60]. A limited number of studies have only explored subspace clustering to solve skeleton-based human action recognition (HAR) tasks [239, 113, 34]. This is due to several operational limitations, including difficulty handling the temporal dimension, the inherent noise in skeletal data, and the associated computational challenges.

Two alternative computational strategies to support subspace clustering methods in dealing with the temporal dimensions of action sequences were developed to address these issues. The first approach encodes raw skeletal trajectories using a covariance representation, which aids in solving HAR problems [22]. The second approach involves devising a computational strategy for pruning instantaneous body poses whose temporal aggregation produces an action sequence. As a result of temporal pruning, the most representative timestamps can be selected and used to compress the original action sequence to a fixed duration. Therefore, this temporal pruning can be employed as a successful pre-processing step for utilising a subspace clustering method for HAR.

Convolutional-Residual AutoEncoder for skeleton-based U-HAR

Subsequently, a novel method for handling the spatial correlation of human joints in larger and more complex datasets using a Graph Laplacian regularizer was proposed, which offers the advantage of being lighter than other methods in the literature. A recent paper submission also proposed a novel unsupervised method that uses a convolutional (residual) autoencoder to learn human action representations. This approach demonstrates the benefits of combining

residual convolutions with spatio-temporal convolutions rather than relying on more complex and memory-intensive architectures.

Graph Laplacian regularizer for U-HAR and self-supervised viewpoint-invariance

One key factor in the previous method is the adoption of graph Laplacian regularisation in the reconstruction space, *i.e.,* the reconstructed skeletal action. The graph Laplacian is a well-known tool for analysing weighted undirected graphs, and it was used to promote the alignment of skeletal joints that are connected by bones. Additionally, a method for promoting viewpoint invariance in camera position and orientation was designed, which is particularly useful in practical scenarios where data capturing of human actions is often different from the setups used in the tested dataset. First, the original skeletal data was perturbed with random rotations along the X, Y, and Z axes to improve viewpoint invariance in different camera positions and orientations. Then, a strategy to increase the model's generalizability was developed by pairing the Laplacian-regularised reconstruction loss with a regressor head that attempts to learn the parameters (*i.e.,* the rotation angles using Euler's angles) of the random rotations applied. Finally, adversarial training as a gradient reversal layer was used to learn a feature representation invariant to rotations, thus fooling the regressor. This method does not require annotated data features, as the randomly rotated skeletal actions are directly synthesised from the data itself, representing the core concept of self-supervision.

Human Emotion Recognition and Pose Denoising for HAR

As a final remark, the research was expanded by adapting the devised models to different scenarios, including Human Emotion Recognition (HER) and developing new methods capable of being noise-resistant in real-life perturbations. The proposed unsupervised methods were further evaluated for their effectiveness in emotion recognition from full-body movement data. Human Emotion Recognition is a complex task due to the varying contexts in which emotions are expressed and perceived and the interpersonal differences that impact emotional expression. Current HER datasets are typically smaller than HAR datasets due to the difficulty in collecting and annotating such data with high reliability. As both HAR and HER share similar data structures and commonly use supervised approaches, the previously-published unsupervised methods were applied to recognise emotions expressed through skeletal poses over time.

This exploration of unsupervised feature learning for HER from full-body movements represents a novel approach, applying the previously proposed convolutional residual autoencoder model to infer emotions from skeletal poses. The use of large-scale datasets is common in current unsupervised human activity recognition (U-HAR) and unsupervised human-environment interaction recognition (U-HER) methods. These methods are designed and optimised for high recognition accuracy. Still, they do not consider the resilience of the models to perturbed data, which is common in real-world testing scenarios. To address this issue, a systematical analysis was applied to check the performance decrease of state-of-the-art U-HAR algorithms when using perturbed or altered data, such as removing skeletal joints, rotating the pose, or injecting geometrical aberrations.

Based on the findings, a method called SKELTER was devised, a novel framework based on a transformer encoder-decoder with strong denoising capabilities to counter such perturbations. Additional losses were also introduced to improve the robustness of the model against rotation variations and provide temporal motion consistency. In the case of perturbed skeleton poses, the proposed model showed lower performance decreases in the presence of noise compared to previous approaches, making it a suitable solution for challenging in-the-wild settings.

1.2 Summary of Contributions

A summary of this thesis' contributions is the following:

- Developing a subspace clustering algorithm for fully-unsupervised human action classification
- Propose a novel unsupervised feature learning method that uses a convolutional (residual) autoencoder to learn human action representations.
- Adopting graph Laplacian regularization in reconstruction space improves the alignment of skeletal joints connected by bones.
- Design a method for promoting viewpoint invariance in camera position and orientation using a gradient reversal layer.
- Expanded research to adapt action recognition models for the human emotion recognition task from full-body skeletal movement data.
- Systematic analysis of the performance of state-of-the-art unsupervised action recognition algorithms when using perturbed or altered data.

1.3 Publications

The work presented in this thesis has produced the following publications:

- Giancarlo Paoletti, Jacopo Cavazza, Cigdem Beyan, and Alessio Del Bue. Subspace clustering for action recognition with covariance representations and temporal pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6035–6042. IEEE, 2021. *Oral presentation.*
- Giancarlo Paoletti, Jacopo Cavazza, Cigdem Beyan, and Alessio Del Bue. Unsupervised human action recognition with skeletal graph laplacian and self-supervised viewpoints invariance. In *32nd British Machine Vision Conference 2021*, BMVC 2021, Online, November 22-25, 2021. BMVA Press, 2021. *Oral presentation.*
- Giancarlo Paoletti, Cigdem Beyan, and Alessio Del Bue. Graph Laplacian-Improved Convolutional Residual Autoencoder for Unsupervised Human Action and Emotion Recognition. *IEEE Access.*
- Giancarlo Paoletti, Cigdem Beyan, and Alessio Del Bue. SKELTER: Unsupervised Skeleton Action Denoising and Recognition using Transformers. *Submitted and under review.*

1.4 Thesis Organization

This thesis is organised as follows. Chapter 2 outlines a general overview of skeleton-based models for Unsupervised Human Action Recognition, with a thorough presentation of 3D skeleton-action datasets utilised in overall works, Chapter 3 presents subspace clustering methods deployed for action classification, Chapter 4 includes a proposed encoder-decoder model to learn spatio-temporal skeletal features with extensive ablation studies and introducing the emotion classification from body poses, Chapter 5 illustrates a proposed model to denoise corrupted skeletal poses, and Chapter 6 conclusions are drawn *w.r.t.* overall thesis.

Chapter 2

Background & Datasets

In this chapter, the overall definitions of action and emotion recognition are described, with the related works in literature for this task and an extensive description of all 3D skeleton-based action datasets used for the entirety of this thesis.

2.1 Human Action Recognition

Human Action Recognition (HAR) task can be defined as classifying which action is displayed in a trimmed sequence. This task plays a crucial role in computer vision since it is related to a broad spectrum of artificial intelligence applications (such as video surveillance, human-machine interaction or self-driving cars, and so forth [166, 223, 168]). Given a trimmed sequence in which a single action or activity is assumed to be present, the final goal of HAR is to classify it correctly. Although significant progress has been made in recent years, accurate action recognition in videos is still a challenging task because of the complexity of the visual data, *e.g.,* due to varying camera viewpoints, occlusions and abrupt changes in lighting conditions.

To perform HAR, several modalities have been exploited, such as video frames (RGB) [20, 14, 53, 12, 96, 103, 120, 15, 119], video frames with depth information (RGB+D) [68, 115, 232, 148, 171, 170, 172, 131], and skeleton data [225, 76]. One advantage of using depth videos over conventional RGB videos is their ease of foreground human subject segmentation (even in cluttered scenes), allowing researchers to focus more on robust feature descriptors for action recognition rather than low-level segmentation. However, depth

images are also susceptible to noise and do not always guarantee good action recognition performance.

As an all-in-one solution to these problems, skeleton-based HAR is surely the paradigm to embrace, also considering its beneficial characteristics of being privacy-preserving since not a single RGB image needs to be stored, and it is a representation is easily given by off-the-shelf body pose detectors and potentially allowing performing HAR in real-time [19, 224, 188, 226, 227]. The onset of many recent skeleton-based action datasets had been possible thanks to the introduction, to the general market, of sensors like the Microsoft Kinect camera. Its ability to capture real-time RGB and depth videos, as well as the availability of a publicly available toolkit for computing human skeleton models from depth videos, has spawned a multitude of research papers on 3D HAR using the Kinect camera [187, 209, 181, 210, 94, 95, 169, 98, 240, 105, 212, 196, 123, 86, 104, 189, 213, 121, 6, 247, 85, 222, 56, 208, 248, 122, 1, 61, 188, 110, 183, 184].

In skeleton-based HAR, action/activity sequences are represented through the multidimensional time series of joints located at the intersection of skeletal bones, whose position is tracked in time, typically through motion capture systems or depth sensors. Existing methods for skeleton-based action recognition can be divided into two categories: joint-based and body part-based. Joint-based methods model the positions and movements of joints using coordinates already extracted. These coordinates can be defined *w.r.t.* a reference joint [232, 209, 94, 95], or the joint orientations can be computed relative to a fixed coordinate system[222]. On the other hand, body part-based methods model the human body as a system of rigid cylinders connected by joints. These methods often use information such as joint angles [210], the temporal evolution of body parts [181, 98, 227, 1], and 3D relative geometric relationships between rigid body parts [209, 210, 227] to represent the human pose for action recognition.

Recently, skeleton-based HAR has undergone the same paradigm shift, which was registered in other fields of pattern recognition: hand-crafted data encodings fed into engineered classifiers [141, 214, 209, 210, 147, 231, 148, 62, 209] have been replaced by data-driven feature representation with an end-to-end classification pipeline [99].

As for deep neural networks, recent studies are based on Recurrent Neural Networks (RNNs) [56, 181, 191, 241, 242, 114, 213, 247, 186, 56, 208, 248, 122], Convolutional Neural Networks (CNNs) [95, 107, 55, 116, 94, 197] and Graph Convolutional Networks (GCNs) [110, 183, 184, 188, 217, 227, 243, 245, 226] demonstrating the benefits of learning intrinsic

properties of skeletal actions performed over time. The current mainstream paradigm in skeleton-based HAR is the possibility of learning a feature representation from the data in tandem with the final action classifier. As one of the seminal works in this direction, a hierarchy of bidirectional recurrent neural networks is used by [56] to represent in a bottom-up fashion all the structural relationships between body parts (torso, legs, arms) in the human skeleton. Long Short-term Memory (LSTM) networks have been widely used in HAR due to their ability to model temporal dependencies and capture the co-occurrences of human joints. This ability, unique to LSTM networks among Recurrent Neural Networks (RNNs), has been demonstrated in several studies [181, 122, 79, 123, 104, 6, 248, 188]. The use of LSTM networks in this area has proven effective and contributed to their popularity as a choice for modelling human actions. Throughout the years, LSTM networks have been modified to accommodate the task better. For instance, by applying a novel mixed-norm regularization term and dropout [248] or recurring to attention mechanisms [123]. Alternatively, joint trajectories are cast into coloured images by producing the so-called distance maps [218, 106, 95]. Using the well-known convolutional neural networks such as AlexNet, despite originally proposed for image classification, can be adapted to HAR [218, 106]. Surely, the most active and recent research direction leverages the possibility of encoding the whole human skeleton as a graph, processing it through a graph-convolutional neural network [184, 221].

2.2 Unsupervised Human Action Recognition

The literature presented in the previous section leverages a fully supervised learning approach to accomplish the task [28, 241, 227, 143, 137, 185, 11, 29]. Each sequence is assumed to be (manually) annotated by the action/activity involved. Besides being an expensive and time-consuming task, prone to human errors [153], sequence annotations compromise the scalability of the big data regime.

As a (recent) alternative, *unsupervised* approaches [247, 194, 173, 80, 142, 225, 100, 117, 154, 134, 108, 238, 6, 73] are continuously reducing the performance gap with the fully supervised counterpart while dismissing the strong reliance over annotated data. Encoder-decoder recurrent architectures are often used to solve HAR problems [100, 247, 117, 194, 173]. Zheng *et al.* [247] introduce *LongT GAN*, based on GRUs that learns how to represent skeletal body poses in time. At the same time, an adversarial loss supports an auxiliary inpainting task favourably helps the learning stage. MS^2L [117] is also based on GRUs and

benefits from contrastive learning, motion prediction, and jigsaw puzzle recognition. In addition, Kundu *et al.* [100] include a GAN-based encoder in their recurrent architecture (*EnGAN*). *PCRP* [225] builds upon a vanilla autoencoder trained to reconstruct the skeletal data using mean-squared error (MSE) loss. This vanilla model is boosted by an ad-hoc training mechanism based on expectation maximization with learnable class prototypes. Su *et al.* [194] present the Predict & Cluster (*P&C*) method based on encoder-decoder RNN that learns representations for HAR in an unsupervised manner from skeletal joints while solving action classification with a 1-nearest neighbour predictor. *AS-CAL* [173] combines contrastive learning with momentum LSTM, where the similarity between augmented instances and the input skeleton sequence is contrasted. Then a momentum-based LSTM encodes the long-term actions. *SeBiReNet* [142] uses a Siamese denoising autoencoder is used with feature disentanglement, showing good performance across pose denoising and unsupervised cross-view HAR. Recently, Li *et al.* [109] processed the joint, motion, and bone information instead of using the datasets' skeleton data. *ISC* [200] leverages inter-skeleton contrastive learning and spatio-temporal augmentations to learn invariances *w.r.t.* skeleton representations. *AimCLR* [74] builds upon contrastive methods as well, and it is capable of obtaining robust representation from extreme augmentations and novel movement patterns.

2.3 Human Emotion Recognition From Full-Body Movements

Psychological research suggests that affective states are often communicated through body movements [135, 163, 47, 48, 2]. Affective phenomena, such as emotions, feelings, moods, attitudes, temperament, and interpersonal stances, can be categorised based on various factors, including the focus of the event, the appraisal of the event, the synchronisation of bodily responses, the speed of change, the behavioural impact, the intensity, and the duration [18, 149, 179, 178, 51, 40]. Scherer [178] defines emotions as "a synchronised change in the states of the cognitive, physiological, motivational, subjective feeling, and motor expression subsystems in response to the evaluation of a relevant stimulus event". The communication of emotions can be spontaneous or strategic [164, 17], with the former being involuntary and non-propositional, and the latter being goal-oriented and propositional. Basic emotions, such as anger, happiness, sadness, surprise, disgust, and fear, are defined by a specific set of neural and bodily responses, as well as a motivational component [59, 58, 202]. Theories of emotional expression often focus on facial expressions, but there is

also evidence to suggest that bodily expressions can be important indicators of affective states [45, 136, 203, 165]. During daily human-human interactions, people often pay attention to facial and bodily expressions of emotion. A recent study showed that bodily cues could be useful in discriminating between intense positive and negative emotions [163, 3]. Affective states can be expressed through various body movements, including whole-body gestures, arm gestures, and the modulation of functional movements.

Emotion recognition from full-body movement data is a complex task since the act of expressing and perceiving affect differs a lot *w.r.t.* its context, and also their variety increases due to the interpersonal differences (*e.g.,* personality, physical capacity, and personal experience) [93, 146]. This task can be inscribed within affective computing, a combination of artificial intelligence, computer vision, pattern recognition, cognitive science and psychology. It is a field of study that focuses on developing computing systems capable of modelling human affective states such as emotions, moods, and other related psychological phenomena. One of the key challenges is to accurately recognize and interpret various forms of affective expressions, such as hand gestures, facial expressions, physiological changes, and speech patterns. By doing so, these systems can help individuals better express, recognize, and control their affective states and enable machines and other computational systems to respond to and interact with humans in more natural and intuitive ways [151]. This research area is prominently based on the concept of *emotional intelligence*. According to Picard [162], the fundamental aspect of emotional intelligence is comprehending the connection between an individual's emotional states and the corresponding behaviours. These behaviours are closely linked to the emotional state and communication of the person with others. Therefore, emotional intelligence involves the ability to recognize and understand the emotion of the self and others, as well as the ability to communicate and manage those emotions in social settings effectively. Emotion recognition from full-body representation has been so far addressed by: *i)* processing single body pose (*e.g.,* a forward head and chest bend express sadness in [39]), *ii)* recognizing specific gestures which are emblems of the emotions (*e.g.,* raising arms and hands-on-hips are the gestures of pride according to [203, 144]), or *iii)* processing the expressive quality of the movement [44, 145, 65, 8]. Out of these three possibilities, the second and the third use the temporal information of the data, while the first one performs only spatial processing.

The existing related datasets were curated with diverse motion capture (MoCap) systems and various numbers of markers. These datasets are smaller than the HAR counterparts due to the effort needed to collect and, most importantly, *annotate* such data with high reliability. As

annotations are more costly, it is crucial to develop unsupervised feature learning methods that can effectively apply to HER.

Earlier works define hand-crafted features and apply learning methods such as Support Vector Machines (SVM) and Random Forests [21, 66, 32, 161]. For instance, Castellano *et al.* [21] use motion quantity, velocity, and movement fluidity as the descriptors of movements and aggregate them in the temporal dimension to classify four-emotion classes. Instead, Piana *et al.* [161] extend the low-level features by adding high-level features (*e.g.,* contraction index, impulsiveness) and applying an SVM classifier. On the other hand, Fourati *et al.* [66] show the importance of using temporal features (*e.g.,* regularity of a motion profile, overall or single gesture phase impulsiveness) and multi-level body cues (*e.g.,* based on Body Action and Posture Coding System) for emotions elicited during the daily-life actions. In [46], the 3D-skeleton data is represented in the Riemannian manifold and then processed with a covariance operator. This methodology was adapted by Kacem *et al.* [92], where the former applies a Nearest Neighbour classifier and uses a temporal warping and SVM. Both methods improved the emotion recognition from 3D-body movements results *w.r.t.* the prior art. As a different approach, Creen *et al.* [41] synthesize neutral motion by quantizing it with a cost function and then calculate the difference between the neutral class and the other emotions to decide the class 3D-body expression at inference.

Deep learning architectures *e.g.,* Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), have been explored for skeleton-based HER in recent studies. For example, [127] used an RNN with 3-layers to perform emotion classification from MoCap data of daily activities: clapping, drinking, throwing, and waving, etc., associated with four emotions: happy, angry, sad, and neutral. Beyan *et al.* [8] present the joint training of two CNNs such that one of them performs coarse-grained modelling while the other applies fine-grained modelling in the time. The inputs of this network are 8-bit RGB images obtained from 3D-skeleton data over time. This approach [8] achieves better performance compared to [65, 66, 41], showing generalisation properties over the diverse number of emotion classes and contexts.

2.4 3D action recognition datasets

The following action datasets were used for the experimental analyses scattered across different chapters of this thesis.

2.4.1 Florence3D

Florence3D (F3D) [180] is a 9-class action dataset (*answer phone, bow, clap, drink, read watch, sit down, stand up, tight lace, wave*) captured using a Microsoft Kinect camera. The actions were performed two/three times by 10 subjects, resulting in 215 data samples.

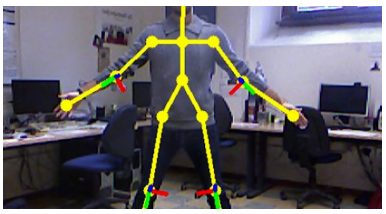


Figure 2.1 A sample from Florence 3D [180]

2.4.2 UTKinect-Action3D

UTKinect-Action3D (UTK) [222] is a 10-class action dataset (*carry, clap hands, pick up, pull, push, sit down, stand up, throw, walk, wave hands*) captured using a single stationary Microsoft Kinect camera. Each action was performed two times by 10 subjects, resulting in 199 data samples. Each estimated skeleton has 20 joints.

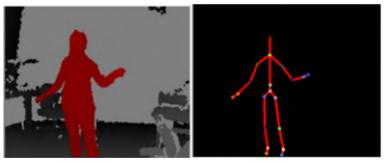


Figure 2.2 UTKinect [222] dataset sample

2.4.3 MSR 3D Action Pairs

MSR 3D Action Pairs (MSRP) [148] includes 12 actions in pairs (*pick up a box, put down a box, lift the box, place the box, push a chair, pull a chair, wear a hat, take off the hat, put on the backpack, take off the backpack, stick poster, remove poster*). Each pair has similar features, but their relationship in terms of motion and shape is different. The actions were performed three times by 10 subjects, resulting in 353 activity samples.

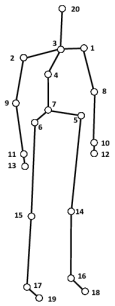


Figure 2.3 MSR 3D Action Pairs [148] skeleton pose

2.4.4 MSR Action 3D

MSR Action 3D (MSRA) [115] is a 20-class action dataset (*bend, draw a circle, draw tick, draw x, forward kick, forward punch, golf swing, hand catch, hand clap, hammer, high arm wave, high throw, horizontal arm wave, jogging, pick up and throw, side boxing, side kick, tennis serve, tennis swing, two-handwave*) captured by a depth-camera. Each action was performed three times by 10 subjects, resulting in 557 data samples. The skeleton in each sequence's frame comprises 20 joints.

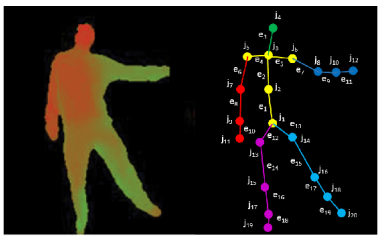


Figure 2.4 A sample from MSR Action 3D [115]

2.4.5 Gaming 3D

Gaming 3D (G3D) [13] is a 20-class gaming actions dataset (*aim and fire gun, clap, climb, crouch, defend, flap, golf swing, jump, kick left, kick right, punch left, punch right, run, steer a car, tennis swing backhand, tennis swing forehand, tennis serve, throw a bowling ball, wave, walk*) captured using a Kinect camera. The actions were repeated seven times by 10 subjects, resulting in 663 activity samples.

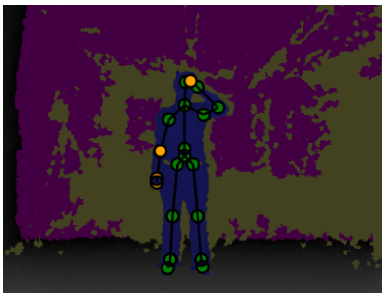


Figure 2.5 A sample from Gaming 3D [13]

2.4.6 HDM05

HDM05 [139], due to class imbalance of the original dataset, for the experimental analysis 14 classes (**HDM-05-14**, *clap above head, deposit floor, elbow to knee, grab high, hop both legs, jog, kick forward, lie down on the floor, rotate both arms backwards, sit down chair, sneak, squat, stand up, throw basketball*, following the protocol of [216, 23]), and 65 classes (**HDM-05-65** were used, following the protocol of [31] by grouping together similar actions). The sequences were captured using VICON cameras at 120Hz, resulting in 686 data samples for the former and 2343 data samples for the latter.

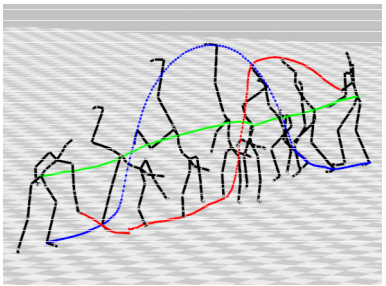


Figure 2.6 An action sequence from HDM05 [139]

2.4.7 MSRC-Kinect12

MSRC-Kinect12 (MSRC) [64] is a 12-class gesturing dataset, grouped into iconic and metaphoric gestures (*beat both, bow, change weapon, duck, goggles, had enough, kick, lift outstretched arms, push right, shoot, throw, wind it up*). Following the protocol as in [87], highly corrupted actions were removed, resulting in 5881 data samples.

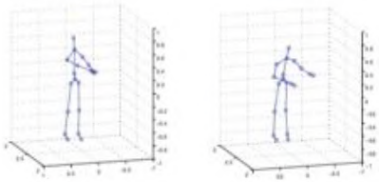


Figure 2.7 Samples from MSRC-Kinect12 [64]

2.4.8 NTU-60

NTU-60 [181] contains 60 action classes performed by 40 subjects, captured with Microsoft Kinect v2 at 30fps (frames-per-second). The videos were collected in a laboratory using Microsoft Kinect V2 cameras, resulting in accurately extracted skeletons with 25 joints, each incorporating more than 56,880 videos and 4 million frames. The dataset covers a range of scenarios, including daily individual and interactive behaviours and medical conditions. These actions were performed by 40 subjects aged between 10 and 35 and were recorded by three cameras positioned at different angles. While the high-quality skeletons in the NTU60 dataset provide a valuable resource for action recognition, there are several challenges associated with this task. These challenges include the variability of skeleton sizes and action speeds among subjects, the different viewpoints from which the skeletons are captured, and the similarity of motion trajectories among different actions. Additionally, the limited number of joints used to depict hand actions can make it difficult to portray them in detail. Three cameras record action sequences, facing frontally *w.r.t.* the subject and diagonally facing the subject with $\pm 45°$ angle. The authors of the NTU60 dataset recommend evaluating the accuracy of action recognition models under two settings. The first setting, referred to as Cross-Subject (C-Subject), involves splitting the 40 subjects evenly into training and validation groups, resulting in 40,320 sequences for training and 16,560 for validation. The second setting, called Cross-View (C-View), involves using sequences captured from the cameras that directly face and are oriented at $+45°$ toward the subject for training (37,920 instances), and the remaining sequences from the $-45°$ orientation view for validation (18,960 instances).

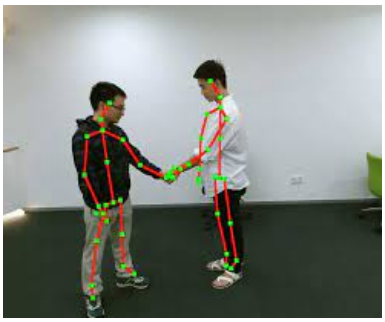


Figure 2.8 A sample action from NTU-60 [181]

2.4.9 NTU-120

NTU-120 [121] encompasses 113,945 samples and 120 classes. These actions were performed by 106 unique subjects with 32 different camera setups, *e.g.,* different backgrounds or locations where the data is captured. It is an extension of NTU60, with 57,367 additional skeleton sequences and 60 extra action categories. The authors suggest substituting the original Cross-View evaluation protocol with the Cross-Setup (C-Setup) protocol, which uses more camera positions and angles. Specifically, 54,468 skeleton sequences from half of the camera setups are used for training, and the remaining 59,477 samples are for validation. For the Cross-Subject (C-Subject) setting, 63,026 skeleton sequences collected from 53 subjects are utilised for training, and the remaining 50,919 samples are for validation.

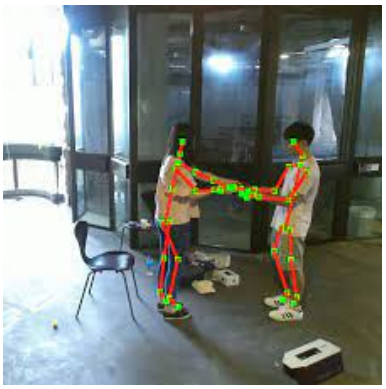


Figure 2.9 A sample action from NTU-120 [121]

2.4.10 Skeletics-152 Action Recognition In-the-wild Dataset

Skeletics-152 [75] was made from the Kinetics-700 dataset [190] by discarding some of the Kinetics-700 dataset's data due to being unfeasible or irrelevant to skeleton-based HAR. For example, videos containing occluded poses, egocentric videos, and videos composed of object interactions were omitted by [75]. Afterwards, VIBE [97] algorithm and some post-processing steps were applied, resulting in 125621 3D-skeleton sequences corresponding to 152 action classes.

Figure 2.10 Samples from Skeletics-152 [75]

2.5 3D emotion recognition datasets

The following emotion datasets were used for the experimental analyses of Chapter 4.

2.5.1 Dance Motion Capture Emotion Database

Dance Motion Capture Emotion Database (DMCD) [52] consists of various dance performances recorded with the PhaseSpace Impulse X2 MoCap system. The contemporary dance sequences were performed by six participants having different dance-related backgrounds. Each choreography the artists perform is associated with one of 12 emotions: excited, happy, pleased, satisfied, relaxed, tired, bored, sad, miserable, annoyed, angry, and afraid. In total, 108 performances correspond to 614898 3D points captured with 38 markers.

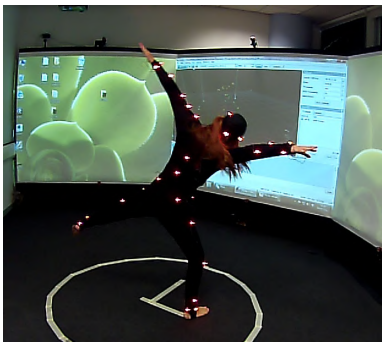


Figure 2.11 A performance from Dance Motion Capture Emotion Database [52]

2.5.2 Emilya Emotional Body Expressions Dataset

Emilya is a 3D-MoCap dataset [65] of emotional body expressions during eight daily actions: simple walking, walking with an object in hands, moving books on a table, knocking, sitting down, being seated, lifting, and throwing. The dataset was collected with 28 markers from 12 people who performed the earlier actions associated with eight emotional states: anxiety, pride, joy, sadness, panic, fear, shame, anger, and neutral. Prior papers have applied two types of cross-validation on the Emilya dataset.

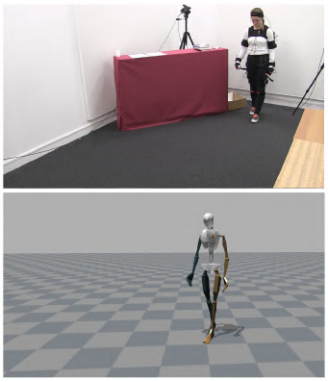


Figure 2.12 Emilya [65] dataset sample

Chapter 3

Subspace Clustering for Action Recognition with Covariance Representations and Temporal Pruning

Subspace clustering, a popular computational framework in the machine learning and computer vision and image processing communities, aims to find subspaces, each fitting a group of data points, and then performs clustering based on these subspaces [211]. It postulates that high-dimensional data (herein; *skeletal joints*) can be represented as a union of subspaces, each of them having a much lower dimensionality (*i.e.,* low-rank) and simpler geometrical structure. Each subspace usually corresponds to a class (*e.g.,* to an action or an activity). The key idea in subspace clustering is to learn encodings that are then used to construct an affinity matrix $\mathbf{W}$ from which the data can be clustered together according to the modelled (dis)-similarities between samples [211]. Although, this is usually achieved through a self-expressive model (Section 3.2.1) or dictionary-based model (Section 3.2.2) in which each data point is expressed as a linear combination of the remaining ones, additional constraints, such as sparsity, were also adopted [60].

Figure 3.1 illustrates the need for subspace clustering for high dimensional data. In such a scenario, many data points of the dataset could be nearly equidistant from each other. This could lead to an impairment of cluster quality of traditional clustering algorithms: by examining the entire dataset, many clusters could be masked and considered irrelevant, redundant, cut, misinterpreted or hidden within noisy data [156].

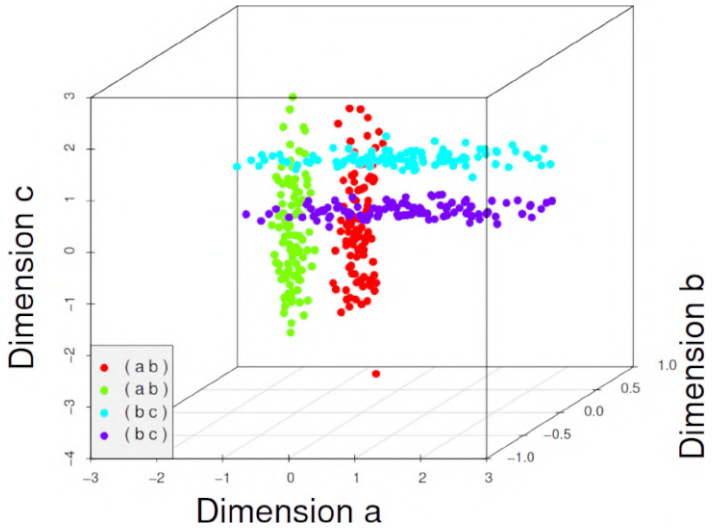


Figure 3.1 A simple three-dimension dataset to illustrate the need for subspace clustering, where points from two clusters can be very close together and confusing many traditional clustering algorithms [156]. It is divided into four clusters of 100 samples each, existing in only two of the three dimensions (the third one represents noise). Red and Green clusters exist in dimensions a and b, whereas Cyan and Purple clusters exist in dimensions b and c.

By implementing subspace clustering for Human Action Recognition (HAR), a key factor must be considered: the consistent variability of the length of the performed actions. Due to the complexity and nature of the action performed, each sample inherits a distinct temporal length in terms of timeframes. Therefore a regularisation must be applied to accommodate such data into subspace clustering algorithms (Section 3.3.2). Therefore, this chapter proposes a novel subspace clustering method, which exploits the covariance matrix to enhance the action's discriminability and a timestamp pruning approach that allows us to handle the temporal dimension of the data better, embracing the *fully unsupervised* paradigm (U-HAR).

3.1 Background and related work

Subspace clustering was first introduced in the computer vision domain to segment dynamic moving objects [38, 70] and implemented to solve other tasks *e.g.,* image representation and compression [81], image segmentation [228], and motion segmentation [63]. Most subspace clustering methods learn an affinity matrix $\mathbf{W}$ and then apply spectral clustering, *e.g.,* low-rank representation [118, 128]. Self-representation-based subspace clustering methods reconstruct a sample from a linear combination of other pieces [60, 118, 82, 129], and they have proven their effectiveness for high-dimensional data. Sparse subspace clustering integrates l1-norm regularisation, which mainly results in improvements in the clustering performances [60]. The temporal Laplacian regularisation was proposed in [113] and also adopted in other works *e.g.,* [34] to better model kinematic data for the sake of action detection and segmentation. As earlier subspace clustering methods rely on handcrafted representations, more recent and powerful representations can be learned through deep learning, which effectively cluster data samples from non-linear subspaces [89]. Deep subspace clustering methods apply embedding and clustering jointly, typically with an autoencoder network *e.g.,* in [89, 230]. This results in an optimal embedding subspace for clustering, which is more effective than conventional clustering methods. On the other hand, deep adversarial subspace clustering methods learn more effective sample representations using deep learning while exploiting adversarial learning to supervise and, thus, progressively improve the performance of subspace clustering. This is done using a subspace clustering generator and a quality-verifying discriminator, which are adversarially learned against each other.

Even though subspace clustering has become a powerful technique for problems such as face clustering or digit recognition, its applicability to the problems like skeleton-based HAR was only explored by a limited number of works [239, 113, 34]. This is due to many operative limitations, including handling the temporal dimensions, the inherent noise present in the skeletal data and the related computational issues.

3.2 Subspace clustering for HAR

To obtain the previously-mentioned affinity matrix $\mathbf{W}$, used to infer the predicted action labels for HAR, subspace clustering methods algorithms can be generally grouped into two main categories: *self-expressiveness based* and *dictionary-based* models.

3.2.1 *Self-Expressiveness based* models

Let us consider a collection of D-dimensional data-points $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Subspace clustering [211] attempts to cluster $\mathbf{x}_1, \ldots, \mathbf{x}_N$ into groups (termed *subspaces*) which share common geometrical relationships as the well-known *self-expressiveness property*.

The problem can be formalised as finding a $N \times N$ matrix $\mathbf{C}$ of coefficients such that

$$\mathbf{X} = \mathbf{XC} \text{ subject to } \text{diag}(\mathbf{C}) = 0, \tag{3.1}$$

where $\mathbf{X}$ is the $D \times N$ matrix, which stacks by columns the data points $\mathbf{x}_j$. The constraint $\text{diag}(\mathbf{C}) = 0$ avoids the trivial solution corresponding to $\mathbf{C}$ being the identity matrix. Ultimately, the geometrical relationship of relevance in modelling is a linear relationship in which each data point can be described as a linear combination. As a consequence of that, the subspaces are linear in turn. The constraint $\text{diag}(\mathbf{C}) = 0$ is fundamental to avoid the trivial (and useless) solution $\mathbf{x}_j = \mathbf{x}_j$. Specifically, the self-expressiveness property Equation 3.1 attempts to estimate each data point as a linear combination of *different data points*. This allows capturing the geometrical inter-dependencies among the data points themselves.

An important aspect regarding subspace clustering is how the matrix $\mathbf{C}$ is obtained. Several works proposed to solve this problem through optimisation [129, 60, 88, 237, 236, 89] and different strategies have been adopted to constraint the solution. In subspace segmentation via Least Squares Regression **(SS-LSR)** [129], a Frobenius norm is introduced to promote a L^2 penalty, obtaining

$$\min \|\mathbf{C}\|_F \text{ subject to } \mathbf{X} = \mathbf{XC}, \text{diag}(\mathbf{C}) = 0. \tag{3.2}$$

Another popular manner of constraining the coefficient matrix $\mathbf{C}$ is to impose sparsity [60, 236, 89]. As in the Sparse Subspace clustering via Alternating Direction Method of Multipliers **(SSC-ADMM)** [60], the problem formulation is framed as

$$\min \|\mathbf{C}\|_1 \text{ subject to } \mathbf{X} = \mathbf{XC}, \text{diag}(\mathbf{C}) = 0, \tag{3.3}$$

while using the alternating direction method of multipliers (ADMM) algorithm to foster convergence by solving a stack of easier sub-problems. As an alternative to ADMM, Sparse Subspace Clustering by Orthogonal Matching Pursuit **(SSC-OMP)** [237] approaches a similar problem with a different optimisation technique.

The previous formalism in Eq. Equation 3.3 was extended in the Deep Subspace Clustering Networks **(DSC-Nets)** [89] by having the hidden layer of an autoencoder implementing either Eq. Equation 3.2 or Eq. Equation 3.3. The Elastic Net **(EnSC)** [236] approach uses a convex combination of L^2 and L^1 constraint on $\mathbf{C}$ to increase performance while also boosting the scalability due to the usage of oracle sets to better pre-condition the solution. Dense subspace clustering **(EDSC)** [88] approaches the problem by attempting to apply the self-expressiveness loss on a dictionary which is used to describe the data while also taking into account outliers.

Once the matrix of coefficient $\mathbf{C}$ is found, an affinity graph matrix $\mathbf{W}$ is built by setting the weights on the edges between the nodes through $\mathbf{W} = \mathbf{C} + \mathbf{C}^\top$.

3.2.2 *Dictionary based* models

Even though subspace clustering methods explained in Section 3.2.1 build the affinity matrix $\mathbf{W}$ by exploiting the self-expressiveness property of data, they do not explicitly take into account the temporal dimension of time-series data while building the model adopted for HAR. As a solution, temporal regularisation was proposed by Temporal Subspace Clustering **(TSC)** [113]. Precisely, given a dictionary $\mathbf{D} \in \mathbb{R}^{d \times r}$ and a coding matrix $\mathbf{Z} \in \mathbb{R}^{r \times n}$, a collection of data points $\mathbf{X} \in \mathbb{R}^{d \times n}$ can be approximately represented as

$$\mathbf{X} \approx \mathbf{DZ}, \tag{3.4}$$

where each data point is encoded using a Least Squares regression, and a temporal Laplacian regularisation $L(Z)$ function encourages the encoding of the sequential relationships in time-series data. This can be done by minimising

$$\min_{\mathbf{Z},\mathbf{D}} \ \|\mathbf{X} - \mathbf{DZ}\|_F^2 \ + \ \lambda_1 \|\mathbf{Z}\|_F^2 \ + \ \lambda_2 L(\mathbf{Z}), \ \text{ subject to } \ \mathbf{Z} \geq 0, \ \mathbf{D} \geq 0, \tag{3.5}$$

by using the ADMM algorithm to encourage convergence by solving a stack of easier sub-problems. Different from Section 3.2.1, the affinity graph matrix $\mathbf{W}$ is given by the coding matrix $\mathbf{Z}$ by using

$$\mathbf{W}(i,j) = \frac{z_i^\top z_j}{||z_i||_2 ||z_j||_2}, \tag{3.6}$$

since the within-cluster samples (for example, the sequential neighbours of a time-series datapoint) are always highly correlated to each other [111, 112].

3.3 Temporal regularisation for HAR

Two alternative computational strategies to help and support subspace clustering methods in handling the temporal dimensions of action sequences are proposed. On the one hand, the raw skeletal trajectories were encoded using a covariance representation (described in Section 3.3.1), which is effective for solving HAR problems [22]. Additionally, a computational strategy is devised, to prune the instantaneous body poses (termed *timestamps* hereafter) whose temporal aggregation produces an action sequence. As a result of temporal pruning, the most representative timestamps can be selected, which are exploited to compress the original action sequence to a fixed duration. Consequently, this temporal pruning (namely *temporalSSC*, described in Section 3.3.2) can be adopted as a successful pre-processing step to accommodate for the usage of a subspace clustering method for HAR.

3.3.1 Covariance encoding for HAR

The idea of encoding 3D-skeleton dynamics within a single hand-crafted kernel representation has been proposed often in HAR. For instance, it has been shown that Hankel matrices can efficiently model action dynamics when used with a Hidden Markov Model [126] or a Riemannian nearest neighbours with class-prototypes [244]. Lie group [209] and associated Lie algebra [210] can effectively model human actions and activities by means of roto-translations. Likewise, generic deforming bodies can be efficiently modelled over variations of Stiefel manifolds [49]. Surely, within the class of kernel representations, a major role is played by a specific symmetric and positive definite (SPD) operator: covariance matrices (COV). Originally envisaged for image classification and detection [205], COV is an effective representation for skeleton-based HAR since capable of modelling second-order statistics. It was used in tandem with various classification pipelines, such as a temporal pyramid [87] or max-margin approaches [216, 98]. Formal studies have tried to enhance the capability of such operators in modelling non-linear correlations among the data [77, 23]. Kernel approximation was recently investigated to speed up the computational pipeline and ensure scalability towards the big data regime [22].

Even though prior work focused on the effectiveness of covariance representations applied to supervised learning pipelines (*e.g.,* in [9, 10]), its capabilities for unsupervised learning are instead demonstrated in this chapter. Using a covariance representation as the data encoder and the subspace clustering for solving HAR can be described as follows. Through either a motion capture system or a depth sensor, an action is represented as the collection-in-time

of K joints 3D positions $\mathbf{p}_1(t), \ldots, \mathbf{p}_K(t)$. By using $\mathbf{p}(t)$ to denote the column vectorisation of all such 3D positions for a fixed timestamp, an action sequence is represented as the covariance matrix

$$\Lambda = \frac{1}{T}\sum_t (\mathbf{p}(t) - \mu)(\mathbf{p}(t) - \mu)^\top, \tag{3.7}$$

where T denotes the number of timestamps and μ is the temporal average of $\mathbf{p}(t)$. The covariance matrix was then vectored through a flattening operation which exploits the property of Λ in being symmetrical. That is, $\Lambda = \Lambda^\top$. Therefore, when flattening, the diagonal elements of Λ (which are Λ_{ii}) were extracted, and the upper-triangular ones (that is, $\Lambda_{ij}, j > i$). The lower triangular part can be ignored since it is equal to the upper triangular one. Such flattening operation casts the $3K \times 3K$ matrix Λ into a $3K \cdot (3K-1)/2$ column vector. The flattened covariance representation is used as one data point, then given to the subspace clustering algorithm as the input.

3.3.2 Temporal pruning via Sparse Subspace Clustering (temporalSSC)

In addition to utilising subspace clustering as a suitable method for U-HAR, such families of techniques were also exploited to solve another task: *temporal pruning*. That refers to utilising subspace clustering on the raw joint coordinates $\mathbf{p}(t)$. Here, different from Section 3.3.1, each data point to be clustered is not an action sequence but a single data point of an action (Figure 3.2 (b)). In other words, rather than applying subspace clustering to group action sequences, subspace clustering was exploited to the group skeletal poses at a given timestamp. The general assumption is that the processed skeleton data might contain similar or redundant poses over time. To address this, temporal pruning was applied, potentially capturing the similarities over time with respect to the kinematic execution.

A relevant parameter for temporal pruning is the number of subspaces ϕ, which corresponds to the length of the new pruned skeleton data, which was set based on the following strategies:

min ϕ: the temporal length of the entire dataset is fixed to be equal to the shortest time duration across all the sequences in the skeletal dataset, this is done by using the random permutation of each sample timestamp.

min temporalSSC: subspace clustering method SSC_ADMM is used to get ϕ equal to the shortest time duration across all the sequences in the skeletal dataset.

percentage temporalSSC: the temporal length of each dataset sample is determined by selecting a percentage value for ϕ (which was chosen to keep the 75%, 50% or 25% of the sample temporal length during experiments) and applying temporalSSC.

threshold temporalSSC: the temporal length of each sample of the dataset is determined by selecting a percentage value for ϕ (which was chosen to keep the 75%, 50% or 25% of the sample temporal length during experiments), which is used as a threshold value for temporalSSC.

If a certain dataset sample has a temporal length superior to ϕ, temporalSSC is applied to match this threshold value. Once ϕ is fixed according to one of the previous strategies, all the timestamps $t_1, \ \dots \ , t_s, \ \dots$ assigned to a given subspace can be retrieved. Afterwards, an average of the corresponding skeletal positions were made $\mathbf{p}(t_1) \ , \ \dots \ , \ \mathbf{p}(t_s) \ , \ \dots$

The so-obtained average skeletal position is adopted to replace the original one, and the procedure is iterated across all the different subspaces.

For the sake of clarity, let us assume that the number of subspaces is set to be $\phi = 2$. The original action sequence has 5 timestamps which are associated with the following body poses $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5]$. Once temporalSSC is run on top of the sequence $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5]$, let assume that the corresponding output is $[1,1,2,1,2]$. So, temporalSSC is grouping $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_4$ in a subspaces and $\mathbf{p}_3, \mathbf{p}_5$ in another one. Then, the pruned action sequence was defined as $[\mathbf{p}'_1, \mathbf{p}'_2]$, where $\mathbf{p}'_1 = \frac{1}{3}(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_4)$ and $\mathbf{p}'_2 = \dfrac{1}{2}(\mathbf{p}'_3 + \mathbf{p}'_5)$.

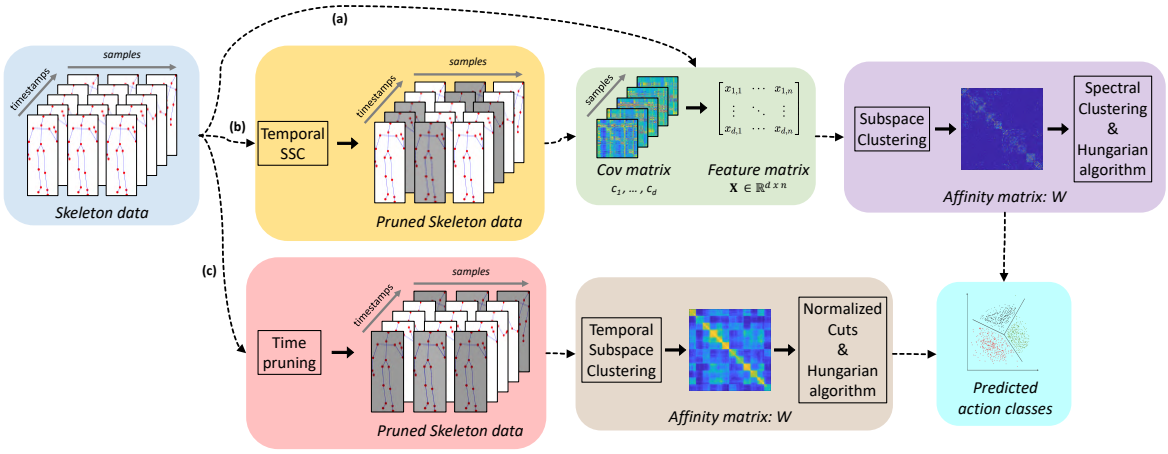


Figure 3.2 The pipeline of the proposed unsupervised methods for HAR: (a) A covariance descriptor is applied to each sample. Given the obtained covariance matrix is square and symmetrical, only the upper (can be also lower) triangular part was taken, including the diagonal and flattened it. This results in a new matrix (X) having size $samples \times features$. Following that, any subspace clustering technique can be applied to obtain an affinity graph matrix $\mathbf{W}$. Then, spectral clustering is applied using $\mathbf{W}$ to obtain cluster labels. The Hungarian algorithm finds the matching between the cluster labels (predicted action classes) and the ground-truth labels. (b) The skeletal data of each sample is temporally pruned using temporalSSC, and then the pruned data is processed as in (a). (c) Each sample is pruned by using various strategies. Afterwards, temporal subspace clustering is applied to obtain an affinity graph matrix $\mathbf{W}$. The normalized cuts are applied to obtain cluster labels, and the Hungarian algorithm matches the cluster labels with the ground-truth labels.

3.4 Methodology and Experimental Analysis

This section, through a comprehensive experimental analysis, validates the impact on U-HAR of covariance representations and temporal pruning defined in Section 3.3.1 and Section 3.3.2, respectively. Eventually, it was also demonstrated their degree of complementary to the extent that the performance of a fully unsupervised recognition pipeline can be enhanced. Interestingly, the overall performance of the proposed unsupervised approaches can almost fill the gap with state-of-the-art supervised methods. Overall, these experimental findings would help practitioners re-thinking how HAR is approached, raising attention to the desirable shift towards more agile unsupervised learning frameworks.

There exists a consistent variability in every HAR dataset due to the length of the performed actions and their complexity, the number of action classes, and the technology that was used to capture them. Prior to experimental analysis, a pre-processing step is performed [126, 244, 209, 210, 98, 23, 122] to fix one root joint located at the hip centre and compute the relative differences of all other $J-1$ 3D joint positions. This pre-processing is performed at any timestamps $t = 1,\dots,T$ to obtain a $3(J-1)$-dimensional (column) vector $p(t)$ of the relative displacements. The following dataset for experimental analysis were used (see Chapter 2, Section 2.4 of for a full description): Florence3D (F3D) [180], UTKinect-Action3D (UTK) [222], MSR 3D Action Pairs (MSRP) [148], MSR Action 3D (MSRA) [115], Gaming 3D (G3D) [13], HDM05 [139], MSRC-Kinect12 (MSRC) [64].

In order to properly ablate on their relative importance of them, it was taken considered the following computational variants of the pipeline[1]. The performance in U-HAR was monitored by taking advantage of classification accuracy, expressed as a percentage and defined as:

$$ACC(\%) = \left(1 - \frac{\#\,of\ misclassified\ labels}{\#\,of\ total\ labels}\right) \times 100 \tag{3.8}$$

[1] Code available here: https://github.com/IIT-PAVIS/subspace-clustering-action-recognition

3.4.1 U-HAR using subspace clustering and covariance descriptors

As reported in Figure 3.2(a), the first step is to apply the covariance encoding (Section 3.3.1) as the descriptor, whose result is given as an input to the state-of-the-art subspace clustering methods that are based on the self-expressiveness property of the data (Section 3.2.1) to obtain the affinity matrix **W**. These methods are: EDSC [88], OMP [237], DSCN [89], LSR [129], SSC [60], and EnSC [236].

Spectral clustering is later applied to the obtained affinity matrix **W** to infer the clustering labels by assigning each of the N datapoint $\mathbf{x}_j$ into its corresponding subspace. The final step is to apply the Hungarian algorithm to compare and map subspace labels into actual class labels [211]. Additionally, as a baseline method, two of the most popular clustering method were considered: K-means clustering (**Km**) and spectral clustering (**Sc** [159]) and all the corresponding results are reported in Table 3.1.

The best-performing method is Elastic net Subspace Clustering (EnSC) [236], which ranked highest for five of the nine datasets. For three of these five, *i.e.,* UTK, MSRA, and G3D datasets, EnSC's performance is approximately 5% better than the second-best performing method.

Dataset	Km	Sc	EDSC	OMP	DSCN	LSR	SSC	EnSC
F3D	45,58	66,05	54,42	61,40	57,02	60,47	69,12	**70,23**
UTK	34,67	66,83	52,71	58,79	69,35	57,79	73,97	**78,90**
MSRP	42,78	52,69	**51,90**	50,14	49,26	47,31	49,60	49,86
MSRA	41,11	65,17	52,69	43,99	59,91	54,40	57,27	**62,84**
G3D	31,22	64,71	44,48	45,70	62,59	64,25	65,16	**72,25**
HDM-05-14	32,36	53,35	52,42	47,67	**56,27**	51,60	49,13	**56,00**
HDM-05-65	31,41	44,46	**44,43**	36,07	30,95	42,98	35,98	42,38
MSRC	61,54	84,34	81,30	51,20	71,35	**87,04**	62,27	83,27
AVG	40,08	62,22	54,29	49,37	57,09	58,23	57,81	64,46
STD	9,63	11,28	10,82	7,58	11,92	12,66	11,63	13,38

Table 3.1 Clustering accuracy (%) of subspace clustering methods as well as k-means (Km) and spectral clustering (Sc). AVG and STD represent the average and standard deviation results in each column. The best performance for each dataset is emphasised in **bold**.

3.4.2 U-HAR using temporalSSC

For this set of experiments (Figure 3.2(b)), the first step is to apply the proposed temporal pruning approach (namely *temporalSSC* as in Section 3.3.2) as a pre-processing stage. while the rest of the pipeline follows the same setting as the pipeline of Section 3.4.1. Following, different pruning strategies for the temporal dimension of data by using SSC (see Figure 3.2(b)) were applied to raw data, before the encoding of the covariance descriptor. For the subspace clustering implementation, SSC [60] was chosen for its computational efficiency and rapid convergence time.

Table 3.2 reports the clustering accuracy of different temporalSSC strategies, along with SSC results of Table 3.1 as a baseline comparison. Results of *percentage temporalSSC* and *threshold temporalSSC* are related to the best accuracy along the different percentage values of ϕ (*i.e.,* 75%, 50% and 25%).

Only with the exception of F3D (due to its original low dimensionality of the dataset and the extreme pruning of timestamps), the results show that applying temporalSSC overall contributes positively to the clustering performance of SSC [60]: the performance improvement is up to an average 8% among all dataset, where on MSRC (the biggest dataset available) the improvement goes up to 21%.

Dataset	SSC	min ϕ	min temporalSSC	percentage temporalSSC	threshold temporalSSC
F3D	**69,12**	67,91	66,51	65,12 (ϕ = 75%)	68,84 (ϕ = 50%)
UTK	73,97	64,82	**80,90**	68,34 (ϕ = 25%)	72,86 (ϕ = 75%)
MSRP	49,60	48,88	47,88	**50,42** (ϕ = 25%)	49,58 (ϕ = 25%)
MSRA	57,27	59,61	57,09	62,66 (ϕ = 25%)	**63,02** (ϕ = 75%)
G3D	65,16	64,86	64,10	69,68 (ϕ = 75%)	**71,49** (ϕ = 75%)
HDM-05-14	49,13	**63,12**	59,04	59,33 (ϕ = 25%)	59,77 (ϕ = 25%)
HDM-05-65	35,98	41,31	**44,00**	43,66 (ϕ = 25%)	41,53 (ϕ = 50%)
MSRC	62,27	**83.79**	83,62	83,41 (ϕ = 75%)	83,14 (ϕ = 75%)
AVG	57,81	61,79	62,89	62,83	63,78
STD	11,63	11,90	13,23	11,40	12,53

Table 3.2 Clustering accuracy (%) of temporalSSC combined with different strategies and when standard SSC applied for the final clustering. ϕ is the number of subspaces utilised (Section 3.3.2). The first column shows the SSC's performances alone. AVG and STD represent the average and standard deviation results in each column. The best performance of each dataset is emphasised in **bold**.

3.4.3 U-HAR using dictionary-based subspace clustering models

With this last set of experiments (Figure 3.2(c)), the Temporal Subspace Clustering was utilised to show the effectiveness of a dictionary-based subspace clustering (Section 3.2.2) for temporal series of data when applying temporal regularization on top of the (optional) encoding through covariance (Section 3.3.1).

In Sections 3.4.1 and 3.4.2, a (flattened) covariance representation was adopted to encode the actions' kinematics. Computationally, this operation cast an action sequence with a variable temporal duration into a fixed-size embedding passed in input to subspace clustering methods based on the self-expressiveness property. Here, TSC leverages a dictionary learning framework which, together with the temporal regularization, should effectively capture the temporal variability of the data. To understand to which extent this is true, the covariance representations within the computational pipeline were *intentionally* left apart to separately evaluate these two alternative strategies of handling the temporal dimensions of the data.

TSC approach is combined with the following pruning strategies such that a constant temporal length ϕ for all the datasets in use is set as:

TSC min: the temporal length ϕ of the entire dataset is fixed to equal the shortest time duration across all the skeletal dataset sequences. This is done by using the random permutation of each timeframe.

TSC max: the opposite process of *TSC min*. For each instance, its timeframes are replicated until the temporal length ϕ is equal to the longest time duration across all the sequences in the skeletal dataset.

temporalSC + TSC: spectral clustering is used to get ϕ equal to the shortest time duration across all the sequences in the skeletal dataset.

temporalKm + TSC: k-means clustering is used to get ϕ equal to the shortest time duration across all the sequences in the skeletal dataset.

As the final steps of the pipeline, the standard Normalized Cuts [182] and Hungarian algorithms determine the clustering labels necessary for evaluation against the ground truth.

Tables 3.3 and 3.4 report the unsupervised clustering accuracy of the approach given in Section 3.4.3 (as well as illustrated in Figure 3.2(c)), where *TSCmin*, *TSCmax*, *temporalSC + TSC*, and *temporalKm + TSC* results were given with and without covariance descriptor.

Dataset	TSCmin	TSCmax	temporalSC + TSC	temporalKm + TSC	supervised s.o.t.a.
F3D	84,65	94,88	**95,81**	87,91	99,07 [105]
UTK	93,97	**99,50**	96,98	93,47	100,00 [244]
MSRP	93,48	**98,02**	88,67	96,32	95,50 [22]
MSRA	87,18	85,64	82,47	**88,51**	97,40 [22]
G3D	88,99	85,07	**90,20**	88,84	96,02 [218]
HDM-05-14	**89,80**	80,32	88,48	83,97	99,10 [22]
HDM-05-65	70,51	**75,97**	72,13	68,42	96,92 [56]
MSRC	97,96	**99,08**	98,81	99,00	98,50 [22]
AVG	88,32	89,81	89,19	88,31	
STD	7,79	8,62	8,18	8,80	

Table 3.3 Clustering accuracy (%) of TSC combined with different strategies of the uniforming temporal dimension of each dataset, *without* the usage of a covariance descriptor. The supervised state-of-the-art (s.o.t.a) results are also given. AVG and STD stand for each column's average and standard deviation results. The best-unsupervised performance of each dataset is emphasised in **bold**.

Dataset	cov TSCmin	cov TSCmax	temporalSC + TSC cov	temporalKm + TSC cov	supervised s.o.t.a.
F3D	81,40	81,86	**88,84**	87,44	99,07 [105]
UTK	**96,98**	92,96	**96,98**	83,92	100,00 [244]
MSRP	81,30	**84,70**	76,20	71,10	95,50 [22]
MSRA	79,89	83,30	81,13	**87,61**	97,40 [22]
G3D	90,20	92,61	92,46	**92,91**	96,02 [218]
HDM-05-14	**86,73**	83,82	84,84	81,63	99,10 [22]
HDM-05-65	83,57	85,62	84,64	**86,00**	96,92 [56]
MSRC	91,09	**99,05**	97,42	91,07	98,50 [22]
AVG	86,40	87,99	87,81	85,21	
STD	5,59	5,72	7,05	6,31	

Table 3.4 Clustering accuracy (%) of TSC combined with different strategies of the uniforming temporal dimension of each dataset, *with* the usage of a covariance descriptor. The supervised state-of-the-art (s.o.t.a) results are also given. AVG and STD represent the average and standard deviation results in each column. The best-unsupervised performance of each dataset is emphasised in **bold**.

The last column of both tables reports the state-of-the-art performance obtained for each dataset. It is important to highlight that the corresponding state-of-the-art methods are all supervised, while all other results given in that table are unsupervised.

The results show that applying TSC gives the best overall accuracy among all techniques adopted in this paper. Table 3.3 and Table 3.4 demonstrate that the average of results of each implementation (column) is over 85% among all cases. Except for G3D and HDM-05-65 datasets, the average accuracy of each method without covariance (*cov*) descriptor is approximately 2% better than a method with *cov* descriptor. The comparisons between the temporal frame selection approaches show that in 5-out-of-8 datasets, the pruning of data, therefore reducing its temporal dimension, is beneficial to encode and represent this type of dataset. Whereas, for MSRP and MSRC datasets, augmenting the data in temporal dimension leads to performance levels better than the state-of-the-art methods, which are all supervised.

3.5 Concluding Remarks

The context and novelties introduced in this chapter, where the focus lies on skeletal data analysis embracing a fully unsupervised approach to tackling HAR, were published in [153]. The experimental analysis was validated on eight different datasets, which are different from each other in terms of action types, the number of action classes involved, and the experimental protocol they were captured. Across such a wide variety of experimental benchmarks, this chapter's findings show that the proposed pipeline is superior to previous subspace clustering methods relying on the self-expressiveness property of data. Subspace clustering methods based on the self-expressiveness property can remarkably be enhanced in performance by covariance representation to the point that other baseline methods are systematically outperformed. On the other hand, the temporal subspace clustering method that relies on dictionary learning and temporal Laplacian regularization combined within the pipeline results in remarkably good HAR performances: This demonstrates the benefits of pruning action sequences along the temporal dimension. Overall, combining the experimental findings enables a fully unsupervised pipeline for HAR to always reduce the gap with supervised approaches while surprisingly outperforming them in some cases.

Chapter 4

Unsupervised Human Action and Emotion Recognition with Skeletal Graph Laplacian and Self-Supervised Viewpoints Invariance

After an introduction related to pure U-HAR described in Chapter 3, the follow-up step is shifting towards more extensive and complex data regimes. This is due to overcome one of the main drawbacks of subspace clustering algorithms related to the size of the given dataset: the space complexity of the affinity matrix $\mathbf{W}$ (of size $n \times n$ and required for the classification task) is $\mathcal{O}(n^2)$, where n is the size of the dataset. Therefore the applicability of such algorithms is restricted only to smaller datasets, impairing the learning of richer nuances of human actions.

To overcome this, recent literature is shifting towards the usage of *unsupervised feature representation* to solve the U-HAR task (check Figure 4.1 for a general depiction and Chapter 2 Section 2.2 for a detailed description). The focus of this chapter is to propose a novel end-to-end method with a convolutional (residual) autoencoder (Section 4.1) that uses graph Laplacian regularisation (Section 4.2) to model the skeletal geometry across the temporal dynamics of actions. Using unannotated 3D skeleton sequences, feature representations were learned (as formalised *e.g.,* in [194] and illustrated in Figure 4.2), which is then fed to an action recognition classifier (*e.g.,* 1-nearest neighbour, see Section 4.4) to validate the method performance as defined in standard evaluation protocols [247, 194, 173,

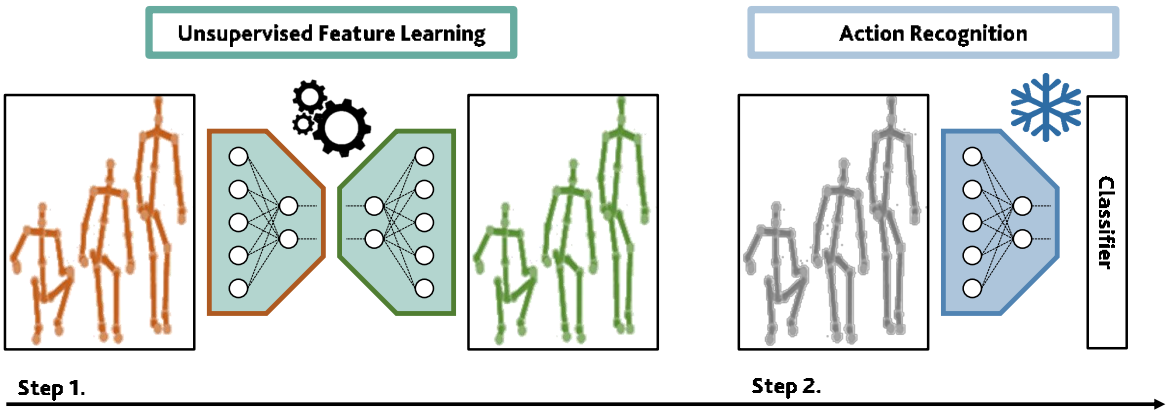


Figure 4.1 Unsupervised Human Action Recognition (U-HAR) from skeleton data. Features were computed without supervision but by learning how to reconstruct skeleton data extracted with a generative approach. U-HAR evaluation relies on applying 1-Nearest Neighbour (*1-NN*) classifier or Linear Evaluation Protocol (*LEP*) [247, 194, 173, 80, 142, 225, 100, 117].

80, 142, 225, 100, 117]. This proves the benefits of performing residual convolutions to jointly learn representations with spatio-temporal convolutions instead of relying on more complex and memory-intense architectures, which use *e.g.,* contrastive learning, GANs, gated networks, or recurrent networks [247, 194, 173, 80, 142, 225, 100, 117] (check Section 4.8 for a comparison).

To boost the performance even further, the adoption of *(graph) Laplacian regularisation* [5] ensures the learning of representations that are aware of the spatial configuration of the *skeletal geometry*. This regularisation was applied in the *reconstruction* space (*i.e.,* the space induced by the last layer of the decoder) to inject a "continuity pattern" while making this "approximation" smoother. This is the first attempt where Laplacian Regularisation is used within an unsupervised feature learning paradigm for HAR.

In addition, the proposed approach is robust towards viewpoint variations by including a self-supervised gradient reverse layer (Section 4.3) that ensures generalisation across camera views. To promote the deployment of the proposed method in practical scenarios, the problem of viewpoint invariance was also tackled, as camera positions and orientations used to capture humans very likely differ from the setup used in the tested dataset. Improvements of *viewpoint-invariance* were made possible by perturbing the original data with random rotations. Then, to increase the model's generalizability, the unsupervised learned data representations were enhanced by pairing the Laplacian-regularised reconstruction loss with a regressor head. This regressor attempts to learn the applied random rotations' parameters (rotation angles). Using adversarial training in the form of a gradient reversal layer [67],

the learned feature representation can fool this regressor, being, thus, not influenced by the rotational perturbation. This is a proxy for rotational invariance achieved with a different (and more effective - see Table 4.6 and Section 4.7) method than the Siamese network proposed in [142] (only attempting to align rotated with non-rotated data). It is important to notice that invariance was not achieved towards some annotated data features. Still, the random rotations generated were directly synthesised from the data itself: thus leveraging the concept of *self-supervision*. To validate the proposed claims, experiments were performed on three large-scale skeletal action datasets: NTU-60 (Cross-Subject and Cross-View) [181], NTU-120 (Cross-Subject and Cross-Setup) [121], and Skeletics-152 [75]. Ablation studies were performed to dissect the impact of the autoencoder, the skeletal graph Laplacian, and the adaptation of gradient reversing to U-HAR (Section 4.4). The proposed *end-to-end* approach outperformed prior unsupervised skeleton-based methods for U-HAR, and it also favourably scored *w.r.t.* state-of-the-art supervised methods, even outperforming a few of them (see Figure 4.8).

To further enhance the flexibility of the method proposed in this chapter, it was also tested *w.r.t.* a different research scenario: the Human Emotion Recognition task (HER) using full-body motion-based *3D skeletal data*. The analysis of human emotions can include several modalities, such as text, physiological signals, acoustic data, facial landmarks, facial images, or full-body motion. It is worth mentioning that representing the full-body motion with skeletal joints is principled and rooted in cognitive perception [90]. Therefore, a continuous human body moving in time is approximated with a collection of discrete trajectories. The successes in skeleton-based HER [8, 161, 46, 65, 41] highlight the effectiveness of processing full-body motion represented in terms of 3D-skeleton data.

Although unsupervised approaches for HAR are tremendously increasing their impact in terms of action classification while competing to reduce the performance gap with the fully supervised counterparts, there have been yet no attempts to apply unsupervised HER (U-HER) using skeleton data. Therefore, Section 4.6 presents an experimental analysis *w.r.t.* SOTA-unsupervised and supervised methods for HER.

The last sections are related to the extensive analyses *w.r.t.* the different components of the proposed model for both action (U-HAR) and emotion (U-HER) scenarios, proving its usefulness under various evaluation protocols with observed higher-quality feature representations, *e.g.,* with fine-tuning and end-to-end training protocol (Section 4.11), even if when it is trained with fewer data (Section 4.12), showing its remarkable transfer-ability across various domains (Section 4.17).

4.1 Convolutional Autoencoder

The proposed Convolutional Autoencoder (***AE***) input is a set of 3D human body joints in time extracted from a video sequence with one or more subjects performing an unlabelled action. Let $\mathbf{X}$ denote an input sequence of body joints represented as a $d \times m \times t$ tensor, containing the x, y, z coordinates ($d = 3$), the number of joints ($m = 25$ on NTU-60 [181], and NTU-120 [121]) and the number of timestamps t[1]. This aims at obtaining *unsupervised feature representations* by learning an autoencoder that reconstructs the input data $\mathbf{X}$ using a Mean-Squared Error (MSE) loss:

$$\mathscr{L}_{\text{MSE}} = \tfrac{1}{2}\mathbb{E}_{\mathbf{X}\sim\mathscr{B}}\left[\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2\right], \tag{4.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, *i.e.,* the Euclidean norm of the vector obtained after flattening the tensor. The MSE loss in Equation 4.1 is minimised by using gradient descent (Adam optimiser) over mini-batches $\mathscr{B}$. The reconstructed data are defined as

$$\hat{\mathbf{X}} = \mathbf{D}_\theta \circ \mathbf{E}_\varphi(\mathbf{X}) \tag{4.2}$$

and computed using an encoder-decoder architecture, where φ denotes the learnable parameters of the encoder $\mathbf{E}$ and θ are the analogous parameters for the decoder $\mathbf{D}$. The complete architecture of the convolutional autoencoder is detailed in Figure 4.2.

4.1.1 Residual blocks of convolutions

The proposed ***AE*** architecture stacks different fully-residual blocks for both encoder and decoder, whereas each block is made of convolutions capable of jointly learning spatial representations of skeletal data in time, treating each skeletal data $\mathbf{X}$ as 2D convolutions. Padded convolutions with fixed size kernels (either 1×1 or 1×3) and stride 1, applied inside $\mathbf{E}$ and $\mathbf{D}$, are capable of capturing spatial and temporal relationships of data along tensor rows for the former and along tensor columns for the latter. Hence it is called *convolutions-in-time*. In detail, within the encoder blocks, the residual layer is made of a series of three *2D-convolutional* layers (each with *ReLU* activations) stacked together. At the same time, decoder blocks share a similar structure but use instead *2D-deconvolutional* layers with the addition of *2D-BatchNorm* applied after each *ReLU* activation.

[1] To be comparable with the prior art, each skeleton sequence was cast to a fixed temporal length [194].

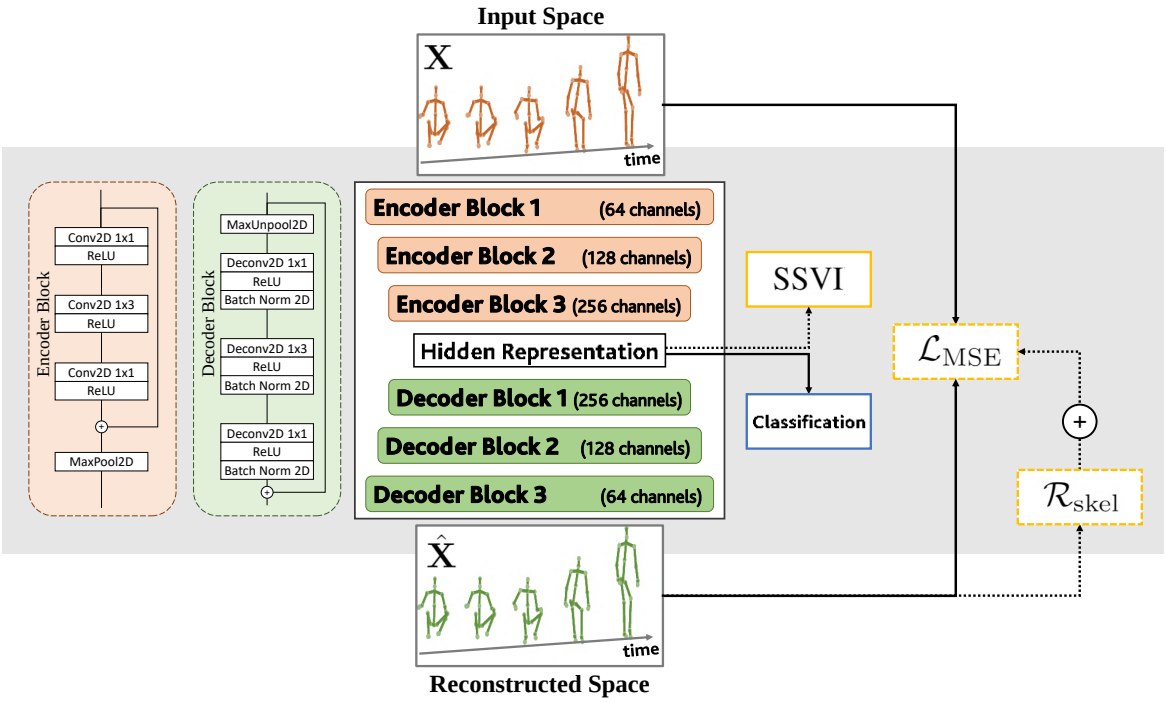


Figure 4.2 The proposed method: exploiting a convolutional autoencoder (***AE***) trained with $\mathscr{L}_{\text{MSE}}$ (Equation 4.1). In the reconstruction space, *Skeletal Laplacian Regularisation* (**L**; Section 4.2 was performed, Equation 4.7), enriching the learned (hidden) feature representations with the skeletal geometry information. The additional inclusion of a *self-supervised viewpoint-invariance* (***SSVI*** module, Section 4.3), which adapts a gradient reversal layer [67] achieves robustness towards different viewpoints. The convolutional encoder and deconvolutional decoder blocks exploit residual connections, while batch normalisation is exclusive to the decoder.

Figure 4.3 The learning curves of the *AE* model. Train/test accuracy values – *left pane* – and MSE loss – *right pane* – of the proposed model trained on DMCD [52] dataset. The proposed model achieves a stable performance at the testing time across training epochs: a favourable characteristic given the plateau in performance across training epochs.

To ensure the bottleneck structure of the convolutional autoencoder, a *MaxPool* layer is applied at the end of each encoder block, whereas a *MaxUnpool* layer is used at the beginning of each decoder block (see Figure 4.2).

4.1.2 Model Selection and Hyperparameters

The proposed model consists of a concatenation of three encoder blocks and three decoder blocks with ReLU activation layers, defined in the previous Section. It is trained for 100 epochs using Adam optimiser with a learning rate of 10^{-3} when the batch size is 128. At the end of the encoder, a fully-connected layer represents the latent space $\mathbf{z}$ of size 2048. The size of $\mathbf{z}$ was determined by testing various numerical combinations, *e.g.,* 32, 128, 512. For the convolutional autoencoder, 2048 results in the best performances (up to +10% in NTU-60 and +23% in NTU-120) out of all combinations. Thus, this value was fixed in all experiments. The features extracted from that layer were used, which are later given to the classifiers (*i.e., 1-NN* protocol [194], or Linear Evaluation Protocol [247]. In Figure 4.4, the learning curves of the proposed model after applying z-normalisation are given. As seen in this figure, the proposed model achieves a stable performance at the testing time across training epochs. This is an affirmative characteristic, also showing that representations can be learned without over-training.

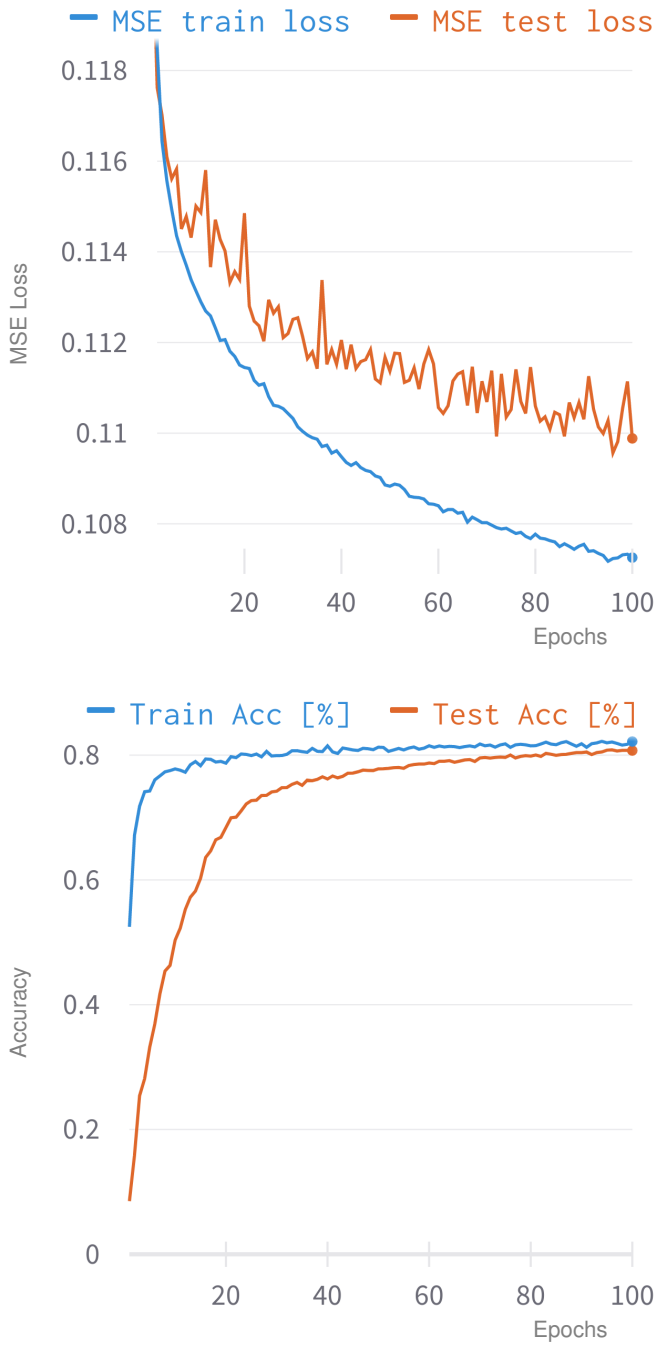


Figure 4.4 The learning curves of the *AE* model. Train/test accuracy values – *top pane* – and MSE loss – *bottom pane* – of the proposed model trained on NTU-60 [181] in the Cross-Subject protocol. The proposed model achieves a stable performance at the testing time across training epochs: a favourable characteristic given the plateau in performance across training epochs.

4.2 Skeletal Laplacian Regularisation

Belkin *et al.* [5] propose to regularise a model using the implicit geometry of the feature space, regardless of the distribution of their labels, by using the Laplacian of the graph built over the cross-similarity of examples. A similar approach was pursued by a recent end-to-end trainable approach for image denoising [150]. A different approach was followed by applying Laplacian regularisation in space while the proposed autoencoder learns to reconstruct input skeletal data (*i.e., reconstruction* space). In this way, the goal is to inject information of skeletal geometry into the proposed model.

Different from *supervised* HAR methods (*e.g.,* [122, 229]) that directly exploit the "raw" adjacency matrix to encode skeletal connectivity, a more powerful mathematical tool was taken into consideration, the graph Laplacian, since it better capitalises from the skeletal geometry. This differs from prior works, *e.g.,* [247, 194] relying on Mean-Squared Error (MSE)-based action reconstruction only.

The graph Laplacian is a well-known and established mathematical tool to analyse weighted undirected graphs. It builds upon the graph adjacency matrix $\mathbf{W}$, whose entries W_{ij} are defined such that $W_{ij} = 1$ if and only if the nodes i and j are connected through an edge. The (un-normalized) graph Laplacian $\mathbf{L}$ is easily computable from $\mathbf{W}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{4.3}$$

where $\mathbf{D}$ is the degree matrix (obtained as the diagonal matrix where its (i,i)-th element is $D_{ii} = \sum_j W_{ij}$) [50]. The Laplacian regulariser

$$\mathscr{R}(\mathbf{z}) = \sum_{i,j} W_{ij}(z_i - z_j)^2 \tag{4.4}$$

can be applied to a hidden vectorial embedding $\mathbf{z}$ to learn the geometry of the feature space (where $\mathbf{z}$ belongs to) and to capitalise from these cues to solve a semi-supervised learning paradigm [5]. This is true because, thanks to the weights W_{ij}, the alignment between the scalar components z_i and z_j can be prioritised by simply putting a stronger penalty between pairs of components that must be well aligned.

This chapter attempts to do so by promoting the alignment of skeletal joints, which are connected through a bone (*e.g., an edge exists if and only if joints are connected*).

Therefore, to correctly compute the used Graph Laplacian regularization, it was set as fixed $\mathbf{W}$, which corresponds to the adjacency matrix of each dataset in use. The results given in Section 4.9 show that such a setting is also empirically favourable compared to other ways of initialising $\mathbf{W}$. This is intended as a valid proxy for injecting the knowledge of skeletal geometry while learning the action representations. The reason why $\mathscr{R}$ is termed Laplacian regularizer lies in the fact that

$$\mathscr{R}(\mathbf{z}) = 2\,\mathbf{z}^{\top}\mathbf{L}\mathbf{z} \tag{4.5}$$

That is, $\mathscr{R}(\mathbf{z})$ implements a "$\mathbf{L}$-weighted weight decay", since

$$\mathscr{R}(\mathbf{z}) = \|\mathbf{Q}\mathbf{z}\|_2^2 \tag{4.6}$$

if setting $\mathbf{Q} = \sqrt{\mathbf{L}}$.

Unlike prior art [5, 150], Laplacian regularization was applied to the ***reconstruction space*** learned by the proposed decoder, *i.e.,* the space where $\hat{\mathbf{X}}$ belongs to. The proposed ***skeletal Laplacian regularizer*** was computed as:

$$\mathscr{R}_{\text{skel}} = \mathbb{E}_{\mathbf{X}\sim\mathscr{B}}\Big[\mathbb{E}_{t,d}\big[\hat{\mathbf{x}}^{(t,d)\top}\mathbf{L}\hat{\mathbf{x}}^{(t,d)}\big]\Big], \tag{4.7}$$

where $\hat{\mathbf{x}}^{(t,d)}$ is the m-dimensional column vector stacking the scalar (abscissae, ordinatae or quotae) coordinates along the dimension d obtained from the reconstructed sequence $\hat{\mathbf{X}}$ at time t. In Equation 4.7, the regularizer $\mathscr{R}_{\text{skel}}$ is averaged over the mini-batch $\mathscr{B}$, considering the reconstructions produced by the convolutional autoencoder across coordinates and timestamps. The Laplacian regularization attempts to inject the connectivity of the skeleton to learn a feature representation, which is aware of the ***skeletal geometry***.

This can be deemed to be a proxy of features that are aware of the fact that the representation learned, *e.g.,* from the shoulder and elbow joints, cannot be decorrelated from each other since those joints are closed in space, while there can be joints, which are more distant in space (*e.g.,* left foot vs right hand) are allowed to be more independent (as seen in Figure 4.5).

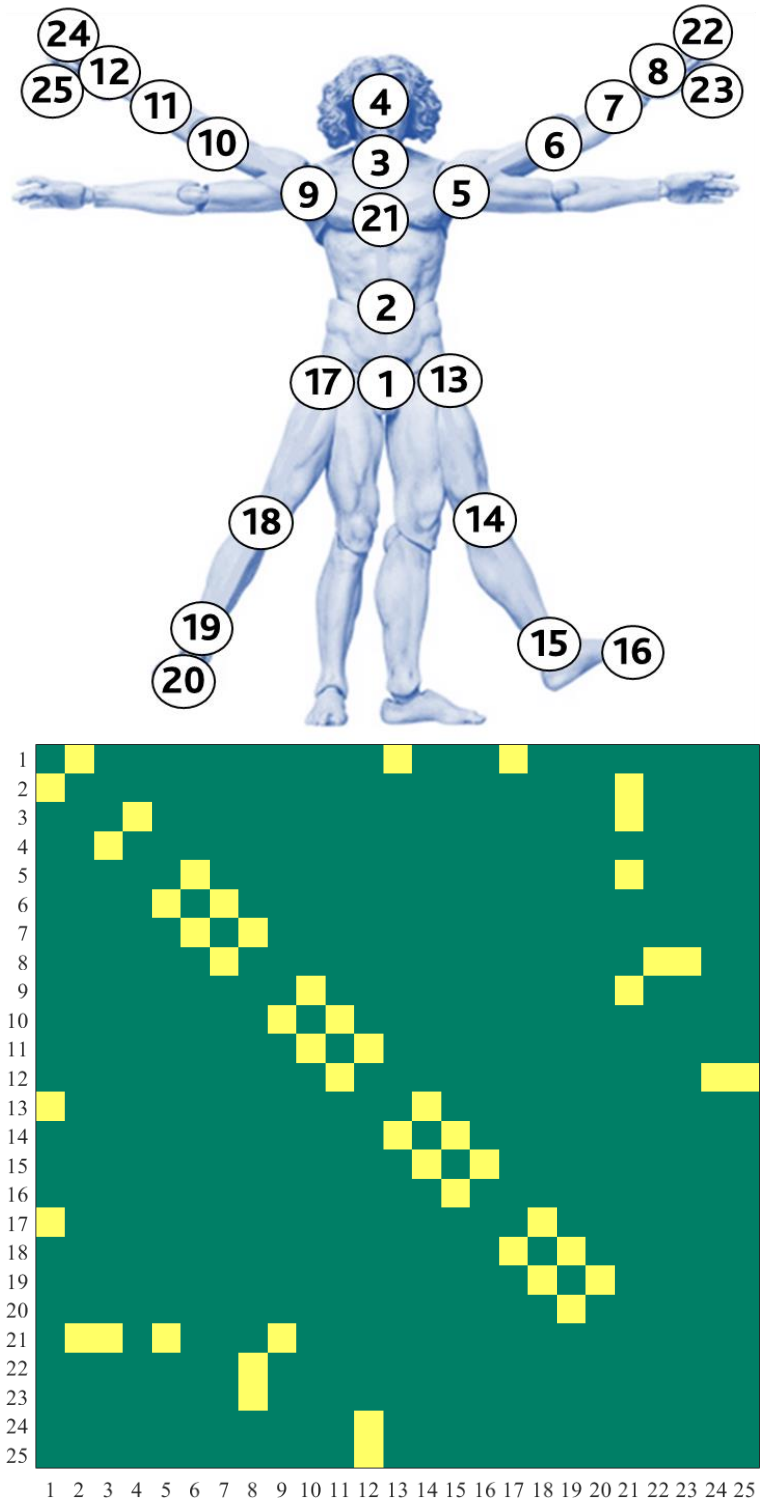


Figure 4.5 Skeletal Laplacian Regularisation. *Top*: location of the skeletal joints on NTU-60 [181]. *Bottom*: corresponding adjacency matrix $\mathbf{W}$ (binary).

4.3 Self-supervised Viewpoints Invariance (*SSVI*)

Originally proposed for domain adaptation, the gradient reversal layer (GRL) [67] is arguably helpful to achieve a better generalisation: *e.g.,* classify actions performed by multiple subjects [249]. Differently, a non-discriminative architecture (an autoencoder) for viewpoint invariance was proposed. This was performed by synthesising the skeletal joints' auxiliary rotations to simulate different viewpoints. Then, by achieving the invariance across viewpoints by a GRL layer that is fooling a predictor attempting to infer the viewpoint from the hidden representation of the autoencoder. Li *et al.* [108] adapts GRL to obtain view-invariant action representations.

However, that work differs from this chapter's proposal by (a) relying on RGB-D data and, more importantly, (b) using the annotated viewpoints of the datasets as the source and target domains and learning how to distinguish them by classification.

A viewpoint-invariant action representation can be obtained by synthesising multiple viewpoints of the original skeletal data. Geometrically, this operation can be easily framed as (right) multiplying $\mathbf{X}^t$, the $m \times 3$ matrix stacking the m 3D joints captured at a given timestamp t, by Ω defined as the product of Ω_x, Ω_y, and Ω_z, each corresponding to the independent three (planar) rotations performed around the x, y, z axis, respectively. Ω_x depends upon the pitch angle α, Ω_y depends upon the yaw angle β, and Ω_z depends upon the roll angle γ.

By the means of the so-defined Ω, $\mathbf{Z}^t = \mathbf{X}^t\Omega$ could be obtained and, hence, ***synthesize*** a rotation under a ***generated viewpoint*** by iterating the process over all timestamps t of the sequence $\mathbf{X}$ and, afterwards repeating the whole procedure for all sequences $\mathbf{X}$ in the mini-batch $\mathscr{B}$, generating the transformed sequences $\mathbf{Z}$. When $\mathbf{Z}$ is obtained from $\mathbf{X}$ according to this procedure, the action class referring to them remains unaltered in its information content, while only the viewpoint has changed.

$\mathbf{Z}$ and $\mathbf{X}$ were made indistinguishable, being the latter a proxy for an improved hidden representation that the proposed autoencoder learns from data since, in this way, the autoencoder will be robust towards different viewpoints, claiming that this requirement is a proxy for an improved viewpoint generalisation. An L1 norm was used to train a ***regressor*** that predicts the triplet $[\alpha, \beta, \gamma]$, used to rotate the data.

The ***gradient reversal layer (GRL)*** [67] took advantage to flip the gradients coming from the regressor. By doing so, invariance across synthetic rotations could be promoted by explicitly

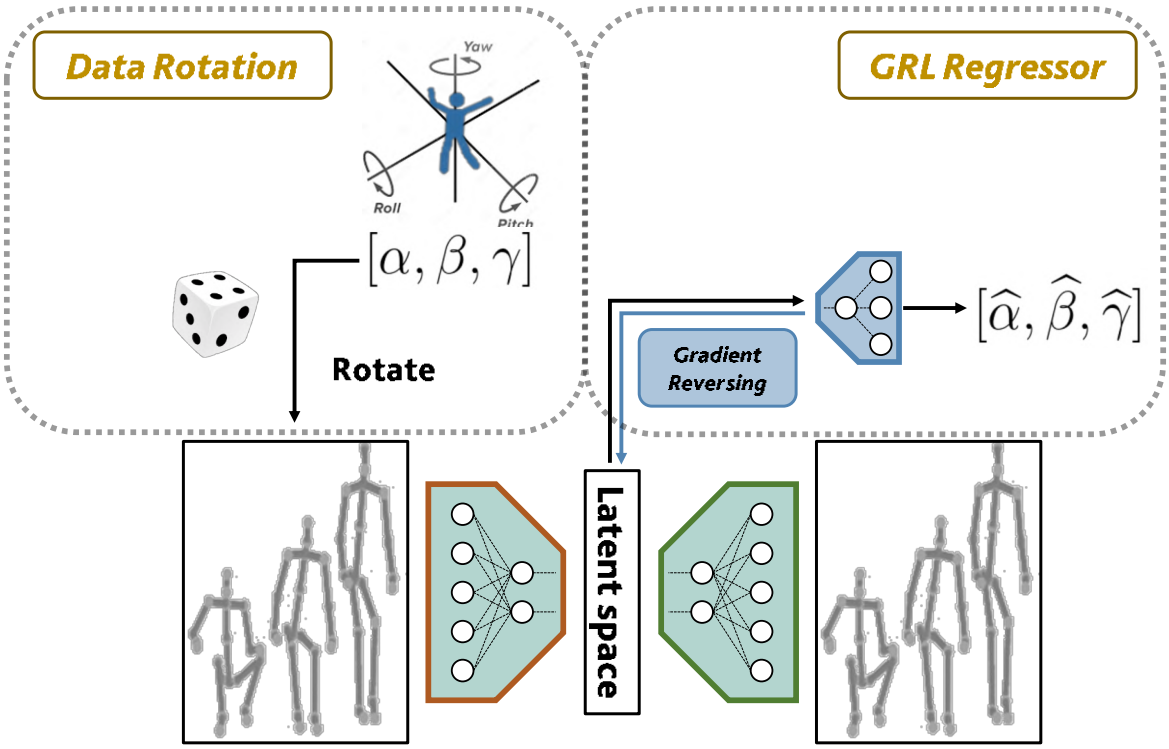


Figure 4.6 Self-Supervised Viewpoints Invariance using a regressor and a gradient reversal layer [67]. The encoder learned the hidden representation to be invariant across synthetic rotations applied to the input data $\mathbf{X}$, using the Euler's angles α, β, γ. This could be seen as a proxy to achieve viewpoints invariance and generalise across random rotations (parametrized by Euler's angles α, β, γ).

optimising the learned representation to fool a regressor attempting to predict the $[\alpha, \beta, \gamma]$ triplet used to rotate the data of each mini-batch before every forward pass.

This can be referred to as the ***self-supervised viewpoints invariance*** **(*SSVI*)** module, which is visualised in Figure 4.6 and connected to the hidden representation of the autoencoder (see Figure 4.2).

For the *SSVI* experiments, data rotation along the z-axis was applied. A sigmoid activation function, multiplied by 2π to match the Euler rotation angle, was applied for the fully connected layer of GRL. The GRL loss is an $L1$ loss calculated between the original Euler angle of rotations and the predicted Euler angle. Additionally, a penalty term was included in the GRL loss and a penalty term for the GRL layer (*i.e.,* the alpha value depicted in [67]): both are set to 10^{-3}.

4.4 Experimental Analysis

The proposed method was validated using the three large-scale skeletal action datasets (NTU-60 [181], NTU-120 [121], and Skeletics-152 [75]) and two skeletal emotion datasets (DMCD [52], and Emilya [65]). Refer to Chapter 2, Sections 2.4.8 to 2.4.10, 2.5.1 and 2.5.2 for a detailed description of those datasets used as input for the experimental analysis. The proposed method was valuated[2] on NTU-60 dataset for Cross-Subject (C-Subject) and Cross-View (C-View) settings [181], and NTU-120 for Cross-Subject (C-Subject) and Cross-Setup (C-Setup) settings [121]. For HER datasets, on par with [8], a 25-frames overlapping time-patches were applied while still retaining the temporal length of 100 frames. Figure 4.3 shows the stable performance of *AE-L w.r.t.* emotion datasets. The pseudo-code of the proposed method is given in Algorithm 1. Below, details on how a trained autoencoder is used for inference are given. All implementation details, including learning curves, can be found in Section 4.1.2.

For all experiments, the following evaluation protocols were applied:

- **Linear Evaluation Protocol (*LEP*):** This is the most standard evaluation protocol for unsupervised feature learning [247, 173, 100, 80, 142, 225, 109]. A downstream task verifies the methods by attaching a linear classifier (a fully-connected layer followed by a softmax layer) to the *frozen* encoder (shown as $\mathbf{E}$ in Section 4.1). Then, the linear classifier is trained by using the available labels.
- **1-Nearest Neighbour Predictor (*1-NN*):** Another standard evaluation protocol is applying a 1-nearest neighbour predictor [194]. In detail, the class inference of a test data $\widetilde{\mathbf{X}}$ is performed by applying a 1-nearest neighbour predictor, fed by $\mathbf{E}_{\varphi}(\widetilde{\mathbf{X}})$, and exploiting a Euclidean Gram matrix computed over the whole training set, which, in turn, is obtained using the splits of the datasets.

4.4.1 Data Pre-processing

Missing time-frames were discarded, as applied in Predict & Cluster [194]. Each skeleton was normalised in terms of bone length (in the range of [-1, 1]), followed by a regularization of the temporal length of each sample by setting it up to 100 time-frames (cutting frames of longer samples or replicating frames for shorter samples), and finally splitting data *w.r.t.* Cross-Subject, Cross-View and Cross-Setup settings of benchmarks [181, 121]. This

[2] The code is available in: www.github.com/IIT-PAVIS/UHAR_Skeletal_Laplacian

procedure is adapted from Predict & Cluster [194] except from the temporal length of each sample (choosing 100 time-frames instead of 50) and replication of the frames where instead Predict & Cluster [194] uses zero padding for the actions having less than their fixed temporal length.

4.5 U-HAR - Comparisons against the state-of-the-art

This section compares *AE-L* against the state-of-the-art (SOTA) unsupervised and supervised learning methods for the Human Action Recognition task. Only skeletal data was used in the experiments, *i.e.,* discarding the RGB and depth images, normalising data as in prior works [194] and detailed in the previous section, feeding the proposed $\mathscr{R}_{\text{skel}}$-regularized autoencoder (*AE-L*). Then, one of the two evaluations was applied: *1-NN* or *LEP*, as described in Section 4.4. It is important to highlight that the main competitors are the methods performing unsupervised feature learning. Still, this section includes the fully supervised methods in comparison to show each dataset's current upper bound performance and the gap between unsupervised and supervised methods. The corresponding results are given in Tables 4.1 to 4.3 for NTU-60 [181], NTU-120 [121], and Skeletics-152 [75] datasets, respectively. The Confusion matrices belonging to *AE-L* in testing can be found in Section 4.14. In addition, readers can find in Section 4.14.1 the complete list of action classes that benefit from the usage of Laplacian regularization. Detailed discussion is provided below.

Algorithm 1 Training of the proposed approach

Randomly initialise $\mathbf{E}_\varphi$, $\mathbf{D}_\theta$ and the *SSVI* module
Compute the skeletal graph Laplacian $\mathbf{L}$ from adjacency matrix $\mathbf{W}$
while not converged **do**
    Sample a mini-batch of data $\mathscr{B}$
    Do a forward pass through $\mathbf{E}_\varphi$ and $\mathbf{D}_\theta$, obtaining $\hat{\mathbf{X}}$
    Update $\mathbf{E}_\varphi$, $\mathbf{D}_\theta$ using the MSE loss as in Equation 4.1
    ▷ (OPTIONAL SKELETAL LAPLACIAN REGULARISATION)
    Update $\mathbf{E}_\varphi$, $\mathbf{D}_\theta$ using the $\mathscr{R}_{\text{skel}}$ loss as in Equation 4.7
    ▷ (OPTIONAL VIEWPOINTS INVARIANCE)
    Randomly sample α, β, γ in $[0, 2\pi]$
    Rotate all data in $\mathscr{B} \longrightarrow \mathscr{B}^{(\alpha,\beta,\gamma)}$
    Do a forward pass through $\mathbf{E}_\varphi$
    Update $\mathbf{E}_\varphi$ using the *SSVI* module fed by $\mathscr{B}^{(\alpha,\beta,\gamma)}$
end while
Freeze encoder parameters $\mathbf{E}_\varphi$ and append a linear classifier (*LEP*) or a 1-Nearest Neighbour classifier (*1-NN*)

4.5.1 Results for NTU-60

For NTU-60 C-Subject and C-View, the learned features of *AE-L* are superior to any other unsupervised feature learning SOTA. In addition, a favourable comparison *w.r.t.* supervised methods demonstrates the effectiveness of the proposed *AE-L*.

HAR on NTU-60 [181]			
Method	Feature Learning	C-Subject ACC (%)	C-View ACC (%)
Lie Group [170]	supervised	50.1	52.8
Cavazza *et al.* [22]	supervised	60.9	63.4
H-RNN [56]	supervised	59.1	64.0
Spatio-Temporal LSTM [122]	supervised	69.2	77.7
Part-Aware LSTM [181]	supervised	62.9	70.3
TCN [193]	supervised	74.3	83.1
VA-LSTM [241]	supervised	79.2	87.7
DGNN [183]	supervised	89.9	96.1
4s-ShiftGCN [28]	supervised	90.7	96.5
CTR-GCN [26]	supervised	92.4	96.8
U-HAR on NTU-60 [181] – *1-NN* Protocol [194]			
P&C FS* [194]	unsupervised	50.6	76.3
P&C FW* [194]	unsupervised	50.7	76.1
Baseline ***AE***	unsupervised	*50.1*	*80.4*
AE	unsupervised	*52.3*	*81.0*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***54.1***	***83.1***
U-HAR on NTU-60 [181] – Linear Evaluation Protocol (*LEP*) [247]			
LongT GAN [247]	unsupervised	39.1	48.1
MS^{2}L [117]	unsupervised	52.5	–
PCRP [225]	unsupervised	53.9	63.5
VAE-PoseRNN [100]	unsupervised	56.4	63.8
AS-CAL [173]	unsupervised	58.5	64.6
MM-AE [80]	unsupervised	61.2	70.2
EnGAN-PoseRNN [100]	unsupervised	68.6	77.8
SkeletonCLR joint [109]	unsupervised	68.3	76.4
Baseline ***AE***	unsupervised	*68.5*	*84.3*
AE	unsupervised	*69.2*	*85.1*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***69.9***	***85.4***

Table 4.1 Performance comparisons on NTU-60 [181] in terms of accuracy (%). The numbers of methods proposed in this chapter are in *italic*. Improved performance over the prior art is underlined. The best of all unsupervised results are in **bold**. Refer to [36] for the complete list of supervised benchmark results. Only a few example approaches that the proposed method surpasses and the top scorers are listed herein. *FS and FW stand for a decoder with "fixed states" and "fixed weights", respectively [194].

Cross-Subject evaluation protocol

Ablation study shows that *AE-L* improves the performance of *AE* model, demonstrating the advantages of using Laplacian regularization: +1.8% in *1-NN*, +0.7% in *LEP* for NTU-60 C-Subject setting. The proposed *AE* is preferable to the Baseline *AE* (*i.e.,* not using residual layers in design) as performing +2.2% in *1-NN*, +0.7% in *LEP* for NTU-60 C-Subject, showing the contribution of using residual convolutions layers. For NTU-60 C-Subject, the learned features of *AE-L* and *AE* models are superior to P&C [194]: +3.5% as compared to P&C FS [194] and +3.4% as compared to P&C FW [194]. While exploiting *LEP*, *AE-L* again performs better than the approaches based on RNNs [100, 173], performing +11.4% better than AS-CAL [173] and +8.7% than MM-AE [80]. *AE-L* improves over VAE-PoseRNN [100], EnGAN-PoseRNN [100] and SkeletonCLR joint [109] by +13.5%, +1.3%, +1.6%, respectively. It also surpasses MS^2L [117] (+17.4%), which benefits from contrastive learning, motion prediction, and jigsaw puzzle recognition.

Cross-View evaluation protocol

For NTU-60 [181] C-View, *AE-L* improves the performance by +6.8% and +7.0% over P&C FS [194] and P&C FW [194], respectively within the *1-NN* Protocol. In the same protocol, the ablation study shows that *AE-L* improves the performance of Baseline *AE* by +2.1%, and using residual layers (*i.e.,* the proposed *AE*) performs 0.6% better than not using (Baseline *AE*). On NTU-60 [181] C-View, with *LEP*, the superiority of *AE-L* is much visible such that it notably exceeds LongT GAN [247] (+37.3%), PCRP [225] (+21.9%), AS-CAL (+20.8%), VAE-PoseRNN (+21.6%), MM-AE (+15.2%), EnGAN-PoseRNN (+7.6%) and SkeletonCLR joint [109] (+9%). *AE* also surpasses "Baseline *AE*" by 0.8%, again showing residual layers' positive contribution.

Comparison with supervised methods

A performance comparison of the proposed *AE-L* with SOTA-supervised skeleton-based HAR approaches is also included, although they are not direct competitors. This comparison includes kernel-based methods [170, 22] and the methods realising feature learning [56, 122, 181, 241, 193, 183, 28, 26] with several different deep learning architectures, *e.g.,* RNNs, LSTMs, CNNs, and Graph Convolutional Networks (GCNs). Although based on unsupervised learning, *AE-L* can achieve better performance than the fully supervised kernel-based methods [170, 22], with a +7.2% to +19.8% improvement in C-Subject and a +22% to +32.6% improvement in C-View setting. It also outperforms several fully supervised

deep architectural methods: H-RNN [56] (providing an increase of 10.8% in C-Subject and up to 21.4% in C-View), Spatial-Temporal LSTM [122] (resulting in a boost of +0.7% in C-Subject and up to +7.7% in C-View) and part-aware LSTM [181] (achieving an improvement of +7% in C-Subject and up to +15.1% in C-View) while performing better than temporal CNN (TCN) [193] (up to +2.3%) in C-View setting. These results show that the proposed unsupervised residual convolutions with Laplacian regularization exceed even supervised GRUs, RNNs, and LSTMs (and variants) for HAR.

Besides the favourable results of *AE-L* it is important to note that fully supervised techniques, *e.g.,* [241, 183, 28, 26], perform better than *AE-L*. These supervised methods mainly implement GCNs, and some of them additionally adapt LSTMs [188], or a variable temporal dense block [220]. As expected, the best performing method for this dataset is [26], with 92.4% and 96.8% in C-Subject and C-View, respectively.

4.5.2 Results for NTU-120

For NTU-120 C-Subject, *AE-L* once again performs better than all unsupervised SOTA, showing a complementary behaviour noticed in NTU-60 [181].

Cross-Subject evaluation protocol

Ablation study shows that *AE-L* improves the performance of *AE* model, demonstrating the advantages of using Laplacian regularization: +1.4% in *1-NN*, +2% in *LEP* for NTU-120 C-Subject setting. The proposed *AE* is preferable to the Baseline *AE* (*i.e.,* not using residual layers in design) as performing +0.8% in *1-NN*, +0.7% in *LEP* for NTU-120 C-Subject, showing the contribution of using residual convolutions layers. For NTU-120 C-Subject, *AE-L* outperforms P&C [194] (+0.7%) when *1-NN* is applied, with an increase in performance *w.r.t.* both AS-CAL [173] (+10.5%) and PCRP [225] (+17.4%) in *LEP*.

Cross-Setup evaluation protocol

On NTU-120 [121] C-Setup, *AE-L* again performs better than P&C within the *1-NN* Protocol (+2.0%), and in *LEP*, it performs better than AS-CAL and PCRP by margins of +13.2% and +17.3%, respectively. In this setting, *AE* achieves better results than "Baseline *AE*" by +0.2% for *1-NN* and +1.5% for *LEP*.

HAR on NTU-120 [121]			
Method	Feature Learning	C-Subject ACC (%)	C-Setup ACC (%)
Part-Aware LSTM [181]	supervised	25.5	26.3
Soft RNN [84]	supervised	36.3	44.9
Dynamic Skeletons [83]	supervised	50.8	54.7
Spatio-Temporal LSTM [122]	supervised	55.7	57.9
Internal Feature Fusion [122]	supervised	58.2	60.9
Qiuhong *et al.* [95]	supervised	58.4	57.9
DualHead-Net [25]	supervised	88.2	89.3
EfficientGCN-B4 [192]	supervised	88.7	89.1
CTR-GCN [26]	supervised	88.9	90.6
U-HAR on NTU-120 [121] – *1-NN* Protocol [194]			
P&C† [194]	unsupervised	41.7	42.7
Baseline ***AE***	unsupervised	*40.2*	*44.3*
AE	unsupervised	*41.0*	*44.5*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***42.4***	***44.7***
U-HAR on NTU-120 [121] – Linear Evaluation Protocol (*LEP*) [247]			
PCRP [225]	unsupervised	41.7	45.1
AS-CAL [173]	unsupervised	48.6	49.2
Baseline ***AE***	unsupervised	*56.4*	*60.3*
AE	unsupervised	*57.4*	*61.8*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***59.1***	***62.4***

Table 4.2 Performance comparisons on NTU-120 [121] in terms of accuracy (%). The numbers of methods proposed in this chapter are in *italic*. Improved performance over the prior art is underlined. The best of all unsupervised results are in **bold**. Refer to [35] for the full list of supervised benchmark results. Herein, only a few example approaches that the proposed method surpasses, as well as the top scorers, are listed. †Taken from PCRP [225].

Comparison with supervised methods

The performance gap between the unsupervised and supervised learning methods is bigger in NTU-120 [121] C-Subject and C-Setup split compared to the NTU-60 dataset. Still, *AE-L* is able to achieve better performance than other more complex methods, *e.g.,* [181, 83, 122], which rely on variations of LSTM and RNNs. On the other hand, similar to the NTU-60 dataset's results, the best performance achieved in NTU-120 is also based on GCNs (*e.g.,* [192, 26]).

4.5.3 Results for Skeletics-152

HAR on Skeletics-152 [75]		
Method	Feature Learning	ACC (%)
4s-ShiftGCN [28]	supervised	56.1
MS-G3D [125]	supervised	56.4
U-HAR on Skeletics-152 [75] – *1-NN* Protocol [194]		
P&C FS* [194]	unsupervised	45.1
P&C FW* [194]	unsupervised	47.4
Baseline ***AE***	unsupervised	*46.2*
AE	unsupervised	*48.5*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***49.0***
U-HAR on Skeletics-152 [75] – Linear Evaluation Protocol (*LEP*) [247]		
MS^{2}L [117]	unsupervised	20.4
PCRP [225]	unsupervised	21.1
AS-CAL [173]	unsupervised	25.9
LongT GAN [247]	unsupervised	30.7
SkeletonCLR joint [109]	unsupervised	37.3
Baseline ***AE***	unsupervised	*45.0*
AE	unsupervised	*46.4*
AE-L (*AE* + $\mathscr{R}_{\text{skel}}$)	unsupervised	***52.0***

Table 4.3 Performance comparisons on Skeletics-152 [75] in terms of accuracy (%). The numbers of methods proposed in this chapter are in *italic*. Improved performance over the prior art is underlined. The best of all unsupervised results are in **bold**. Refer to [36] for the full list of supervised benchmark results. Herein, only few example approaches that the proposed method surpasses, as well as the top scorers, are listed. *FS and FW stand for a decoder with "fixed states" and "fixed weights", respectively [194].

This section compares the performance of *AE-L* against supervised and unsupervised SOTA methods. Especially in *LEP*, *AE-L* has promising results *w.r.t.* the supervised SOTA, which performs only 4.1% and 4.4% less than 4s-ShiftGCN [28] and MS-G3D [125], respectively. It is important to notice that 4s-ShiftGCN [28] and MS-G3D [125] are based on multiple numbers of spatial-temporal graph convolutional blocks, *i.e.,* more complex than the proposed architecture, also requiring fully annotated large-scale training data. The performance gaps between *AE-L* and 4s-ShiftGCN [28] and MS-G3D [125] decreased in this dataset compared to the NTU-60 dataset. As for unsupervised results, *AE-L* performs better than *1-NN* competitors (+3.9% over P&C FS [194], and +1.6% over P&C FW [194]), exceeding *LEP* competitors as well (MS^{2}L [117] +31.6%, PCRP [225] +30.9%, AS-CAL [173] +26.1%, LongT GAN [247] +21.3%, and SkeletonCLR joint [109] +14.7%).

4.6 U-HER - Comparisons against the state-of-the-art

This section compares *AE-L* against the state-of-the-art (SOTA) unsupervised and supervised learning methods for the Human Emotion Recognition task. It is important to highlight that the main competitors are the methods performing unsupervised feature learning. Still, fully supervised methods were included in the proposed comparisons to show each dataset's current upper bound performance and the gap between unsupervised and supervised methods.

4.6.1 Results for DMCD

Performance of *AE-L* on DMCD dataset [52] greatly outperforms both supervised (+22.4% over Beyan *et al.* [8]) and unsupervised counterparts: exceeding P&C FS [194] (+21.3%), P&C FW [194] (+11.9%), MS^{2}L [117] (+69.8%), PCRP [225] (+66.2%), AS-CAL [173] (+54.4%), LongT GAN [247] (+21.4%), and SkeletonCLR joint [109] (+9.3%).

4.6.2 Results for Emilya

Evaluating *AE-L* on Emilya dataset [65] highlights the comparable results *w.r.t.* supervised counterpart Crenn *et al.* [41] and even outperforming Fourati *et al.* [65] (+7.3%). As for comparisons against unsupervised SOTA, *AE-L* is superior than P&C FS [194] (+10.2%), MS^{2}L [117] (+49.9%), PCRP [225] (+50.2%), AS-CAL [173] (+35.6%), LongT GAN [247] (+11.6%), and SkeletonCLR joint [109] (+2.1%), showing once again its effectiveness for HER.

HAR on DMCD [52]		
Method	Feature Learning	F1-score
Beyan *et al.* [8]	supervised	74.7
U-HAR on DMCD [52] – *1-NN* Protocol [194]		
P&C FS* [194]	unsupervised	75.1
P&C FW* [194]	unsupervised	84.5
AE-L ($AE + \mathscr{R}_{\text{skel}}$)	unsupervised	***96.4***
U-HAR on DMCD [52] – Linear Evaluation Protocol (*LEP*) [247]		
MS^{2}L [117]	unsupervised	27.3
PCRP [225]	unsupervised	30.9
AS-CAL [173]	unsupervised	42.7
LongT GAN [247]	unsupervised	75.7
SkeletonCLR joint [109]	unsupervised	87.8
AE-L ($AE + \mathscr{R}_{\text{skel}}$)	unsupervised	***97.1***

Table 4.4 Performance comparisons on DMCD [52] in terms of F1-score. The numbers of methods proposed in this chapter are in *italic*. Improved performance over the prior art is underlined. The best of all unsupervised results are in **bold**. *FS and FW stand for a decoder with "fixed states" and "fixed weights", respectively [194].

HAR on Emilya [65]		
Method	Feature Learning	ACC (%)
Fourati *et al.* [65] $^\triangledown$	supervised	75.0
Beyan *et al.* [8] $^\triangledown$	supervised	90.5
Crenn *et al.* [41] $^\diamond$	supervised	82.2
Beyan *et al.* [8] $^\diamond$	supervised	91.3
U-HAR on Emilya [65] – *1-NN* Protocol [194]		
P&C FS* [194] $^\diamond$	unsupervised	65.0
P&C FW* [194] $^\diamond$	unsupervised	**76.8**
AE-L ($AE + \mathscr{R}_{\text{skel}}$) $^\triangledown$	unsupervised	*71.8*
AE-L ($AE + \mathscr{R}_{\text{skel}}$) $^\diamond$	unsupervised	*75.2*
U-HAR on Emilya [65] – Linear Evaluation Protocol (*LEP*) [247]		
MS^{2}L [117] $^\diamond$	unsupervised	32.4
PCRP [225] $^\diamond$	unsupervised	32.1
AS-CAL [173] $^\diamond$	unsupervised	46.7
LongT GAN [247] $^\diamond$	unsupervised	70.7
SkeletonCLR joint [109] $^\diamond$	unsupervised	80.2
AE-L ($AE + \mathscr{R}_{\text{skel}}$) $^\triangledown$	unsupervised	*76.4*
AE-L ($AE + \mathscr{R}_{\text{skel}}$) $^\diamond$	unsupervised	***82.3***

Table 4.5 Performance comparisons on Emilya [65] in terms of accuracy (%). The numbers of methods proposed in this chapter are in *italic*. Improved performance over the prior art is underlined. The best of all unsupervised results are in **bold**. *FS and FW stand for a decoder with "fixed states" and "fixed weights", respectively [194]. $\diamond$ and $\triangledown$ stand for the cross validation set-up applied in [41], and [65], respectively.

4.7 Transfer across viewpoints for U-HAR

U-HAR: Transfer Across Viewpoints		# of params.	where?	NTU-60 C-View	NTU-120 C-Setup
Baseline	[194]	0.58M	input (pre-proc)	76.3%	42.7%
SeBiReNet	[142]	0.27M	input (data-aug)	79.7%	–
GRAE	(*AE* + *SSVI*)	*0.39M*	*feature space*	*81.9%*	*47.0%*
GRAE-L	(*AE* + $\mathscr{R}_{\text{skel}}$ + *SSVI*)	*0.39M*	*feature space*	***82.4%***	***48.9%***

Table 4.6 A comparison of the proposed *SSVI* module plugged into either *AE* and *AE-L* (using the Linear Evaluation Protocol [247], the numbers reported in *italic*) with published results of [194, 142].

Since U-HAR is, by design, better tailored to real-world applications, this section aims to push the proposed approach to the limit and compete against SeBiReNet [142] to transfer across viewpoints. SeBiReNet and ***GRAE-L*** (*AE* + $\mathscr{R}_{\text{skel}}$ + *SSVI*) leverage random rotational noise to perturb the input data with a sharp algorithmic difference. The two-stream Siamese architecture of SeBiReNet [142] is jointly fed by rotated and non-rotated data while using non-adversarial optimization to promote viewpoints invariance. Differently, the proposed method exploits gradient reversing [67] to achieve viewpoint invariance in a model which is fed by *rotated data only*, attempting to fool a regressor (one ReLU-hidden layer MLP with a sigmoid readout layer) to predicting the triplet of Euler's angles used to rotate each mini-batch (see Algorithm 1). By relying on a single stream, and as opposed to having two lightweight streams helping each other in generalising better [142], the proposed network is deeper (also depends upon a greater number of learnable parameters - 0.27M versus 0.39M, see Table 4.6) but achieves a better invariance across viewpoints. Furthermore, the proposed approach does not benefit from auxiliary skeletal datasets as commonly happening in unsupervised domain adaptation [67] (*e.g.,* in SeBiReNet [142], a pre-training is performed on Cambridge-Imperial APE dataset, and then transfer learning is applied for NTU-60). As seen in Table 4.6, *GRAE* (*AE* + *SSVI*) and *GRAE-L* (*AE* + $\mathscr{R}_{\text{skel}}$ + *SSVI*) approaches score favourably against SeBiReNet [142], and *GRAE-L* has a +2.7% on NTU-60 C-View setting. In the same table, a comparison with the baseline solution [194] was reported, applying view-invariant transformations to "clean" the data from rotations as pre-processing.

Notably, despite being trained with more complex data to be fitted (the single-stream of *GRAE-L* never sees non-rotated data), the proposed method still outperforms this baseline by big margins (+6.1% on NTU-60 [181] C-View and +6.2% on NTU-120 [121] C-Setup).

4.8 Time and Space Complexity

	# of Parameters	NTU-60 [181]		NTU-120 [121]		Skeletics-152 [75]	DMCD [52]	Emilya [65]
		C-Subject	*C-View*	*C-Subject*	*C-Setup*			
P&C FS* [194]	57.7M	35.06 *s*	43.06 *s*	104.57 *s*	123.58 *s*	24.68 *s*	29.07 *s*	28.75 *s*
P&C FW* [194]	57.7M	35.35 *s*	40.58 *s*	104.10 *s*	123.74 *s*	24.12 *s*	28.90 *s*	27.46 *s*
MS^{2}L [117]	11.2M	24.66 *s*	29.81 *s*	64.63 *s*	69.59 *s*	17.14 *s*	24.37 *s*	17.92 *s*
SkeletonCLR joint [109]	3.6M	16.96 *s*	19.36 *s*	51.53 *s*	58.99 *s*	11.52 *s*	20.25 *s*	15.20 *s*
LongT GAN [247]	10.2M	15.84 *s*	18.85 *s*	64.56 *s*	80.32 *s*	11.40 *s*	14.05 *s*	12.47 *s*
PCRP [225]	19.4M	14.30 *s*	16.44 *s*	41.97 *s*	48.97 *s*	10.39 *s*	10.84 *s*	11.27 *s*
AS-CAL [173]	340K	9.42 *s*	10.37 *s*	28.45 *s*	33.54 *s*	6.71 *s*	10.19 *s*	8.08 *s*
AE-L	38.5M	**3.41** *s*	**3.91** *s*	**9.91** *s*	**11.91** *s*	**2.52** *s*	**3.84** *s*	**3.08** *s*

Table 4.7 Inference time of one epoch (in *seconds*) of the proposed *AE-L* and unsupervised competitors. All experiments were performed on a single machine equipped with an Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz, 64GB RAM, and a single NVIDIA RTX2080 GPU. *FS and FW stand for a decoder with "fixed states" and "fixed weights", respectively [194].

Table 4.7 reports the time complexity of *AE-L* and the most prominent unsupervised competitors in terms of the inference time of one epoch using the testing split of both HAR and HER datasets. All analyses were performed with the machine equipped with an Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz, 64GB of RAM, and a single NVIDIA RTX2080 GPU. In the same table, the space complexity of the proposed model and its counterparts in terms of the number of parameters is also declared. Despite *AE-L* having higher (or comparable) space complexity in terms of the number of parameters *w.r.t.* to some other architectures, it achieves the lowest per-epoch inference time, proving the effectiveness of using residual convolutional layers instead of relying on contrastive-based approaches, GANs, gated networks, or recurrent networks. It is also noticeable that the proposed method has a low space complexity compared to P&C [194], which is based on recurrent networks.

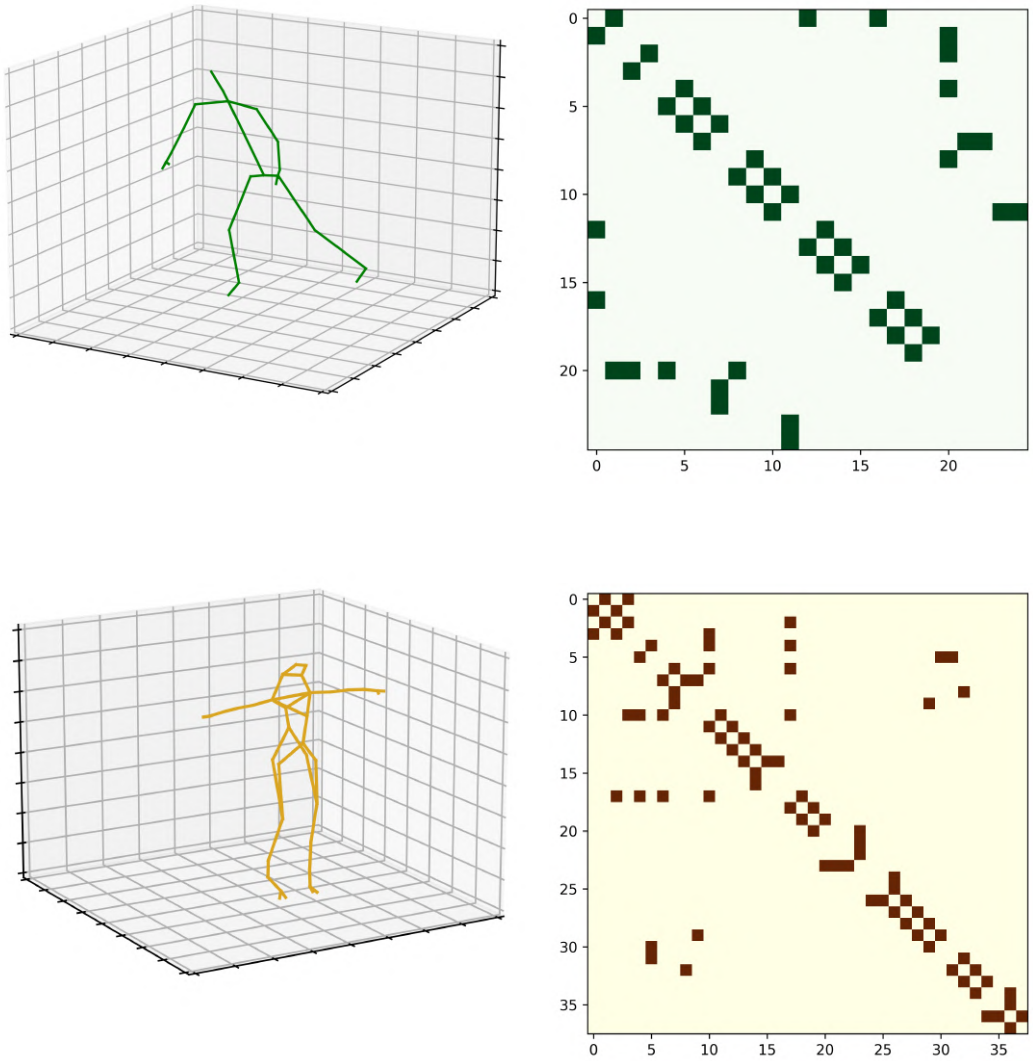


Figure 4.7 (*Top-Left*) The location of the skeletal joints in NTU-60 [181], (*Top-Right*) The corresponding binary adjacency matrix for NTU-60 [181], (*Bottom-Left*) The location of skeletal joints in DMCD [52], (*Bottom-Right*) The corresponding binary adjacency matrix for DMCD.

4.9 Graph Laplacian weight matrix initialisation

The goal of this section is to examine how the initialisation of Graph Laplacian weight matrix $\mathbf{W}$ affects the proposed method's performance: *AE-L*. This is done in order to promote the alignment of skeletal joints, connected through a bone (*e.g.,* ***an edge exists if and only if joints are connected***). The reason behind this is to inject the knowledge of skeletal geometry while learning action representations. This is referred as *Fixed W*, a binary and symmetric $n \times n$ skeleton adjacency matrix, including the connectivity between pairs of skeletal joints (as shown in Figure 4.7). n is equal to the number of joints of each skeleton.

	NTU-60 [181]		NTU-120 [121]		Skeletics-152 [75]	DMCD [52]	Emilya [65]
	C-Subject	C-View	C-Subject	C-Setup			
W Initialisation – *1-NN* Protocol [194]							
Baseline *AE*	50.1	80.4	40.2	44.3	46.2	75.3	52.8
AE	52.3	81.0	41.0	44.5	48.5	78.9	55.3
AE-L w/ Random *W*	52.6	80.3	40.9	42.8	47.7	90.8	71.8
AE-L w/ Fixed *W* (*AE-L*)	**54.1**	**83.1**	**42.4**	**44.7**	**49.0**	**96.4**	**75.2**
W Initialisation – Linear Evaluation Protocol (*LEP*) [247]							
Baseline *AE*	68.5	84.3	56.4	60.3	45.0	81.4	55.7
AE	69.2	85.1	57.4	61.8	46.4	86.2	58.1
AE-L w/ Random *W*	69.0	84.8	57.1	60.9	50.3	92.5	74.5
AE-L w/ Fixed *W* (*AE-L*)	**69.9**	**85.4**	**59.1**	**62.4**	**52.0**	**97.1**	**82.3**

Table 4.8 Ablation study and the effect of Graph Laplacian Weight Matrix (W) initialisation for *AE-L*, using a random weight matrix $\mathbf{W}$ or the fixed one. Baseline *AE* refers to the proposed model (*AE-L*) without residual layers within. *AE* refers to *AE-L* without the Graph Laplacian regularisation. All the scores are in terms of accuracy (%) except the F1-scores (%) given for DMCD dataset [52].

For action datasets *i.e.,* NTU-60 [181], NTU-120 [121], and Skeletics-152 [75], n is set to 25 joints. For emotion datasets *i.e.,* DMCD [52] and [65], n is set to 38 and 28 joints, respectively. The W_{ij} entries of $\mathbf{W}$ are defined such that $W_{ij} = 1$ if and only if the joints i and j are connected through an edge (*i.e.,* a *bone*); otherwise, $W_{ij} = 0$.

A natural alternative to this approach is randomly initialising the weight matrix $\mathbf{W}$ ($n \times n$). This setting is called *Random W*, and the range of W_{ij} is $[0, 1]$.

Table 4.8 shows and demonstrates the effectiveness of the proposed *AE-L* with Fixed *W* against Random *W* by comparing as well with different model ablations: "Baseline *AE*"

(*i.e.,* proposed method without Laplacian regularization *and* without residual layers), and "*AE*" (*i.e.,* proposed method without Laplacian regularization).

Ablation study (Table 4.8) shows that *AE-L* improves the performance of the *AE* model, demonstrating the advantages of using Laplacian regularization in all datasets and all evaluation protocols. The comparisons among initialising the Graph Laplacian weight matrix **W** in the proposed way (*i.e., Fixed W*) versus initialising it randomly (*Random W*) show that *Fixed W* achieves better performance independent of the number and the position of the joints in the skeletal data, showing that injecting the skeletal geometry into the regularization is useful. Results also show that *AE* is preferable to Baseline *AE*, *i.e.,* using residual layers in the design contributes positively to all datasets and evaluation protocols.

HAR datasets evaluated with the *1-NN* evaluation protocol show that the usage of Laplacian regularization brings improvements for NTU-60 [181] up to +4% for *C-Subject* and +2.8% for *C-View*, same behaviour for NTU-120 [121] with increments of +2.2% for *C-Subject* and +1.9% for *C-Setup*. Similarly, for Skeletics-152 [75], *AE-L* achieves up to +2.8% performance increase. When evaluated with the *LEP* evaluation protocol, HAR datasets got an increase in performance of +1.4% for NTU-60 [181] *C-Subject* and +1.1% for NTU-60 [181] *C-View*, +2.7% for NTU-120 [121] *C-Subject* and +2.1% for NTU-120 [121] *C-Setup*, and +7% for Skeletics-152 [75].

Especially for HER datasets, this improvement is remarkable. For DMCD dataset [52], the increase is up to +21.1% in *1-NN* evaluation protocol and +15.7% in *LEP* evaluation protocol. As for Emilya dataset [65], the increase is up to +22.4% in *1-NN* evaluation protocol and +26.6% in *LEP* evaluation protocol.

4.10 Using synthetic data in training

		AE	*AE-L*	*GRAE* (*AE* + *SSVI*)	*GRAE-L* (*AE-L* + *SSVI*)
NTU-60 [181] C-View	Real data (pre-processed)	85.1	85.4	∼	∼
	Real + Synthetic data	80.4	80.6	∼	∼
	Synthetic data	80.1	81.3	**81.9**	**82.4**
NTU-120 [121] C-Setup	Real data (pre-processed)	61.8	62.4	∼	∼
	Real + Synthetic data	45.7	45.2	∼	∼
	Synthetic data	46.1	46.4	**47.0**	**48.9**

Table 4.9 Performances (accuracy) of the proposed methods using the real and/or synthetic data in training. *Notice that methods with* SSVI *rely only on synthetic data.*

This section investigates the impact of using synthetic data in training for C-View and C-Setup scenarios. Synthetic data were obtained as described in Chapter 4 Section 4.3, and it was fixed for all experiments in Table 4.9. The models are trained with a) real data only, b) real + synthetic data, c) synthetic data only. Real data refers to pre-processed data, so-called clean data in Chapter 4 Section 4.7, which is already aligned to the same viewpoint.

Recalling that *SSVI*-based experiments rely only on synthetic data, whose amount is as much as the real training data, the training set size of real + synthetic experiments is twice of real only and synthetic only. Synthetic data includes rotational perturbations of *not pre-processed* real data. Thus, experiments only with synthetic data and real + synthetic data result in performance degradation for all models. Experiments with real data perform the best out of all, but it is important to notice that *the applied pre-processing is mostly not applicable in real-world applications as the viewpoints might not be known*. When the amount of synthetic data in real + synthetic setting is decreased, performance increases, *e.g., AE* performs 83.2% and 46.8%, *AE-L* performs 84.3% and 47.4% on NTU60, and NTU120 with "real + (20%) synthetic data". In this case, *SSVI*-based models perform better than *AE* and *AE-L* (both synthetic & real + synthetic) for all cases, showing that they can handle *viewpoint perturbations* in a better way.

4.11 Fine-tuning Protocol and End-to-end Training

	Fine-tuning Protocol [109]	End-to-end Training
NTU-60 [181] *C-Subject*	69.9 ↔	70.5 ↑
NTU-60 [181] *C-View*	83.7 ↓	83.8 ↓
NTU-120 [121] *C-Subject*	57.1 ↓	57.5 ↓
NTU-120 [121] *C-Setup*	59.6 ↓	61.1 ↓
Skeletics-152 [75]	45.5 ↓	54.3 ↑
DMCD [52]	97.2 ↑	97.4 ↑
Emilya [65]	80.7 ↓	76.9 ↓

Table 4.10 Performance of the proposed method when the fine-tuning protocol and the end-to-end training are applied. All the scores are in terms of accuracy (%) except the F1-scores (%) given for the DMCD dataset [52]. ↑ ↓ and ↔ stand for the performance improvement, decrease, and no-change, respectively *w.r.t. LEP* results obtained for the proposed method.

The performances of *AE* with the fine-tuning protocol [109] and end-to-end supervised training are reported in Table 4.10. Both experiments were applied for 100 epochs with a learning rate of 0.001.

- **Fine-tuning Protocol [109]:** This refers to first end-to-end pre-training of *AE-L* in an unsupervised way. Then appending a linear classifier to the encoder and fine-tuning the whole model for the target task. Therefore, this protocol is *supervised*.
- **End-to-end Training:** This refers to *fully supervised* learning of *AE-L* from scratch using the class labels of the training data.

While fine-tuning performs the best out of all, fine-tuning and supervised HAR results are always better than "*AE* Unsupervised" (as expected) by +3∼18% for NTU-60 [181] and +15∼17% for NTU-120 [121]. As the Laplacian regularizer is trained on the reconstructed skeleton from the *decoder*, and the experiments presented herein are regarding applying a linear classifier appended to the *encoder*, the results of *AE-L* are the same as *AE*. Notice that, when applying *LEP* and *1-NN* evaluation protocols, the encoder is frozen (*i.e.,* the encoder is *detached*), and the feature learning is *unsupervised*. Besides, *1-NN* does not learn any classifier but relies only on a distance metric.

	Data Percentage			
	1%	25%	50%	75%
NTU-60 [181] *C-Subject*	32.3	60.9	65.0	66.4
NTU-60 [181] *C-View*	26.5	70.5	76.0	78.6
NTU-120 [121] *C-Subject*	16.1	47.3	50.7	51.6
NTU-120 [121] *C-Setup*	18.9	49.7	53.7	55.1
Skeletics-152 [75]	8.8	25.0	31.6	36.5
DMCD [52]	23.0	71.7	81.3	86.5
Emilya [65]	26.4	61.5	68.7	71.9

Table 4.11 Performance of the proposed method when the Linear Evaluation Protocol is applied with fewer labels. All the scores are in terms of accuracy (%) except the F1-scores (%) given for the DMCD dataset [52].

On the other hand, the encoder is *not frozen* in the application of the fine-tuning protocol and the end-to-end training, *i.e.,* it is *learnable* and, the proposed *AE-L* is no longer unsupervised. These evaluation protocols align with the recent SOTA, *e.g.,* [109] to show that *AE-L* is flexible to adjust between supervised and unsupervised settings.

It is important to highlight that for this set of experiments, the training procedure was not optimised (*e.g.,* by adjusting the hyper-parameters of AE-L). Instead, all implementation settings are kept as it was used in unsupervised training to supply a direct comparison with *LEP*. In some cases, the fine-tuning and end-to-end protocol results are lower than the *w.r.t. LEP* performance (*e.g.,* NTU-60 C-View [181], and Emilya [65], while still achieving better scores than several supervised SOTA). As a concluding remark, it is undoubtedly correct stating that these results can be improved by performing a hyper-parameter search on the validation sets.

4.12 Linear Evaluation Protocol with Fewer Training Data

To better examine the learning capability of *AE-L*, the first step was to train them in an unsupervised way (as described in Chapter 4) with all training data. During inference, *LEP* evaluation protocol was adopted, but the linear classifier is trained with only 1%, 25%, 50%, and 75% randomly selected data while keeping the class balance the same as the

original datasets. Also, the hyper-parameter search was not performed for these experiments, keeping all settings as in Chapter 4.

The results in Table 4.11 show that, for all cases, when the percentage of the training data is increased, the performance of the proposed method also improves. Moreover, *AE-L* is able to surpass several SOTA when it is trained on much fewer data (*e.g.,* 25%, 50%) compared to the amount of the data SOTA is trained on (*i.e.,* 100%).

In detail,

- **NTU-60 [181] *C-Subject*.** By using 25% of the data, *AE-L* is able to achieve better results compared to: Lie Group [170], Cavazza *et al.* [22], H-RNN [56], P&C [194], LongT GAN [247], MS^{2}L [117], PCRP [225], VAE-PoseRNN [100] and AS-CAL [173], whose are trained with 100% of the data.
- **NTU-60 [181] *C-View*.** When using 50% of the training data, *AE-L* surpasses the performance of Lie Group [170], Cavazza *et al.* [22], H-RNN [56], LongT GAN [247], MS^{2}L [117], PCRP [225], VAE-PoseRNN [100], AS-CAL [173] and MM-AE [80] trained with the whole training data.
- **NTU-120 [121] *C-Subject*.** By training *AE-L* with the 50% of the training data, better results were achieved compared to Part-Aware LSTM [181], Soft RNN [84], P&C [194], PCRP [225] and AS-CAL [173] trained by using 100% of the data.
- **NTU-120 [121] *C-Setup*.** By using 50% of the training data, *AE-L* is able to achieve better results compared to Part-Aware LSTM [181], Soft RNN [84], P&C† [194], PCRP [225] and AS-CAL [173] whose model are learnt with the whole training data.
- **Skeletics-152 [75].** By being trained with the 50% of the training data, *AE-L* surpasses the methods: MS^{2}L [117], PCRP [225], AS-CAL [173] and LongT GAN [247], all trained with the 100% of the data.
- **DMCD [52].** *AE-L* trained on 50% of the training data, achieves better performance compared to Beyan *et al.* [8], P&C [194], MS^{2}L [117], PCRP [225], AS-CAL [173], and LongT GAN [247] trained with the whole training data.
- **Emilya [65]**. By using 50% of the training data, *AE-L* surpasses the methods: P&C FS* [194], MS^{2}L [117], PCRP [225] and AS-CAL [173] trained on whole dataset.

4.13 Comparisons Against Supervised Methods

This section compares the performance of *AE-L* with SOTA supervised skeleton-based HAR approaches on NTU-60 dataset [181]. This comparison includes kernel-based methods [170, 22] and the methods realising feature learning [56, 122, 181, 241, 193, 124, 107, 227, 220, 110, 184, 188, 183, 28] with several different deep learning architectures, *e.g.,* RNNs, LSTMs, CNNs, and Graph Convolutional Networks (GCNs). The corresponding results are presented in Figure 4.8, while an in-depth comparison is given in Table 4.12. Performance comparisons between *AE-L* and the state-of-the-art unsupervised and supervised skeleton-based HAR methods on NTU-60 dataset [181] are given in Figure 4.8. The results in Table 4.12 provide the quantitative values, summarised in Figure 4.8.

AE-L outperforms all prior unsupervised skeleton-based approaches on the Cross-Subject and Cross-View settings. Importantly, the learnable representation, although unsupervised, allows the proposed method even to surpass a few supervised skeleton-based action recognition methods: [170, 22, 56, 122, 181, 193].

AE-L, although based on unsupervised learning, can achieve better performance than the fully supervised kernel-based methods [170, 22], with a +7.2% to +19.8% improvement in C-Subject and a +22% to +32.6% improvement in C-View setting. *AE-L* also outperforms several fully supervised deep architectural methods: hierarchical RNN [56] (providing an increase of 10.8% in C-Subject and up to 21.4% in C-View), spatial-temporal LSTM [122] (resulting in a boost of +0.7% in C-Subject and up to +7.7% in C-View) and part-aware LSTM [181] (achieving an improvement of +7% in C-Subject and up to +15.1% in C-View) while performing better than temporal CNN [193] (up to +2.3%) in C-View setting. These results show that the proposed unsupervised residual convolutions with Laplacian regularization exceed even supervised GRUs, RNNs, and LSTMs (and variants) for HAR. Besides the favourable results of *AE-L*, it is important to note that fully supervised techniques [241, 124, 107, 227, 220, 110, 184, 188, 183, 28] perform better than *AE-L*. These methods mostly implement GCNs [227, 220, 110, 184, 188, 28], and some of them additionally adapt LSTMs [188] or a variable temporal dense block [220]. The best performing method is [28] with 90.7% and 96.5% in C-Subject and C-View, respectively.

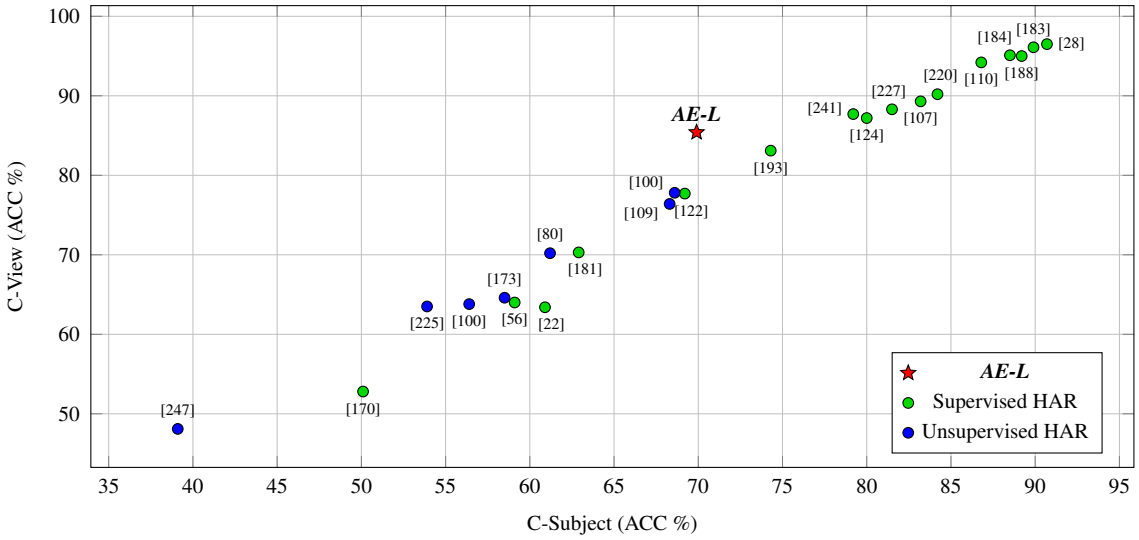


Figure 4.8 Comparisons between *AE-L* and SOTA unsupervised and supervised skeleton-based HAR methods on NTU-60 dataset [181].

	Year	Method	Classifier	Architecture	C-Subject	C-View
Rahmani *et al.* [170]	2016	supervised	*linear SVM*	*3D Spatio-temporal interest points*	50.1	52.8
Cavazza *et al.* [22]	2019	supervised	*linear SVM*	*Kernel-approximating random feat maps*	60.9	63.4
Du *et al.* [56]	2015	supervised	*softmax*	*Hierarchical RNN*	59.1	64.0
Liu *et al.* [122]	2016	supervised	*softmax*	*Spatial Temporal LSTM*	69.2	77.7
Shahroudy *et al.* [181]	2016	supervised	*softmax*	*Part-Aware LSTM*	62.9	70.3
Kim *et al.* [193]	2017	supervised	*softmax*	*Temporal CNN*	74.3	83.1
Zhang *et al.* [241]	2017	supervised	*softmax*	*View-Adaptive LSTM*	79.2	87.7
Liu *et al.* [124]	2017	supervised	*softmax*	*Multi-stream CNN*	80.0	87.2
Liu *et al.* [55]	2017	supervised	*softmax*	*CNN*	83.2	89.3
Yan *et al.* [227]	2018	supervised	*softmax*	*Spatio-Temporal GCN*	81.5	88.3
Wen *et al.* [220]	2019	supervised	*softmax*	*Motif GCN + Variable Temporal Dense Block*	84.2	90.2
Li *et al.* [110]	2019	supervised	*softmax*	*Actional-structural GCN*	86.8	94.2
Shi *et al.* [184]	2019	supervised	*softmax*	*2-stream Adaptive GCN*	88.5	95.1
Si *et al.* [188]	2019	supervised	*softmax*	*Attention GCN+LSTM*	89.2	95.0
Shi *et al.* [183]	2019	supervised	*softmax*	*Directed Graph Neural Networks*	89.9	96.1
Cheng *et al.* [28]	2020	supervised	*softmax*	*Shift GCN*	**90.7**	**96.5**
Holden *et al.* [80]	2015	unsupervised	*linear classifier*	*Denoising AE*	61.2	70.2
Zheng *et al.* [247]	2018	unsupervised	*linear classifier*	*Adversarial GRU-AE*	39.1	48.1
Kundu *et al.* [100]	2018	unsupervised	*linear classifier*	*Variational-AE + poseRNN*	56.4	63.8
Kundu *et al.* [100]	2018	unsupervised	*linear classifier*	*Encoder-GAN + poseRNN*	68.6	77.8
Xu *et al.* [225]	2020	unsupervised	*linear classifier*	*Contrastive-AE*	53.9	63.5
Rao *et al.* [173]	2020	unsupervised	*linear classifier*	*Contrastive-AE*	58.5	64.6
Li *et al.* [109]	2021	unsupervised	*linear classifier*	*Contrastive-GCN*	68.3	76.4
AE-L	2021	unsupervised	*linear classifier*	*Regularised convolutional, residual AE*	**69.9**	**85.4**

Table 4.12 Performance comparisons between *AE-L* and the state-of-the-art supervised and unsupervised skeleton-based HAR methods on NTU-60 dataset [181] in terms of accuracy (%). The results that *AE-L* surpasses are underlined. The best results for the supervised and unsupervised methods are individually shown in **black**.

4.14 Confusion matrices

The confusion matrices for testing *AE-L* performance within *1-NN* protocol [194] for datasets NTU-60 [181] (Cross-Subject, Cross-View) are given in Figure 4.9 and Figure 4.10, respectively. In the same figure, the accuracy score of each action class was also reported in the box-plot form.

4.14.1 Accuracy-per-action class comparison

AE-L achieves recognition accuracy above 80% for 8 actions (*sitting down*, *standing up from sitting position*, *wearing jacket*, *taking off jacket*, *jumping up*, *falling*, *walking towards each other*, and *walking apart from each other*) in NTU-60 Cross-Subject setting [181].

In NTU-60 Cross-View setting [181], *AE-L* performs recognition above 90% accuracy for 13 actions (*throwing*, *sitting down*, *standing up from sitting position*, *wearing jacket*, *taking off jacket*, *cheering up*, *kicking something*, *one foot jumping*, *jumping up*, *salute*, *crossing hands in front*, *staggering*, and *falling*) while class accuracy above 80% is observed for 41 actions.

There are 3 actions: *standing up from sitting position*, *jumping*, and *falling* for which *AE-L* recognises with nearly 100% accuracy in Cross-View setting of NTU-60 [181].

4.14.2 Accuracy improvements of *AE-L* on C-Subject protocol

AE-L improves the performance of the *AE* model, showing that Laplacian regularization supplies some advantages.

For NTU-60 Cross-Subject action classes: *brushing hair, drop, reading, wear on glasses, take off glasses and using a fan* and for NTU-120 Cross-Subject action classes; *taking off a shoe, wearing on glasses, making a phone call, putting the palms together, patting on back of other person, applying cream on face and kicking backward*; obtaining at least +5% performance gain by involving Laplacian regularization to the proposed *AE*.

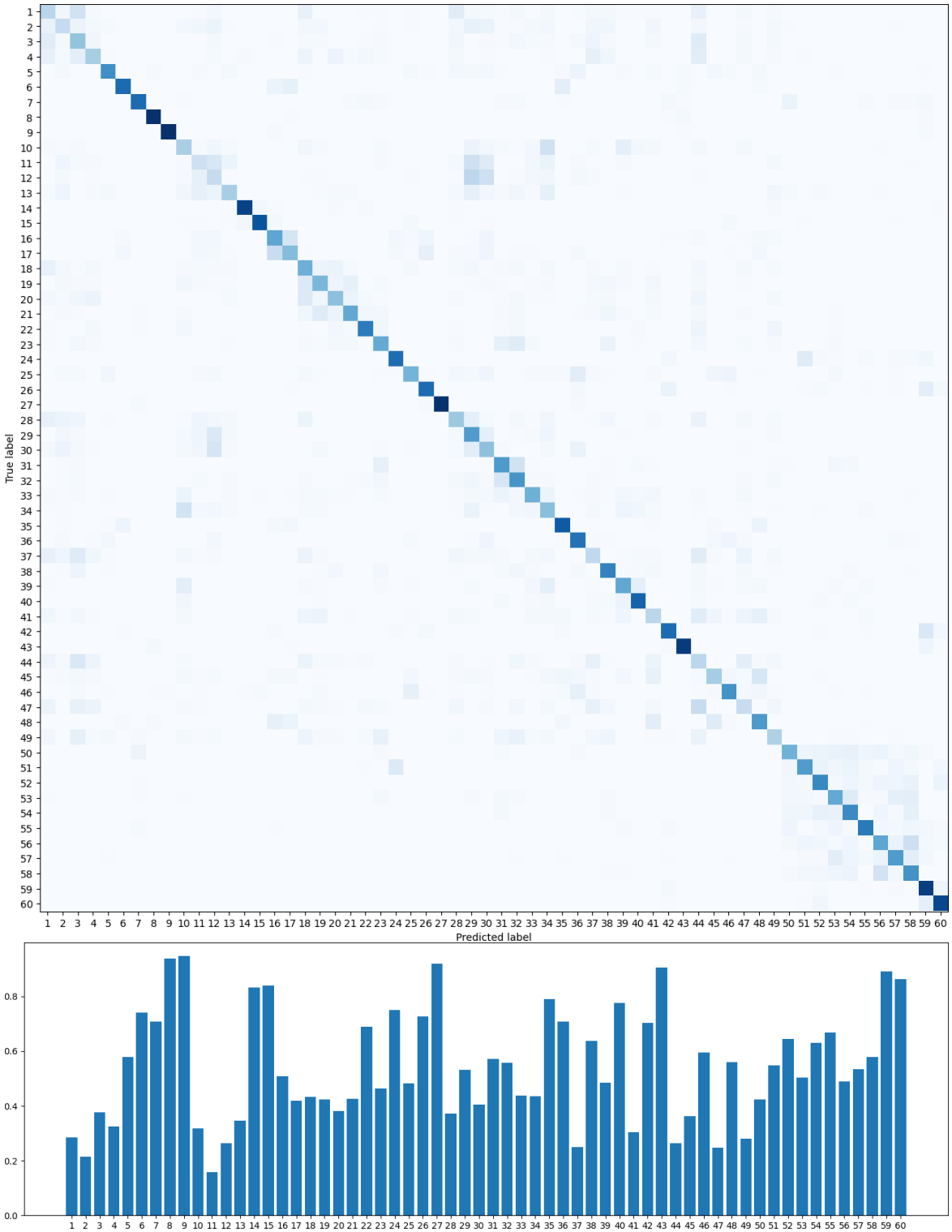


Figure 4.9 Confusion matrices and the corresponding accuracy scores for each action class obtained when *AE-L* is applied with *1-NN* protocol on the NTU-60 [181] C-Subject dataset.

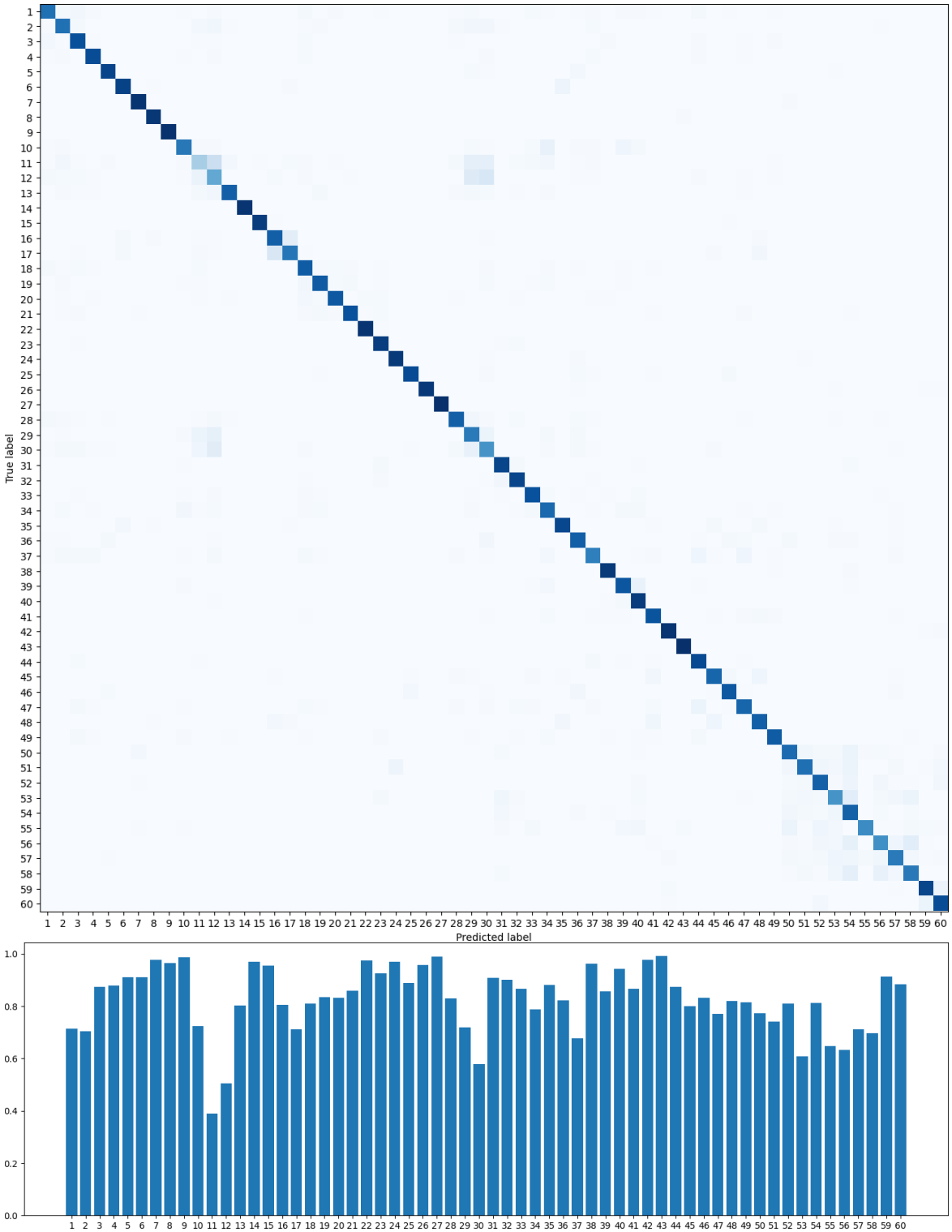


Figure 4.10 Confusion matrices and the corresponding accuracy scores for each action class obtained when *AE-L* is applied with *1-NN* protocol on the NTU-60 [181] C-View dataset.

4.14.3 Accuracy improvements of *AE-L* on C-View and C-Setup protocols

Additionally, *AE-L* performs at least +5% better than *AE* for NTU-60 Cross-View and NTU-120 Cross-Setup actions.

The NTU60 Cross-View actions are *eating meal, brushing teeth, brushing hair, dropping, clapping, reading, tearing up paper, wearing on glasses, taking off glasses, putting on a hat, taking off a hat, reaching into pocket, hopping, make a phone call, playing with phone, taking a selfie, checking time, rubbing two hands together, wiping face, putting the palms together, sneeze/cough, touching head/chest/back, using a fan, punching other person, patting on back of other person* and *touching other person's pocket*.

In addition, the NTU-120 Cross-Setup action classes are: *drinking water, eating meal, putting on a hat, taking off a hat, kicking something, making a phone call, putting the palms together, kicking other person, hushing, thumbing up, making victory sign, sniffing, balling up paper, applying cream on face, taking something out of a bag* and *crossing arms*.

4.15 Visualization of the reconstructed skeletons

Figures 4.11 and 4.12 present the visualisations of the reconstructed skeletons obtained by applying the proposed models (*AE* and *AE-L*). *Blue* skeletons represent the input data (of action "Drink Water" for Figure 4.11 and "Standing Up" for Figure 4.12), *red* and *green* skeletons are reconstructed by *AE* and *AE-L*, respectively. In these examples, while the effectiveness of the models is the same, in other words, *AE* and *AE-L* both classify the actions correctly, the *AE-L* makes the reconstructed skeletons smoother compared to *AE*.

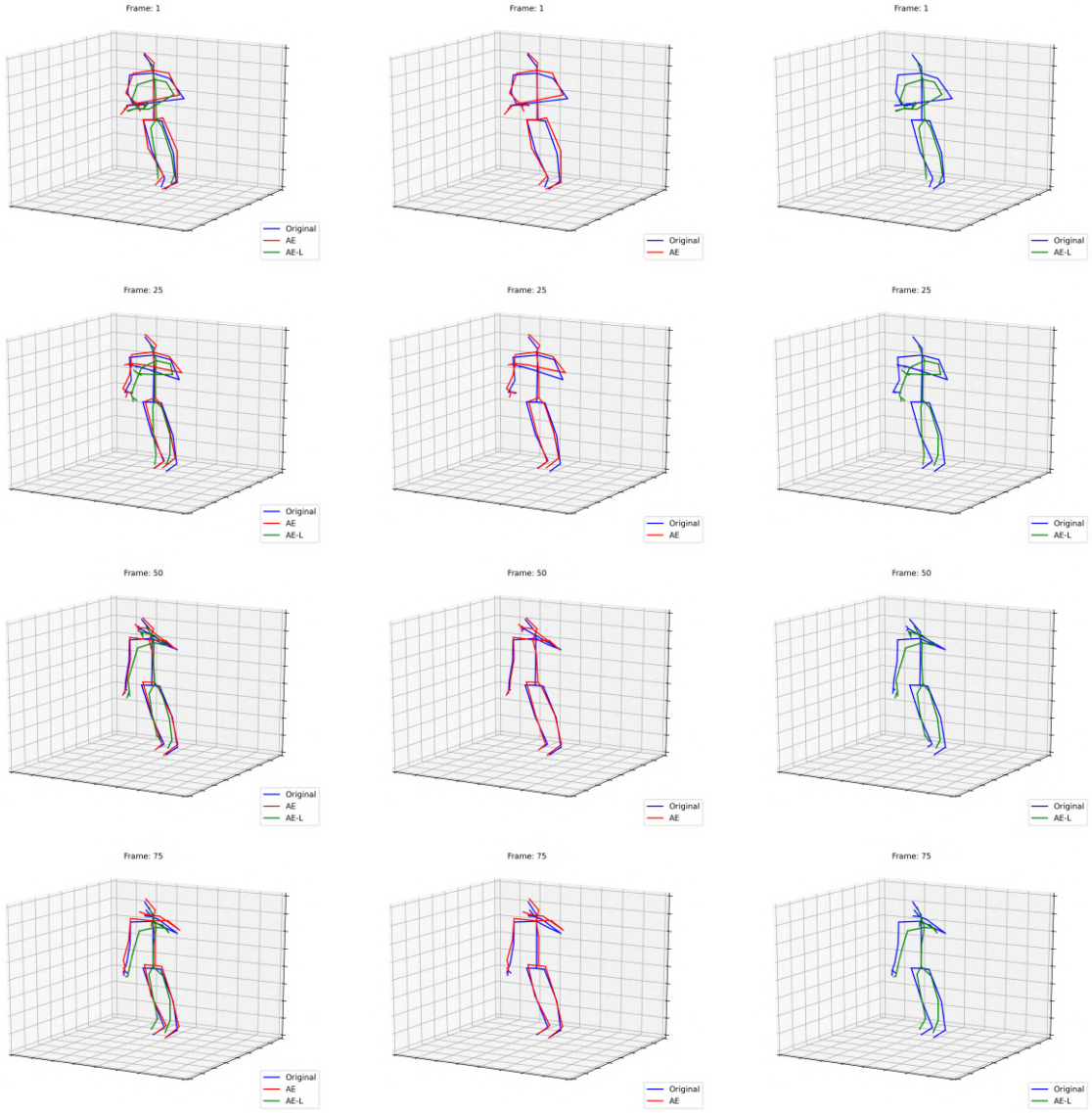


Figure 4.11 Action class "Drink Water" in NTU-60 [181] Cross-View dataset. ***Blue:*** original data, ***Red:*** *AE* reconstruction, ***Green:*** *AE-L* reconstruction. Rows correspond to different time-frames. Both *AE* and *AE-L* correctly classify this action sample.

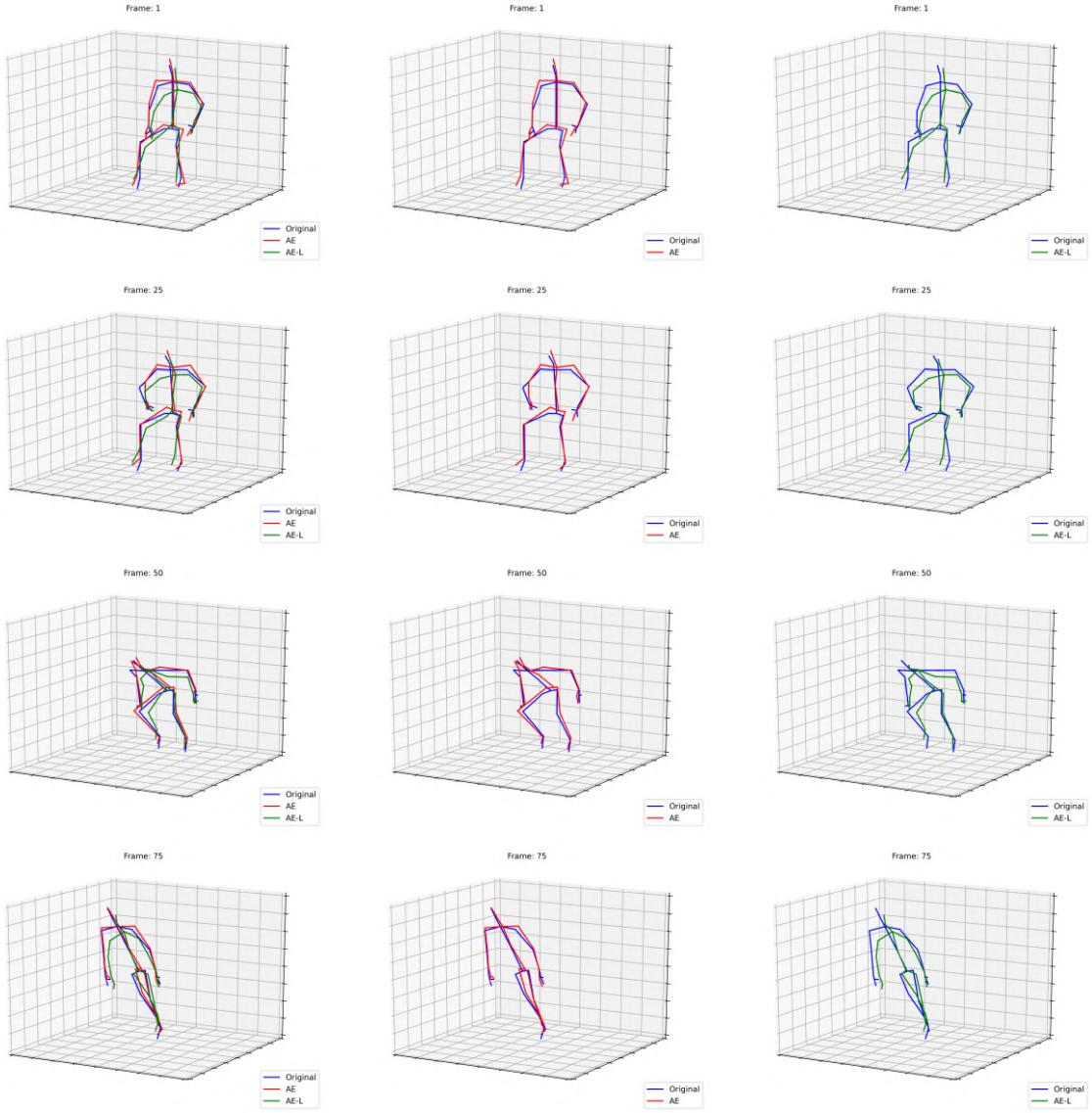


Figure 4.12 Action class "Standing Up" in NTU-60 [181] Cross-View dataset. ***Blue:*** original data, ***Red:*** *AE* reconstruction, ***Green:*** *AE-L* reconstruction. Rows correspond to different time-frames. Both *AE* and *AE-L* correctly classify this action sample.

4.16 Qualitative Results

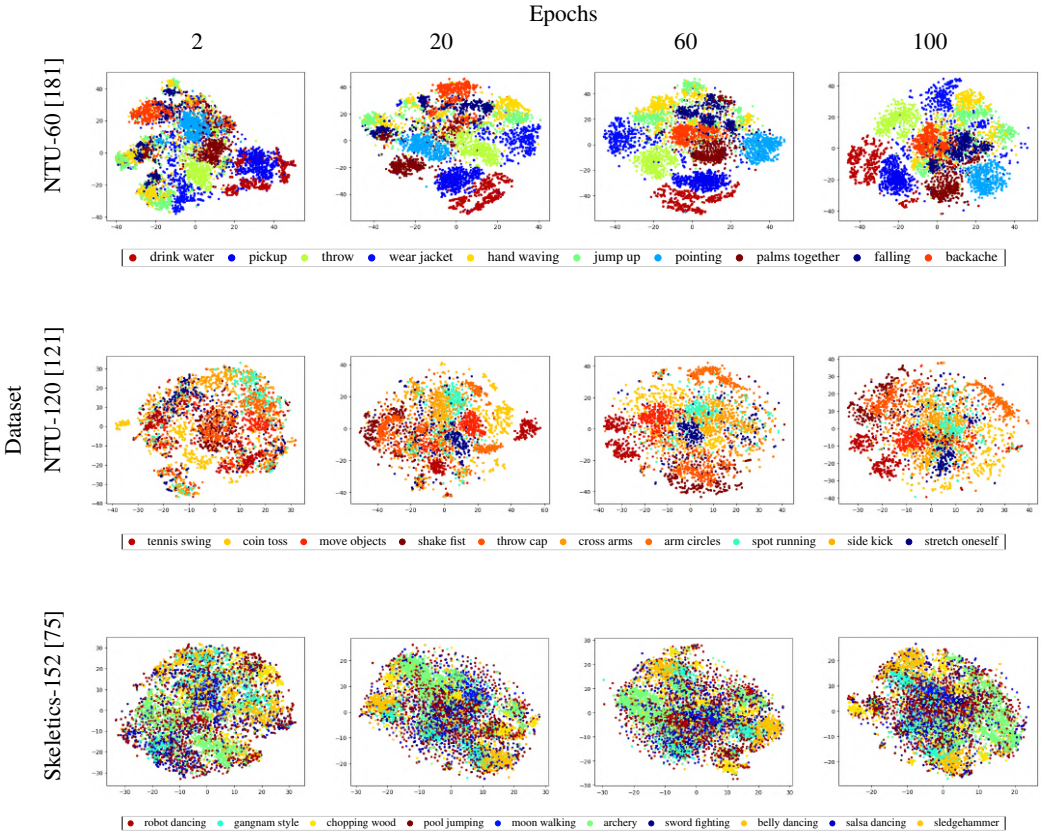


Figure 4.13 The t-SNE visualization of embeddings at different epochs when training *AE-L*. Embeddings of random 10 categories are sampled and visualised with different colours. Illustrations refer to epochs 2, 20, 60, and 100, respectively.

Figure 4.13 shows the embeddings of *AE-L* learned during unsupervised training of it by using t-SNE [206] for the epochs 2, 20, 60, and 100. For NTU-60 [181], NTU-120 [121], and Skeletics-152 [75] datasets, 10 action classes were randomly selected. For NTU-60 [181], these are: “drink water”, “pickup”, “throw”, “wear jacket”, “hand waving”, “jump up”, “pointing”, “palms together”, “falling”, and “backache”. The actions for NTU-120 [181] are: “tennis swing”, “coin toss”, “move objects”, “shake fist”, “throw cap”, “cross arms”, “arm circles”, “spot running”, “side kick”, and “stretch oneself”. For Skeletics-152 [75] the selected actions are: “robot dancing”, “gangnam style”, “chopping wood”, “pool jumping”, “moon walking”, “archery”, “sword fighting”, “belly dancing”, “salsa dancing”, and “sledgehammer”. Embeddings of *AE-L* are more clustered in NTU-60 compared to NTU-120 [181] and Skeletics-152 [75] dataset. This is in line with the quantitative results of

AE-L in, which performs numerically better in NTU-60. On the other hand, one can observe more compact and less overlapping clusters after the epoch of 20 for all datasets.

4.17 Transfer-ability

Pre-trained on \ *Tested on*	NTU-60 [181] *C-Subject*	NTU-60 [181] *C-View*	NTU-120 [121] *C-Subject*	NTU-120 [121] *C-Setup*	NTU-61∼120 [121] *C-Subject*	NTU-61∼120 [121] *C-Setup*	Skeletics-152 [75]	DMCD [52]	Emilya [65]
NTU-60 [181] *C-Subject*	54.1	82.2	42.1	46.4	46.6	43.4	48.9	75.1	76.4
NTU-60 [181] *C-View*	54.6	83.1	42.0	46.2	45.8	45.1	49.1	92.7	75.1
NTU-120 [121] *C-Subject*	52.0	81.0	42.4	44.3	44.0	45.2	48.0	92.6	74.7
NTU-120 [121] *C-Setup*	52.3	81.2	38.9	44.7	44.0	45.4	47.9	92.7	74.7
NTU-61∼120 [121] *C-Subject*	55.6	52.1	39.4	43.8	45.1	46.4	48.9	92.7	76.4
NTU-61∼120 [121] *C-Setup*	54.3	53.3	39.9	44.4	45.2	46.1	48.1	92.6	75.1
Skeletics-152 [75]	47.8	71.3	35.4	39.1	42.2	44.0	49.0	92.7	74.7
DMCD [52]	51.3	70.4	39.1	44.5	44.9	45.2	47.6	96.4	75.0
Emilya [65]	48.3	70.8	38.5	43.7	44.6	45.0	47.1	82.7	75.2

Table 4.13 The transfer-ability of *AE-L* across different datasets. Unsupervised pre-training is performed *w.r.t.* each dataset's training/testing split (except DMCD and Emilya, in which cross-validation is applied as in [8]). NTU 61∼120 refers to using only the action classes from 61 to 120. The darker colour shows better performance compared to a lighter colour in the same column.

This section tests the transfer-ability of *AE-L* across different datasets. The unsupervised pre-training is considered to be useful in a practical scenario in which (in the case presented in this chapter) action and/or emotion classes are varying, and labelling new data is expensive. Herein, the transfer-ability of the proposed model was tested across different datasets, when *a)* in the unsupervised training and inference of the same task but a different set of classes exist (*e.g.,* pre-training on *action* dataset NTU-60 [181] *C-Subject* → transfer learning on *action* dataset NTU-120 [121] *C-Setup*) and *b)* different tasks during unsupervised training and inference are being addressed (*e.g.,* pre-training on *action* dataset Skeletics-152 [75] → transfer learning on *emotion* dataset Emilya [65]). The corresponding results are given in Table 4.13 in terms of *1-NN*protocol. Overall, a drop in performance can be expected due to the domain gap between datasets (*e.g.,* variety in actions and emotions). Still, results show the effectiveness of the proposed approach in dampening this phenomenon. In many cases, the performance even surpasses their same-dataset baseline. For example, in case of actions → actions, a boost in performance can be observed when NTU-60 [181] *C-Subject* is tested with a model pre-trained with NTU-61∼120 [121] *C-Subject* and NTU-60 [181] *C-View* (+1.5% and +0.5%); NTU-120 [121] *C-Setup* is tested with a model pre-trained on NTU-60 [181] *C-Subject* and NTU-60 [181] *C-View* (+1.7% and +1.5%); and NTU-61∼120 [121] *C-Subject* is classified by a model pre-trained on NTU-60 [181] *C-Subject* and NTU-60 [181] *C-View* (+1.5% and +0.7%). On the other hand, for actions → emotions, there are performance improvements (up to +1.2%) when Emilya dataset [65] is recognised by a model pre-trained on NTU-60 [181] *C-Subject* or NTU-61∼120 [121] *C-Subject*.

4.18 Concluding Remarks

This chapter introduced a novel unsupervised feature learning method that results in effective feature representations of actions and emotions from the input 3D skeleton sequences, where all these findings were ultimately published in [154, 152]. The proposed method is based on convolutional autoencoders (*AE*) and adapting Laplacian Regularisation (L) to capture the pose geometry in time. *AE-L* is validated on large-scale HAR benchmarks that exceed all SOTA skeleton-based U-HAR methods for Cross-Subject, Cross-View, and Cross-Setup settings. It is also validated on large-scale HER benchmarks in supervised and unsupervised settings, showing exciting and promising results. This proves that the proposed *AE-L* is able to learn more distinctive action and emotion features compared to the prior art. *AE-L* were also updated with gradient reversing (*GRAE-L*) to provide better invariance to camera viewpoint changes compared to a direct competitor [142]. Overall, this study highlights the potential of unsupervised learning for 3D skeleton-based action and emotion recognition and serves as a valuable contribution to the improvement of the research field. Demonstrating the capabilities of unsupervised approach *w.r.t.* supervised methods, further research is encouraged to explore and improve the proposed framework, evaluating its performance on more extensive and diverse datasets.

Chapter 5

SKELTER: Unsupervised Skeleton Action Denoising and Recognition using Transformers

In Chapter 4 the proposed model *AE-L*, which leverages large-scale datasets to solve the challenging problem for U-HAR and to overcome limitations of pure unsupervised approaches (*i.e.,* Subspace Clustering, presented in Chapter 3). As most of the approaches are dedicated to reaching the best recognition accuracies, no attention has been put into analysing the resilience of such methods given perturbed data, a likely occurrence in real in-the-wild testing scenarios.

The benchmark datasets, on which the existing U-HAR methods are tested (check Chapter 2 Section 2.4 for a detailed description), were recorded using depth sensors [215] (*e.g.,* by using Microsoft Kinect) in relatively controlled experimental settings, being free from several challenges such as noisy data, severe occlusions, *etc.*, thus being far from realistic scenarios. It is also important to notice that in real-world conditions, there can be errors in sensors resulting in missing frames and/or errors occurring due to the misdetection of the pose estimators.

Therefore, this chapter provides a systematic analysis of the state-of-the-art (SOTA) skeleton-based U-HAR methods evaluated on perturbed and altered data, simulating several real-world challenges, *e.g.,* noise, clutter, occlusions and geometrical distortions. To do so, an extensive set of perturbations and alterations are presented to simulate in-the-wild scenarios for HAR (*e.g.,* obtained by removing some skeletal joints, rotating the entire pose, injecting geometrical

aberrations, *etc.*, see Section 5.2) and verifying the decrease in performance of current SOTA, evidencing cases where such loss is more predominant. [78, 235] tackled similar approaches but for different tasks (*i.e.,* not U-HAR) and the type of data were images and videos (*i.e.,* not skeletons).

Then, this chapter proposes a novel framework called SKELTER (SKELeton TransformER), which is based on a transformer encoder-decoder capable of learning robust representations from the spatio-temporal 3D-skeletal data (receiving inputs as 3D-skeletal data over time) in an unsupervised fashion (Section 5.3), and showing remarkable denoising capabilities to counter such perturbations effectively. The success of transformers mainly relies on their property to establish long-range connections among time-series data, *w.r.t.* shorter connections as could occur in RNNs or LSTMs. The choice of a transformer-based encoder-decoder architecture is due to its superior ability to encode skeletal joint information across the entire temporal span. At the same time, its attention modules provide context for any position in the input sequence of sequential data, weighing their influence on different temporal parts.

Since their inception in NLP research [207, 16], transformers have gained popularity in different tasks such as for machine translation [234, 219], visual question answering [130, 195], action recognition [7, 71], and human pose estimation [246] to name a few. Vision Transformer (ViT) [54] is the first pure-transformer model deployed for image classification that was trained on large-scale datasets like Imagenet-21K [175] and JFT-300M achieving remarkable results. On the other hand, ACTOR [160] is a transformer-based conditional VAE, which can generate action-conditioned human motions by sampling from a sequence-level latent vector. The hierarchical transformer from *Cheng et al.* [30] fuses part-based skeletal features to higher-level representations, using self-attention mechanisms from transformers. Although the common final task of U-HAR, this model formulates the unsupervised representation learning as a classification problem, predicting the motion direction of masked poses. However, it does not aim to perform data denoising while this approach is the first transformer-based solution *specifically designed to tackle data denoising for the U-HAR*.

Moreover, additional losses are presented to have robust representations against rotation variances and to provide temporal motion consistency. Overall, the performance of the proposed method is compared with SOTA skeleton-based U-HAR when tested on perturbed and altered data, which is applied on NTU-60 [181] and NTU-120 [121] datasets' Cross-Subject Cross-View and Cross-Setup splits. SKELTER shows limited drops in performance

when skeleton noise is present in comparison with previous approaches, favouring its use in challenging in-the-wild settings by showing its better denoising capability.

Summarising the main contributions of this chapter:

- For the first time, SOTA skeleton-based U-HAR methods are evaluated on perturbed and altered data, which simulate in-the-wild challenging scenarios. The results shown in this chapter could allow the community better to understand the existing methods' applicability to real-world scenarios.
- SKELTER, a novel method based on transformers, processes the skeletal data within a spatio-temporal pipeline by integrating a multi-attention mechanism. This encoder-decoder structure relies upon mean squared error (MSE), so the feature learning is *fully unsupervised*. Also, two additional losses are devised: one for resulting in more robust representations against rotation variances (Section 5.4), and the other to handle the possible temporal motion consistency by integrating triplet loss (Section 5.5).
- Experimental results show that SKELTER is more resilient than the SOTA skeleton-based U-HAR methods when subject to data perturbations and alterations, showing that it can handle various real-world challenges, *i.e.,* performing better denoising compared to other approaches.

5.1 Application scenarios for Skeleton-based HAR

Concerning skeleton-based HAR experimental pipelines, several steps are involved, which can be summarised into two main components:

1. Obtain 3D keypoints from RGB videos, usually as sequences of image frames using specific equipment or using pose estimator algorithms
2. Deploy a state-of-the-art model capable of correctly classifying the correspondent action

The predominant choice bends on benchmark datasets obtained within *staged* scenarios. For example, NTU-60 [181], and NTU-120 [121] are recorded using depth sensors (*i.e.,* Microsoft Kinect v2) inside a constrained and well-controlled setup to achieve the best quality of data.

On the other hand, these conditions could not always be guaranteed in realistic scenarios. For this less common kind of setup, *e.g.,* a surveillance online video feed, a continuous stream of RGB frames represents the input where pose estimator software infers the initial

2D keypoints from the RGB frames [19] and lifts them into the 3D space [158]. Starting from the nature of the scenario itself (online frame-wise 3D pose estimation), depending on the conditions of the scene itself (*e.g.,* overcrowded frames, bad camera recording quality, errors in camera calibration, missing frames from recording, *etc.*), and accounting abrupt and unforeseen events (like noisy estimation, severe occlusions, misdetected keypoints, *etc.*), the quality of keypoints estimation could be severely affected in this type of scenario. Due to its unpredictable nature, the quantity and variability of these unexpected events, action classification from severely-affected 3D keypoints could represent a challenging task for U-HAR SOTA models, which often overlook the particular conditions of this real-world scenario.

The following section illustrates and presents the design choices made *w.r.t.* the perturbation or alteration of given skeletal poses to prove the goodness of SKELTER as a robust model capable of correctly classifying those actions regardless of their conditions. To prove their coherence *w.r.t.* a practical application, Section 5.11 reports a comparison between the perturbed datasets and a test case of the aforementioned *real-world scenario*.

5.2 Data Perturbation & Alteration for HAR

Existing skeleton-based U-HAR methods were evaluated on commonly-used datasets, *e.g.,* NTU-60 [181], and NTU-120 [121], by applying pre-processing steps (normalisation and camera pre-registration). Although such pre-processing represent undoubtedly a common practice to obtain robust features from the skeletal action sequences, the ingredients to apply it might not always be available in real-world processing as well as the methods trained on optimum conditions (such as without considering the noise, missing joints, *etc.*), might result in poor performance in their unconstrained real-world processing. Since the main scope of this chapter is to evaluate the SOTA and SKELTER in the presence of perturbed and altered data, the first step is, therefore, to define a wide range of *perturbations* (*i.e.,* Gaussian Noise, Joint Outlier, Joint Removal, Limbs Removal, Axis Removal, Shear, and Subtract) and *alterations* (*i.e.,* Rotation, and Reverse Motion). The following sections illustrate this claim: the blue skeletal poses represent the original data, whereas the red poses represent the transformation applied.

5.2.1 Data Perturbation

Gaussian Noise (GN)

Additive Gaussian noise is applied over the joints (with a mean equal to zero and standard deviation equal to 0.05) to simulate noisy positions caused by the pose estimator model.

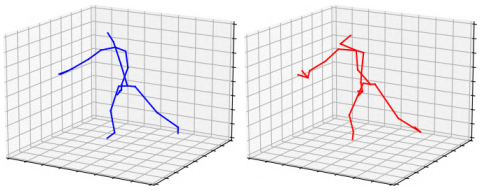


Figure 5.1 *"Gaussian Noise"* perturbation

Joint Outlier (JO)

For each skeletal sample, a random joint is selected and alter its 3D coordinates by adding, for each axis, a fixed value within a range of [-1, 1] to simulate an outlier joint that severe incorrect estimations in the camera feed can cause.

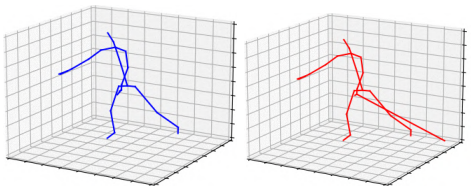


Figure 5.2 *"Joint Outlier"* perturbation

Joint Removal (JR)

For each sample action sequence, a subsection of temporal frames is selected, *i.e.,* a random amount of frames, up to 25% of the entire length, and within these selected frames, a subsection of joints is chosen and set to zero. This random-conditioned selection ensures the simulation of a plausible real-world scenario in which some joints could not be detected.

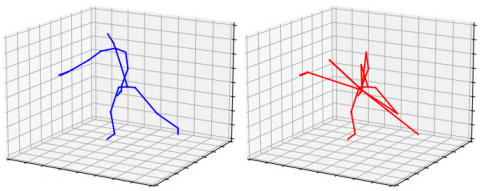


Figure 5.3 *"Joint Removal"* perturbation

Limbs Removal (LR)

For each sample action sequence, the occlusion of an entire limb is simulated by randomly selecting one of the four groups of joints (*i.e.,* left and right arms, left and right legs) and setting their coordinates to zero to simulate *e.g.,* common severe occlusions like "legs occluded due to the subject being sat at a desk".

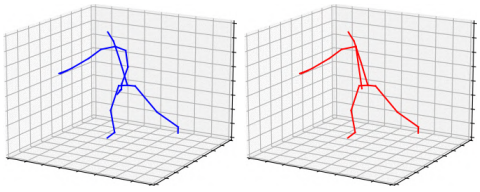


Figure 5.4 *"Limbs Removal"* perturbation

Axis Removal (AR)

Refers to setting an entire axis which is selected randomly to zero. This simulates a failure of a pose estimator to infer 3D poses and as a general-purpose 2D-to-3D hallucination capability of models that are not natively designed for this kind of task.

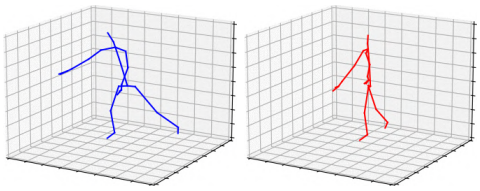


Figure 5.5 *"Axis Removal"* perturbation

Shear (SHR)

Shear simulates the variations in the camera orientation. Each skeletal joint is displaced in a fixed direction (*e.g.,* slant joints with a random angle $S \in [-1,1]$), using a linear mapping matrix:

$$\Omega_s = \begin{bmatrix} 1 & S_X^Y & S_X^Z \\ S_Y^X & 1 & S_Y^Z \\ S_Z^X & S_Z^Y & 1 \end{bmatrix} \tag{5.1}$$

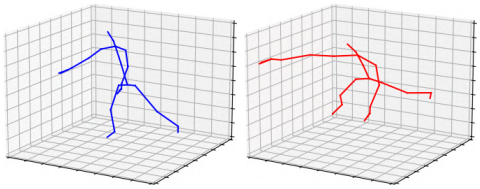


Figure 5.6 *"Shear"* perturbation

Subtract (SUB)

Shift the entire skeleton in 3D space by selecting a random joint and setting it as the new root joint. This is a simulation of the situations arising when *e.g.,* a pose estimator fails to correctly detect a skeletal pose, resulting in an abrupt shift of spatial coordinates.

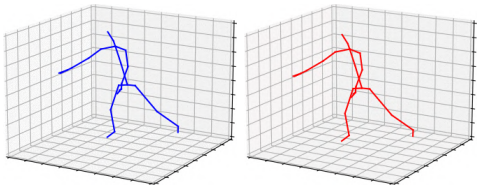


Figure 5.7 *"Subtract"* perturbation

5.2.2 Data Alteration

Rotation (ROT)

3D-skeletal data is rotated along XYZ axes, using the respective rotation matrices given in Equation 5.2. Rotation is involved in testing the strength of a method under view-point variations *e.g.,* in scenarios like camera surveillance where a skeleton pose of a person is captured through multi-camera settings. To simulate plausible contexts, a randomly-sampled Z-axis rotation along all 360 degrees is applied, whereas on X and Y axes, the rotation angles' range spans in-between [-30, 30] degrees.

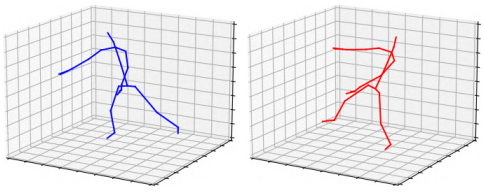


Figure 5.8 *"Rotation"* data alteration

$$\Omega_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \quad \Omega_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \quad \Omega_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

Reversed Motion (RM)

The order of the time frames of a given sample is randomly reversed (with a 50% chance) to ensure a model learns human motion when a reversed perspective is shown. This is useful especially when SKELTER is trained on datasets which contain ambiguous or subtle actions, *e.g.,* actions like *"wear a shoe"* or *"take off a shoe"*, which are theoretically similar but different *w.r.t.* motion execution and action label.

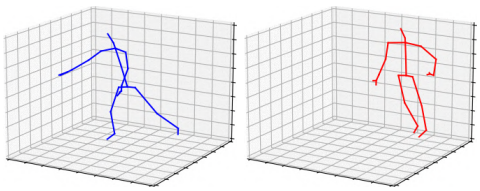


Figure 5.9 *"Reverse Motion"* data alteration

5.3 SKELTER - Model Analysis

The proposed method, SKELTER, was designed by following the general direction endowed by ViT [54] for embedding the input data and ACTOR [160] for the overall encoder/decoder structure. The training paradigm fosters the model to learn robust features for HAR, describing below its components in detail. Following that, additional modules and losses of the method were defined to pursue robustness towards skeletal rotations (Section 5.4) and temporal consistencies (Section 5.5) to disambiguate between specular actions *w.r.t.* time-frames alterations.

5.3.1 Transformer-based Encoder and Decoder

On the proposed frame-wise skeleton encoder, the temporal frames of the given sample represent the input tokens for the transformer module to capture their global dependencies.

The input sequence is defined as $X \in \mathbb{R}^{f\times(3J)}$, where f is the number of time-frames of the action sequence, and J represents the number of joints for each 3D pose.

Skeleton data, which can be (in general) clean or (in this case) perturbed denoted as $\{X_{clean}, X_{pert}\}$ respectively, are fed into the transformer-based encoder and decoder sharing the same architecture (described below). Each 3D-skeletal pose is defined as $X^i_{pert} \in \mathbb{R}^{1\times(3J)},\ i = 1,2,\ldots,f$ of each time-frame f as a *patch token*.

Subsequently, the *patch embedding* $P \in \mathbb{R}^{f\times d}$ is the linear projection of joints into a high-dimensional feature, where d is the embedding dimension, using a trainable linear layer $E \in \mathbb{R}^{(3J)\times d}$:

$$P = [x^1E,\ x^2E,\ \ldots,\ x^fE] + PE \tag{5.3}$$

The *positional embedding* $PE \in \mathbb{R}^{f\times d}$, inherited from [207], come in aid to the transformer module to maintain positional information about the skeletal sequence (*i.e.,* the temporal frame order) as:

$$PE_{(f,2d)} = \sin(f/10000^{2i/d}), \tag{5.4}$$

$$PE_{(f,2d+1)} = \cos(f/10000^{2i/d}). \tag{5.5}$$

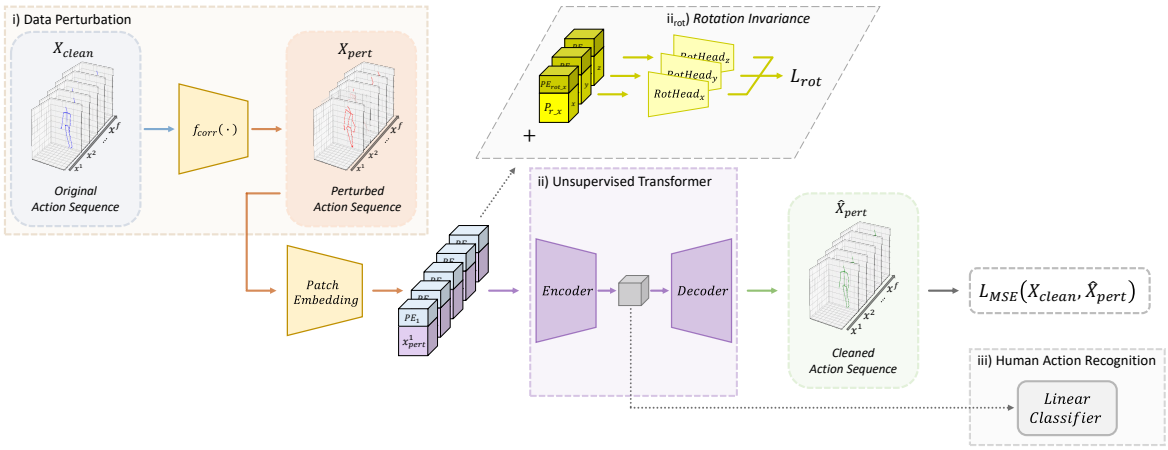


Figure 5.10 Overall Methodology. ***i) Data Perturbation:*** given a clean skeletal action sequence X_{clean} (blue skeleton), a plausible real-world data perturbation is simulated ad applied to the data sample to obtain the input sequence X_{pert} (red skeleton). ***ii) Unsupervised Transformer:*** the proposed approach, SKELTER, is a transformer-based Encoder and Decoder architecture, able to learn how to denoise the X_{pert} data and reconstruct the animated pose as $\hat{X}_{pert}$ (green skeleton), using the reconstruction loss $\mathscr{L}_{\text{MSE}}$. ***ii$_{rot}$) Rotation Invariance***: $RotHead$ are plugged into SKELTER (one for each 3D axis). The rotation loss L_{rot} ensures a correct prediction of the rotation angles, granting invariant properties towards 3D rotations. ***iii) Human Action Recognition (Inference Stage):*** to perform U-HAR, a linear classifier is set on top of the learned feature representations.

5.3.2 Attention in Transformers

The core principle of transformers is the *scaled dot-product attention*, where information coming from different data representations and positions are encoded in a parallel way given by:

$$Attention(Q,K,V) = Softmax(QK^T/\sqrt{d})V, \tag{5.6}$$

where *Attention* is a mapping function using $Q,K,V \in \mathbb{R}^{N\times d}$ (a query, key, and value matrix, respectively). N is the number of sequence vectors, and d represents its dimension which is scaled for normalisation. These matrices are computed from P, by the linear transformations W_Q, W_K, and $W_V \in \mathbb{R}^{d\times d}$ as:

$$Q = PW_Q, \tag{5.7}$$
$$K = PW_K, \tag{5.8}$$
$$V = PW_V. \tag{5.9}$$

5.3.3 Transformer Multiple Self-Attention Heads

To encode attention, multiple h self-attention heads are concatenated together as:

$$MSA(Q,K,V) = Concat(H_1,\ H_2,\ \ldots,\ H_h)W_{out}, \tag{5.10}$$
$$H_i = Attention(Q_i,K_i,V_i),\ i \in [1,\ \ldots,\ h]. \tag{5.11}$$

The general structure of a transformer stack L identical layers given the embedded space $P \in \mathbb{R}^{f\times d}$. Each layer contains a multi-head attention block in conjunction with an MLP layer.

These blocks are placed in-between a Layer Norm $LN(\cdot)$ and a residual connection such that:

$$Y_l' = MSA(LN(Y_{l-1})) + Y_{l-1}, \tag{5.12}$$
$$Y_l = MLP(LN(Y_l)) + Y_l', \tag{5.13}$$
$$Z = LN(Y_l), \tag{5.14}$$

where the transformer output $Z \in \mathbb{R}^{f\times d}$ has the same size of its input $P \in \mathbb{R}^{f\times d}$ and it is averaged in frame dimension to get a vector $\mathbf{z} \in \mathbb{R}^{1\times d}$.

5.3.4 Denoising Property

The transformer-based decoder reconstructs each skeletal action sequence, starting from the unsupervised latent features Z, into $\hat{X}_{pert} \in \mathbb{R}^{f \times (3J)}$. The MSE reconstruction loss ensures the model correctly encodes and rebuilds each data sample free of any noise or corruptions injected during training:

$$\mathscr{L}_{\text{MSE}} = \frac{1}{2}\mathbb{E}_{X \sim \mathscr{B}}\left[\|X_{clean} - \hat{X}_{pert}\|_F^2\right], \tag{5.15}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, *i.e.,* the Euclidean norm of the vector obtained after flattening the tensor. The MSE loss is minimised over mini-batches $\mathscr{B}$.

5.4 SKELTER - Rotation Invariance

Granting the flexibility of the transformer-based approach to combine reconstruction loss with other complementary losses, this section introduces an additional loss to ensure learning consistencies *w.r.t.* rotation invariance. This is visualised in Figure 5.10.

First, each skeletal action sequence was altered by applying ROT (3D rotations, see Section 5.2) to obtain X_{rot}. Following, pseudo labels were defined as y_x, y_y, and y_z, which correspond to the rotation angles applied to the *rotated* action sequence X_{rot}. These pseudo labels are only used for the skeletal rotation prediction task, *but not for U-HAR*.

During training, for each 3D axis, an additional patch token P_r and relative positional embeddings PE_r were stacked (concatenated) on top of the existing ones, thus obtaining:

$$P_{rot} = concat(P_r + P) + concat(PE_r + PE) \tag{5.16}$$

After the encoding stage, the first three vectors were selected from Z (the latent features extracted from P_r) and fed into three different linear layers, representing the axes' rotation heads. The overall goal is to classify the correct rotation angles (as predicted pseudo labels

$\hat{y}_x$, $\hat{y}_y$, and $\hat{y}_z$) using cross entropy losses, defined as:

$$\hat{y}_x = softmax(RotHead_x(Rot(x_{i\_clean}, \alpha))) \tag{5.17}$$

$$\mathscr{L}_{\text{rot\_x}}(\theta) = -\frac{1}{N}\sum_{i}^{N} log\ \hat{y}_x^{\alpha} \tag{5.18}$$

$$\hat{y}_y = softmax(RotHead_y(Rot(x_{i\_clean}, \beta))) \tag{5.19}$$

$$\mathscr{L}_{\text{rot\_y}}(\theta) = -\frac{1}{N}\sum_{i}^{N} log\ \hat{y}_y^{\beta} \tag{5.20}$$

$$\hat{y}_z = softmax(RotHead_z(Rot(x_{i\_clean}, \gamma))) \tag{5.21}$$

$$\mathscr{L}_{\text{rot\_z}}(\theta) = -\frac{1}{N}\sum_{i}^{N} log\ \hat{y}_z^{\gamma}, \tag{5.22}$$

where $Rot(\cdot,\cdot)$ is the rotation function (as shown in Equation 5.2), $RotHead(\cdot)$ is the output of each rotation heads, and θ denotes the encoder parameters. The final loss for this task is:

$$\mathscr{L} = \tfrac{1}{2}\mathbb{E}_{X\sim\mathscr{B}}\left[\|X_{rot} - \hat{X}_{rot}\|_F^2\right] + \mathscr{L}_{\text{rot\_x}} + \mathscr{L}_{\text{rot\_y}} + \mathscr{L}_{\text{rot\_z}}. \tag{5.23}$$

5.5 SKELTER - Temporal Motion Consistency with Triplet Loss

The motion information of a skeletal action sequence can be easily obtained from joints data as it can be represented as the temporal displacement of each joint [109], *i.e.,* $x_{t+1} - x_t$. Herein, the goal of this chapter is to better regularise the model by checking consistencies between the reconstructed skeleton and its data byproduct (*i.e.,* the motion data) using a Triplet Margin Loss [4]:

$$\mathscr{L}_{\text{contr}}(a,p,n) = max\{\|a_i - p_i\|_2 - \|a_i - n_i\|_2 + margin,\ 0\}, \tag{5.24}$$

where a is the anchor samples, joints data coming from X_{pert}, p represents the positive samples obtained from *forward* motion data (unaltered motion data), n is the negative samples consisting of *reversed* motion data and left the default value of 1 for *margin*. This ensures that the latent features learn to reconstruct action samples into the correct temporal motion despite the presence of altered data (RM, see Section 5.2) by attracting the positive samples of the correct motion and pushing afar the inverted motion data which can perturb

the model performance. The final loss for this task is given by:

$$\mathscr{L} = \mathscr{L}_{\text{MSE}} + \mathscr{L}_{\text{contr}}. \tag{5.25}$$

5.6 SKELTER - Implementation Details

Each skeletal action sequence is normalised in terms of bone length in the range of [-1, 1]. As for their temporal sequence length, every missing time-frames were discarded (applying methods introduced in [194]) and regularised the frame numbers to match a fixed size (fixing each sequence length up to 100 time-frames by applying a regularisation in which frames of longer samples were cut or replicate frames for shorter samples).

Both encoder and decoder modules are made of two transformer layers with four attention heads each. Patch embedding and latent space sizes are set to 256. The positional embedding length is set to 100, matching the temporal length of the given action sequences. The model is trained for 100 epochs using AdamW optimiser with a batch size of 64 and a learning rate of 0.001 (with a decay scheduling at epochs 20 and 70).

5.7 Experimental Analysis

The experimental analysis was performed on two large-scale skeletal action datasets: NTU-60 [181] (Chapter 2 Section 2.4.8) and NTU-120 [121] (Chapter 2 Section 2.4.9) using all available data splits, *i.e.,* Cross-Subject, Cross-View and Cross-Setup. For action inference, the common protocol of unsupervised feature learning was used, *i.e.,* linear evaluation [247], such that the latent features (learned without supervision) are given to a linear classifier to perform HAR. Notice that the inference stage is the same with all SOTA competitors.

The performance of the proposed method (SKELTER) was compared against 9 SOTA skeleton-based U-HAR methods: LongTGAN [247], MS2L [117], P&C [194], PCRP [225], AS_CAL [173], AE-L [154], CrosSCLR [109], ISC [200], and AimCLR [74].

Performance Accuracy (ACC %) – Perturbations on *Test set only*										
	NTU-60 [181] C-Subject									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	52.1	4.9	12.9	32.3	10.7	32.7	30.9	29.1	23.0	29.1
MS2L [117]	52.6	16.6	15.2	21.2	15.7	19.8	20.7	34.0	23.9	28.7
P&C FS [194]	50.6	5.9	18.1	37.9	14.4	39.2	35.9	34.9	27.9	22.7
P&C FW [194]	50.7	18.3	14.2	41.7	13.2	42.4	35.2	35.2	29.9	20.8
PCRP [225]	53.9	6.2	12.2	39.6	15.8	40.2	32.7	42.8	28.7	25.2
AS_CAL [173]	58.5	39.7	36.8	46.1	37.9	46.5	41.3	38.9	40.5	18.0
AE-L [154]	69.9	30.3	31.6	65.4	23.0	66.7	59.7	50.1	48.7	21.2
CrosSCLR [109]	**77.8**	51.2	40.1	50.5	40.4	22.4	49.4	57.4	47.8	30.0
ISC [200]	76.3	54.2	50.1	63.8	50.8	63.0	56.9	62.1	58.2	18.1
AimCLR [74]	74.3	55.7	50.0	66.3	58.2	65.0	60.1	63.3	60.5	13.8
SKELTER	69.2	**57.2**	**60.0**	**69.0**	**63.7**	**67.9**	**63.9**	**68.9**	**64.4**	**4.8**

	NTU-60 [181] C-View									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	56.4	8.7	14.6	38.3	11.2	39.2	19.6	12.0	21.8	34.6
MS2L [117]	46.4	10.1	15.9	11.0	14.4	22.2	12.2	30.7	20.5	25.9
P&C FS [194]	76.3	8.5	23.5	60.8	19.7	63.1	50.7	29.5	38.9	37.4
P&C FW [194]	76.1	5.9	15.8	59.1	12.5	61.2	39.2	29.1	34.4	41.7
PCRP [225]	63.5	15.3	14.1	45.5	17.4	46.8	40.7	32.2	32.2	31.3
AS_CAL [173]	64.6	37.7	33.9	46.7	34.7	46.1	35.4	40.2	39.1	25.5
AE-L [154]	**85.4**	11.4	35.4	76.4	24.7	75.5	66.0	58.2	51.6	33.8
CrosSCLR [109]	83.4	58.0	44.6	56.5	53.0	28.6	52.7	57.4	54.1	29.3
ISC [200]	85.2	60.1	49.2	74.0	62.4	72.1	68.8	70.1	66.0	19.2
AimCLR [74]	79.7	60.9	54.9	76.5	65.8	73.8	70.4	74.1	68.8	10.2
SKELTER	78.5	**62.1**	**66.4**	**77.5**	**70.5**	**76.8**	**71.9**	**77.5**	**71.8**	**6.7**

Table 5.1 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when ***only*** the *testing* splits of NTU-60 [181] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The average (AVG) accuracy and the Drop, ↓ *w.r.t.* clean data (CLN), are given (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The best results of each column are given in **bold**.

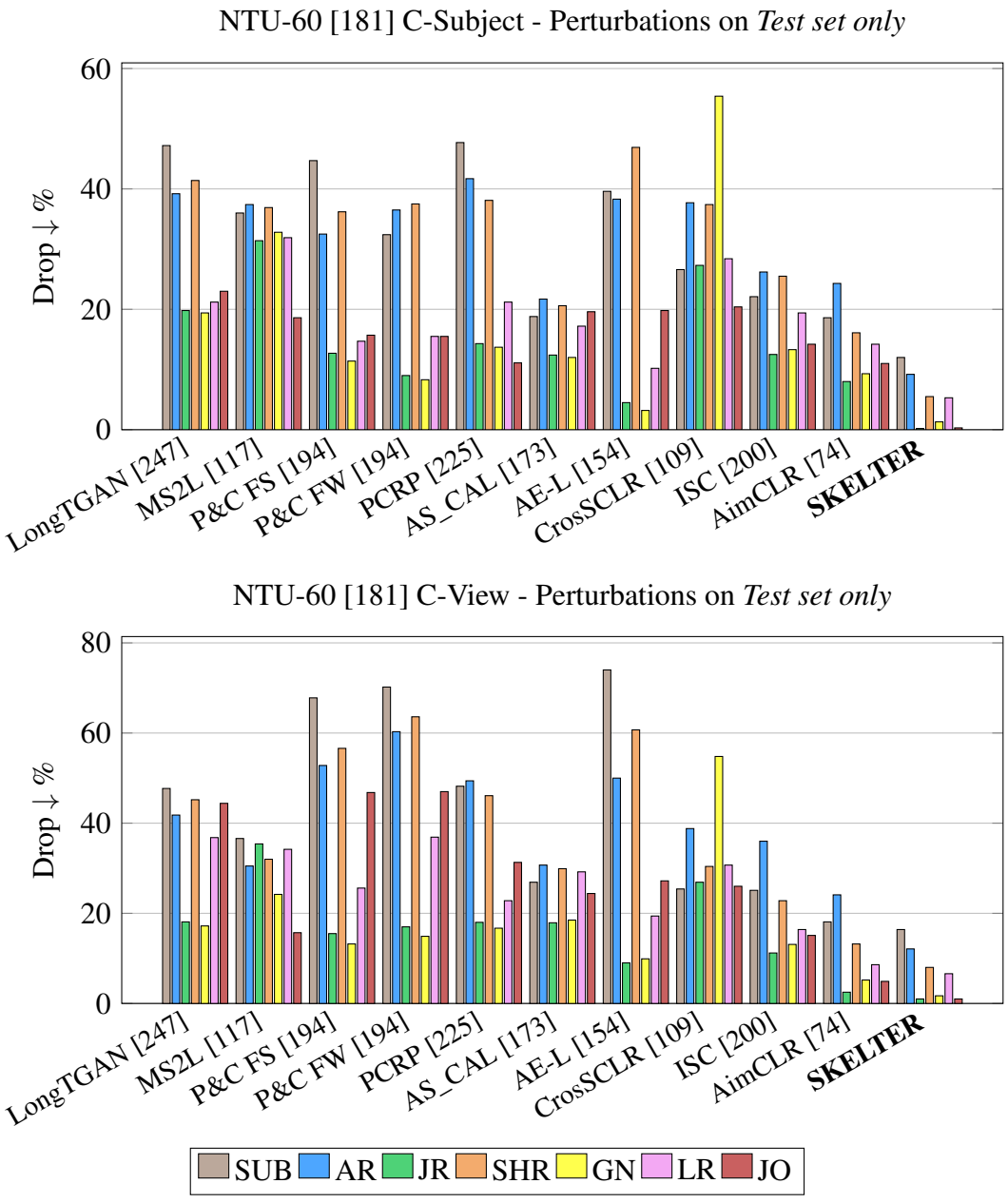


Figure 5.11 Performance drop ↓ % (related to the decrease of accuracy points) of SOTA U-HAR and SKELTER when ***only*** the *testing* splits of NTU-60 [181] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The lowest bars represent the best results.

Performance Accuracy (ACC %) – Perturbations on *Test set only*										
	NTU-120 [121] C-Subject									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	35.6	4.8	7.3	26.7	5.2	26.7	20.8	18.8	17.7	17.9
MS2L [117]	24.3	8.5	10.2	7.7	8.4	9.4	9.1	12.7	10.5	13.8
P&C FS [194]	40.5	2.1	6.0	29.9	6.5	30.2	28.8	24.5	19.8	20.7
P&C FW [194]	40.3	14.3	9.4	30.4	7.6	31.5	28.3	24.2	21,5	18.8
PCRP [225]	41.7	3.7	6.5	27.8	8.5	28.2	22.4	22.8	19.5	22.2
AS_CAL [173]	48.6	27.6	23.9	34.1	25.0	34.7	26.1	30.1	28.3	20.3
AE_L[154]	59.1	7.6	19.3	47.3	11.7	51.3	40.9	47.2	34.9	24.2
CrosSCLR [109]	67.9	40.7	26.4	40.8	35.9	13.5	39.2	47.0	37.7	30.2
ISC [200]	67.1	44.0	37.2	50.8	44.4	52.8	49.3	50.3	47.5	19.6
AimCLR [74]	**68.2**	44.9	42.0	53.2	46.9	54.9	50.1	55.9	50.2	18.0
SKELTER	52.9	**46.5**	**48.7**	**58.2**	**51.7**	**59.1**	**53.9**	**58.9**	**53.9**	**0**

	NTU-120 [121] C-Setup									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	39.7	3.9	8.7	27.2	6.4	28.1	17.4	12.6	16.5	23.2
MS2L [117]	23.8	8.3	10.0	8.4	9.5	8.4	9.9	8.3	10.3	13.5
P&C FS [194]	42.4	12.7	9.7	25.1	7.3	25.1	20.8	22.0	18.4	24.0
P&C FW [194]	42.9	2.1	4.7	32.6	6.9	33.0	23.0	22.1	20.1	22.8
PCRP [225]	45.1	13.5	8.7	30.7	9.7	31.2	27.1	20.7	21.7	23.4
AS_CAL [173]	49.2	26.5	22.6	35.7	23.5	36.2	32.9	35.8	29.9	19.3
AE-L [154]	62.4	7.7	22.6	48.1	12.9	42.8	40.3	32.6	32.9	29.5
CrosSCLR [109]	66.7	41.7	29.0	42.1	36.1	18.0	43.0	50.1	40.8	25.9
ISC [200]	67.9	40.5	34.8	50.8	38.4	43.9	48.1	53.2	46.4	21.5
AimCLR [74]	**68.8**	41.1	37.0	57.1	41.1	44.2	50.9	54.4	48.4	20.4
SKELTER	56.0	**42.9**	**40.6**	**60.9**	**44.1**	**45.7**	**56.5**	**60.5**	**50.2**	**5.8**

Table 5.2 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when ***only*** the *testing* splits of NTU-120 [121] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The average (AVG) accuracy and the Drop, ↓ *w.r.t.* clean data (CLN), are given (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The best results of each column are given in **bold**.

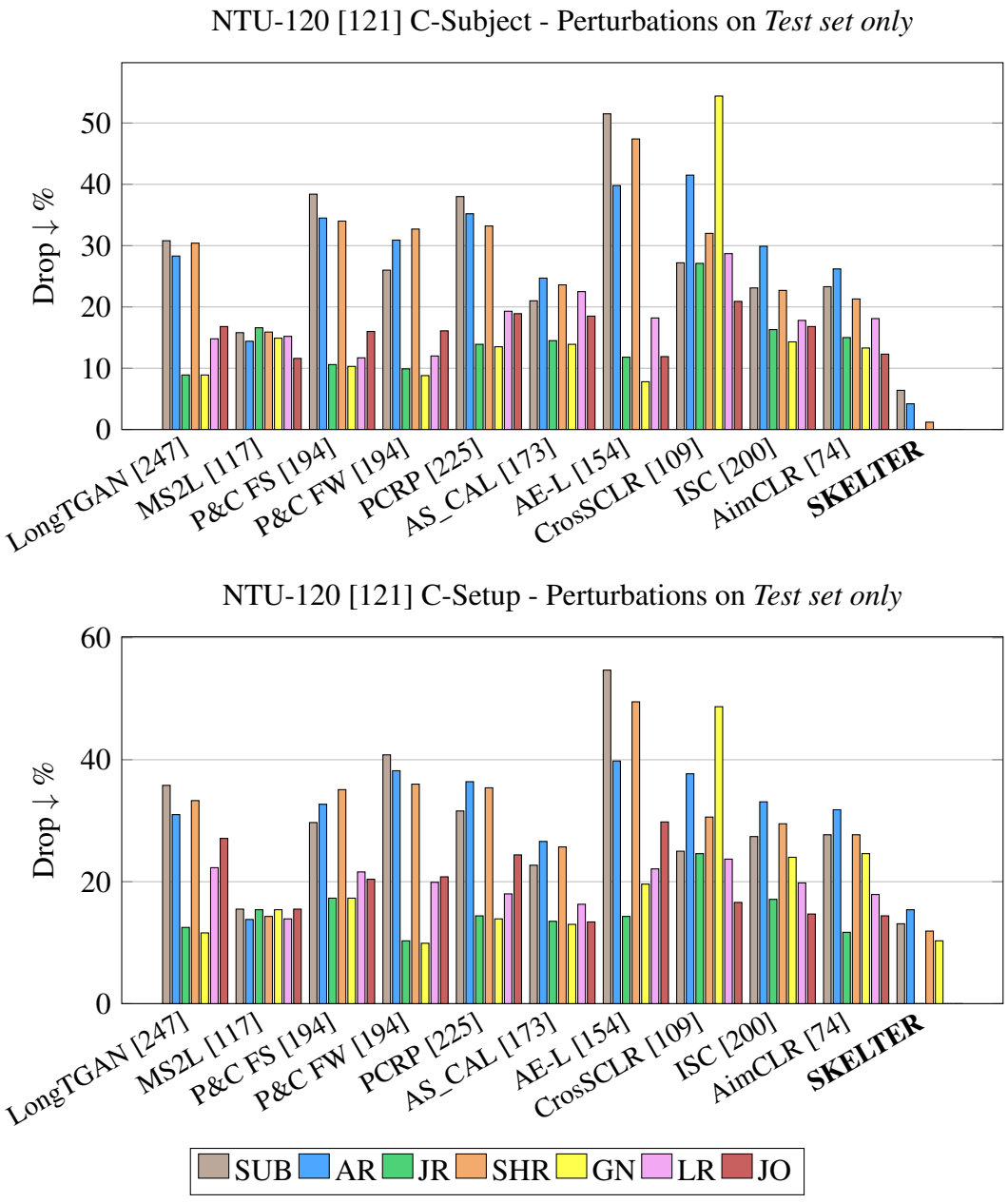


Figure 5.12 Performance drop ↓ % (related to the decrease of accuracy points) of SOTA U-HAR and SKELTER when ***only*** the *testing* splits of NTU-120 [121] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The lowest bars represent the best results.

5.8 Comparison with SOTA - Data Perturbation

By first verifying if the initial claim of this chapter is valid (*i.e.,* U-HAR methods are not resilient to data perturbations), all SOTA were evaluated by supplying their code publicly in two distinct evaluation phases:

- Investigate the accuracy results and performance drop of SOTA U-HAR and SKELTER when data perturbation is applied *only* on the test set, where the SOTA models are pre-trained using the original and unaltered data.
  Tables 5.1 and 5.2 report quantitative results, whereas Figures 5.11 and 5.12 represent the graphical counterpart in terms of bar plots (lower the bars, better the results).
- Investigate the accuracy results and performance drop of SOTA U-HAR and SKELTER when data perturbation is applied on *both* the train and test set, *de-facto* re-training from scratch all SOTA models providing perturbed data.
  Tables 5.3 and 5.4 report quantitative results, whereas Figures 5.13 and 5.14 represent the graphical counterpart in terms of bar plots (lower the bars, better the results).

Overall, the extensive quantitative and qualitative results confirm and demonstrate the sensible weakness in performance (*i.e.,* classification accuracy) of these approaches *w.r.t.* such perturbations, showing that all the methods' performance decrease when the testing data is corrupted, in some cases up to 70%. However, it is important to notice that even for the cases in which the set of data perturbations are introduced to the models in their training, there still exist remarkable drops in the performance, up to 45%. The reader can observe that for the perturbed data, the accuracy of SKELTER is better than the others in all datasets: such strong drops are not observed for SKELTER, proving its better denoising capabilities compared to SOTA. In other words, the performance drop of SKELTER is lower than others, and its performance is more accurate than others with the perturbed data.

It is also important to highlight that some of the methods, such as AS-CAL [173], CrosSCLR [109], ISC [200], and AimCLR [74], all perform contrastive learning while they augment the data in terms of *e.g.,* Shear, Gaussian noise, and Rotation. Therefore, one can expect they would be more resistant to the corresponding perturbations. However, compared to SKELTER, their performance decrease is relevant.

Performance Accuracy (ACC %) – Perturbations on *Train & Test set*										
	NTU-60 [181] C-Subject									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	52.1	11.6	32.8	34.7	20.1	35.3	30.9	32.2	28.5	23.6
MS2L [117]	52.6	32.8	25.4	36.0	22.1	40.0	33.1	42.7	34.3	18.3
P&C FS [194]	50.6	18.9	28.4	45.2	24.9	43.0	29.2	38.8	33.1	17.5
P&C FW [194]	50.7	24.5	27.0	48.3	21.7	44.8	32.7	38.6	34.5	16.2
PCRP [225]	53.9	15.2	18.7	46.7	22.8	51.9	40.7	51.1	35.4	18.5
AS_CAL [173]	58.5	41.9	33.9	45.3	34.4	46.5	40.1	50.0	41.6	16.9
AE-L [154]	69.9	52.8	58.8	66.2	59.1	66.2	63.4	57.2	59.8	10.1
CrosSCLR [109]	**77.8**	54.3	45.2	58.8	45.0	32.9	57.2	63.4	53.1	24.7
ISC [200]	76.3	55.8	52.2	65.2	52.2	65.8	59.0	65.6	60.1	16.2
AimCLR [74]	74.3	56.4	48.9	68.0	60.9	67.2	62.7	66.8	62.1	12.2
SKELTER	69.2	**57.2**	**60.0**	**69.0**	**63.7**	**67.9**	**63.9**	**68.9**	**64.4**	**4.8**

	NTU-60 [181] C-View									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	56.4	20.1	40.7	36.9	28.4	36.2	41.1	29.2	32.3	21.5
MS2L [117]	46.4	30.7	41.6	40.8	24.2	29.9	32.8	42.1	36.1	18.2
P&C FS [194]	76.3	15.4	27.1	63.4	24.7	65.6	55.2	39.7	43.2	15.7
P&C FW [194]	76.1	14.2	26.8	61.7	16.2	64.1	58.3	39.9	41.3	14.9
PCRP [225]	63.5	21.4	22.2	54.2	24.0	54.3	50.8	44.2	39.3	15.9
AS_CAL [173]	64.6	41.5	33.2	45.0	33.8	46.2	44.2	49.5	41.6	17.0
AE-L [154]	**85.4**	57.4	44.8	74.9	55.1	75.8	69.9	68.4	63.7	9.2
CrosSCLR [109]	83.4	60.9	48.1	75.9	60.7	49.1	70.0	70.2	63.3	24.8
ISC [200]	85.2	60.8	50.9	76.1	63.8	74.4	70.1	72.8	67.7	15.7
AimCLR [74]	79.7	61.7	57.2	77.0	68.0	76.0	71.2	75.9	70.1	11.5
SKELTER	78.5	**62.1**	**66.4**	**77.5**	**70.5**	**76.8**	**71.9**	**77.5**	**71.8**	**6.7**

Table 5.3 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-60 [181] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The average (AVG) accuracy and the Drop, ↓ *w.r.t.* clean data (CLN), are given (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The best results of each column are given in **bold**.

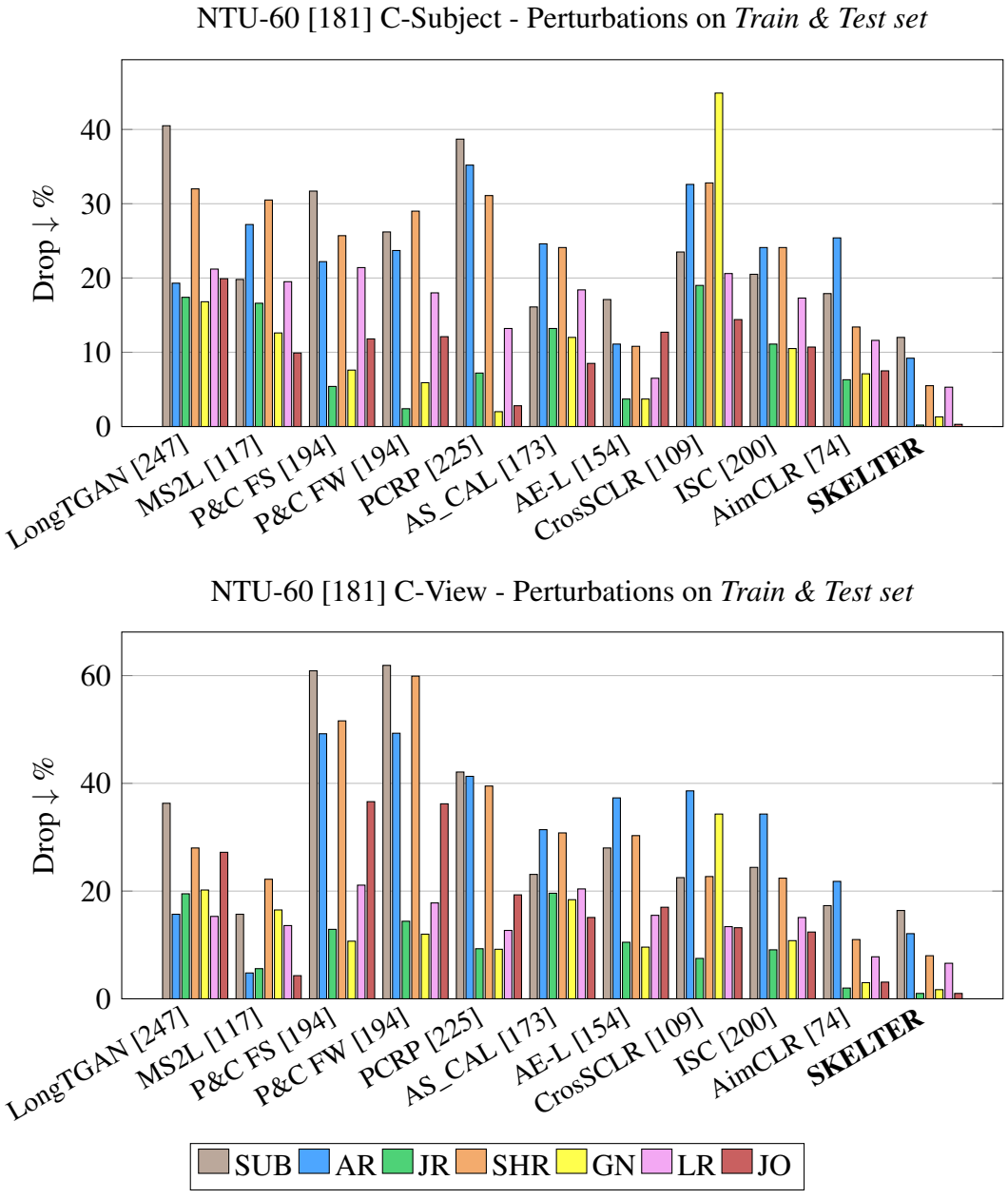


Figure 5.13 Performance drop ↓ % (related to the decrease of accuracy points) of SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-60 [181] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The lowest bars represent the best results.

Performance Accuracy (ACC %) – Perturbations on *Train & Test set*										
	NTU-120 [121] C-Subject									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	35.6	2.5	28.5	33.8	18.7	32.5	32.9	29.9	25.8	9.8
MS2L [117]	24.3	10.5	20.4	17.6	20.1	16.2	22.5	19.2	18.0	6.3
P&C FS [194]	40.5	12.6	22.7	32.7	12.8	35.4	30.8	33.6	26.3	14.2
P&C FW [194]	40.3	18.3	26.0	36.7	13.8	33.9	32.4	34.0	27.7	12.6
PCRP [225]	41.7	10.2	13.4	35.1	12.4	33.7	31.7	33.3	25.4	16.3
AS_CAL [173]	48.6	32.5	21.8	32.4	23.3	21.0	47.2	42.2	31.0	17.6
AE_L[154]	59.1	44.0	32.7	51.9	36.3	53.4	49.9	50.6	45.9	13.2
CrosSCLR [109]	67.9	44.2	35.9	50.1	47.1	36.9	52.4	53.3	46.9	21.0
ISC [200]	67.1	45.1	38.6	51.8	46.5	55.5	50.1	52.9	49.4	17.7
AimCLR [74]	**68.2**	45.9	44.3	54.2	49.9	56.1	52.2	57.4	51.9	16.3
SKELTER	52.9	**46.5**	**48.7**	**58.2**	**51.7**	**59.1**	**53.9**	**58.9**	**53.9**	**0**

	NTU-120 [121] C-Setup									
	CLN	SUB	AR	JR	SHR	GN	LR	JO	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	39.7	5.7	27.8	35.5	10.8	33.4	29.9	21.9	23.8	15.9
MS2L [117]	23.8	17.6	10.1	10.7	20.4	20.0	10.4	12.4	21.9	1.9
P&C FS [194]	42.4	25.9	23.8	36.9	20.0	31.2	30.7	30.7	27.5	14.9
P&C FW [194]	42.9	12.4	20.5	40.1	21.5	36.0	32.7	30.8	28.4	14.5
PCRP [225]	45.1	20.4	18.0	44.0	15.9	38.3	25.9	37.4	28.8	16.3
AS_CAL [173]	49.2	30.7	24.4	33.6	25.0	25.6	30.0	49.0	31.0	18.2
AE-L [154]	62.4	40.0	33.8	47.3	37.2	43.9	42.1	48.2	42.7	19.7
CrosSCLR [109]	66.7	42.4	34.7	45.9	39.9	39.7	43.2	54.4	45.6	21.1
ISC [200]	67.9	42.0	36.3	52.2	40.0	44.0	50.8	55.2	47.9	20.0
AimCLR [74]	**68.8**	42.6	38.8	59.4	43.1	44.9	53.9	58.7	50.3	18.5
SKELTER	56.0	**42.9**	**40.6**	**60.9**	**44.1**	**45.7**	**56.5**	**60.5**	**50.2**	**5.8**

Table 5.4 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-120 [121] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The average (AVG) accuracy and the Drop, ↓ *w.r.t.* clean data (CLN), are given (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The best results of each column are given in **bold**.

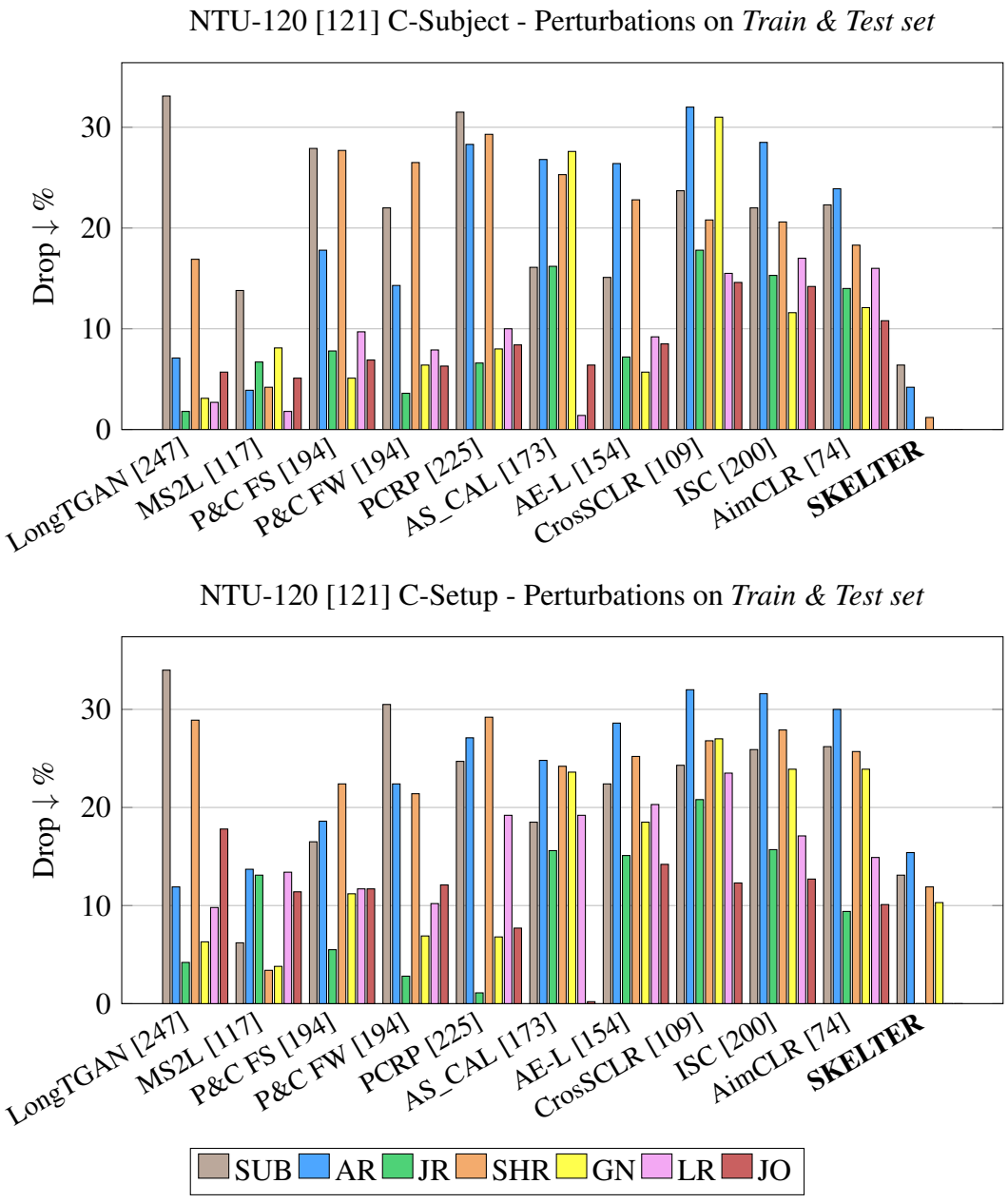


Figure 5.14 Performance drop ↓ % (related to the decrease of accuracy points) of SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-120 [121] are perturbed by: SUB, AR, JR, SHR, GN, LR and JO (see Section 5.2 for definitions). The lowest bars represent the best results.

5.9 Comparison with SOTA - Data Alteration

This section reports the performance of SOTA U-HAR when rotation (ROT) and reversed motion (RM) are applied to the datasets, with the same experimental pipeline described in the previous section (Tables 5.5 and 5.6, Figures 5.15 and 5.16). These results also include SKELTER's performance in three settings to examine the importance of using the proposed rotation-invariance and triplet losses:

- Pure SKELTER: using only the $\mathscr{L}_{\text{MSE}}$ loss (Equation 5.15)
- SKELTER with the rotation invariance loss (Equation 5.23)
- SKELTER with the triplet loss $\mathscr{L}_{\text{contr}}$ (Equation 5.24)

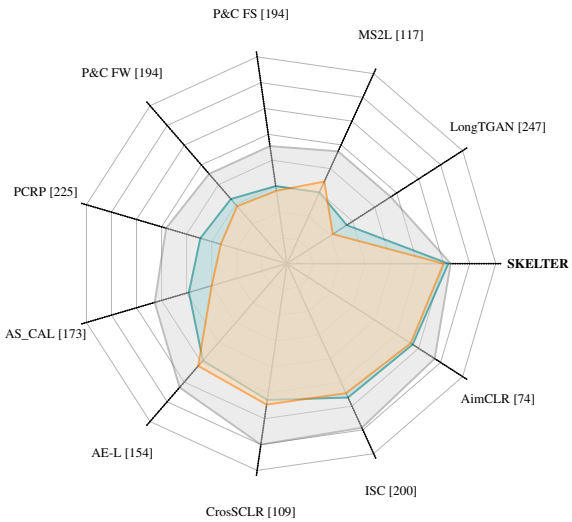
The reader can observe the same trends in the previous section, such that when Rotation and Reversed Motion are applied, the performance of SKELTER drop less than SOTA methods while performing better than all SOTA in terms of accuracy. Additionally, the proposed *rotation invariance* head and the inclusion of *triplet loss for temporal motion consistency* always improve the performance, achieving the best out of all.

Performance Accuracy (ACC %) – Alterations on *Test set only*										
	NTU-60 [181] C-Subject					NTU-60 [181] C-View				
	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	52.1	23.4	30.1	26.7	25.4	56.4	32.0	20.4	26.2	30.2
MS2L [117]	52.6	38.2	33.4	35.8	16.8	46.4	33.8	34.2	34.0	12.4
P&C FS [194]	50.6	31.0	33.8	32.4	18.2	76.3	46.2	47.8	47.0	29.3
P&C FW [194]	50.7	32.6	36.2	34.4	16.3	76.1	37.7	48.7	43.2	32.9
PCRP [225]	53.9	29.9	38.8	34.3	19.6	63.5	37.5	40.0	38.7	24.8
AS_CAL [173]	58.5	33.3	43.7	38.5	20.0	64.6	33.0	43.9	38.4	26.2
AE-L [154]	69.9	57.0	54.1	55.5	14.4	**85.4**	58.8	58.2	58.5	26.6
CrosSCLR [109]	**77.8**	60.1	58.8	59.4	18.4	83.4	66.9	68.8	67.8	15.6
ISC [200]	76.3	60.3	62.9	61.6	14.7	85.2	67.2	70.1	68.6	16.6
AimCLR [74]	74.3	62.7	63.4	63.1	11.2	79.7	69.4	73.0	71.2	8.5
SKELTER (Pure)	69.2	<u>63.8</u>	<u>65.2</u>	64.5	4.7	78.5	<u>70.1</u>	<u>76.1</u>	73.1	5.4
SKELTER (w/*RotHeads*))	69.2	**66.2**	-	66.2	**3.0**	78.5	**75.2**	-	75.2	**3.3**
SKELTER (*w/* $\mathscr{L}_{\text{contr}}$)	69.2	-	**68.7**	**68.7**	**0.5**	78.5	-	**78.0**	**78.0**	**0.5**

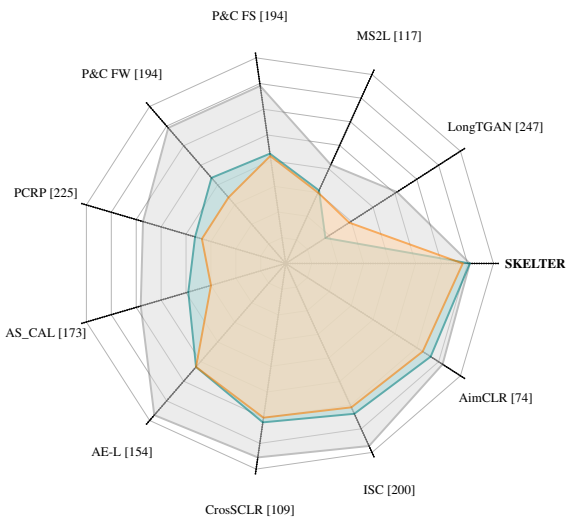
	NTU-120 [121] C-Subject					NTU-120 [121] C-Setup				
	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	35.6	18.2	30.9	24.5	11.1	39.7	24.2	20.0	22.1	17.6
MS2L [117]	24.3	13.3	14.9	14.1	10.2	23.8	12.8	17.5	15.1	8.8
P&C FS [194]	40.5	22.6	27.8	25.2	15.3	42.4	20.8	22.4	21.6	20.8
P&C FW [194]	40.3	20.4	27.4	23.9	16.4	42.9	21.4	34.9	28.1	14.8
PCRP [225]	41.7	24.9	30.5	27.7	14.0	45.1	23.3	30.5	26.9	18.2
AS_CAL [173]	48.6	20.0	32.8	26.4	22.2	49.2	21.9	34.2	28.1	21.1
AE_L[154]	59.1	42.4	46.2	44.3	14.8	62.4	40.7	48.8	44.7	17.7
CrosSCLR [109]	67.9	45.9	50.1	48.0	19.9	66.7	52.8	54.2	53.5	13.2
ISC [200]	67.1	50.7	48.1	49.4	17.7	67.9	53.0	54.7	53.8	14.1
AimCLR [74]	**68.2**	51.2	52.9	52.1	16.1	**68.8**	54.1	56.0	55.1	13.7
SKELTER (Pure)	52.9	<u>54.1</u>	<u>54.6</u>	54.3	0	56.0	<u>55.3</u>	<u>57.7</u>	56.5	0
SKELTER (w/*RotHeads*)	52.9	**56.6**	-	56.6	**0**	56.0	**58.8**	-	58.8	**0**
SKELTER (*w/* $\mathscr{L}_{\text{contr}}$)	52.9	-	**59.0**	**59.0**	**0**	56.0	-	**61.0**	**61.0**	**0**

Table 5.5 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when **only** the *testing* splits of NTU-60 [181] and NTU-120 [121] are altered by: ROT and RM (see Section 5.2 for definitions) in terms of the average (AVG) accuracy and the Drop ↓ *w.r.t.* clean data (CLN) (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The results of SKELTER are given in three settings: (a) pure SKELTER, (b) SKELTER with the rotation head (RotHeads) and (c) SKELTER with $\mathscr{L}_{\text{contr}}$. The best results of each column are given in **bold** while the second best result is <u>underlined</u>.

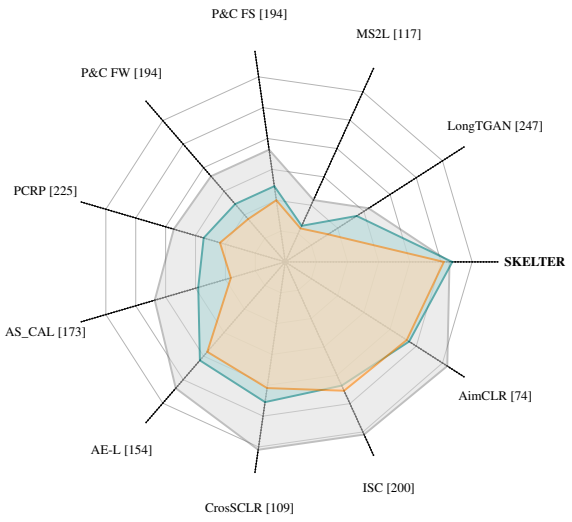
NTU-60 [181] C-Subject - *Test set only*



NTU-60 [181] C-View - *Test set only*



NTU-120 [121] C-Subject - *Test set only*



NTU-120 [121] C-Setup - *Test set only*

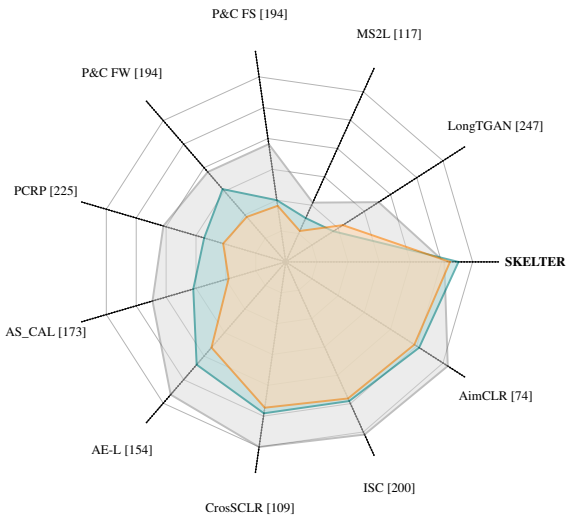


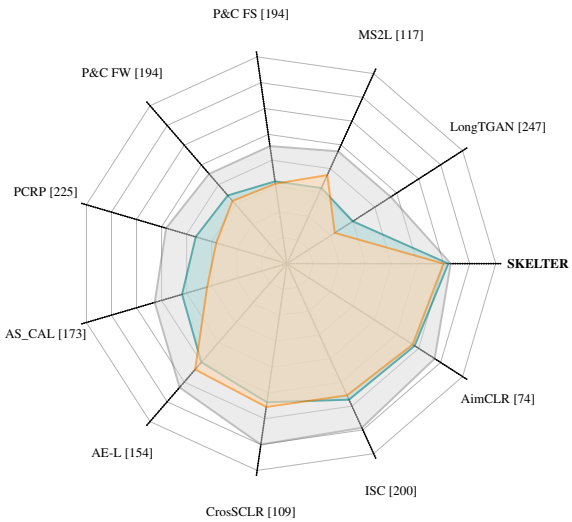
Figure 5.15 Kiviat plots in terms of Accuracy (%) between the SOTA U-HAR and SKELTER when **only** the *testing* splits of NTU-60 [181] and NTU-120 [121] are altered by: ROT and RM (see Section 5.2 for definitions). Each ray line represents the accuracy results of each method (where the centre is the zero), and coloured lines and areas represent the Accuracy values *w.r.t.* the CLN (grey), ROT (blue) and RM (orange) applied. CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets.

Performance Accuracy (ACC %) – Alterations on *Train & Test set*										
	NTU-60 [181] C-Subject					NTU-60 [181] C-View				
	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	52.1	24.9	33.6	29.2	22.9	56.4	34.0	24.3	29.1	27.3
MS2L [117]	52.6	41.0	35.2	38.1	14.5	46.4	35.9	37.0	36.4	10.0
P&C FS [194]	50.6	34.1	35.7	34.9	15.7	76.3	48.5	49.5	49.0	27.3
P&C FW [194]	50.7	35.0	38.1	36.5	14.2	76.1	40.2	50.0	45.1	31.0
PCRP [225]	53.9	31.7	40.1	35.9	18.0	63.5	40.1	42.3	41.2	22.3
AS_CAL [173]	58.5	35.5	46.5	41.0	17.5	64.6	35.2	46.2	40.7	23.9
AE-L [154]	69.9	59.4	55.2	57.3	12.6	**85.4**	62.9	60.4	61.6	23.8
CrosSCLR [109]	**77.8**	61.4	59.9	60.6	17.2	83.4	68.2	70.3	69.2	14.2
ISC [200]	76.3	61.8	63.2	62.5	13.8	85.2	68.4	72.8	70.6	14.6
AimCLR [74]	74.3	63.2	64.9	64.1	10.2	79.7	69.9	74.2	72.1	7.6
SKELTER (Pure)	69.2	<u>63.8</u>	<u>65.2</u>	64.5	4.7	78.5	<u>70.1</u>	<u>76.1</u>	73.1	5.4
SKELTER (w/*RotHeads*))	69.2	**66.2**	-	66.2	**3.0**	78.5	**75.2**	-	75.2	**3.3**
SKELTER (*w/* $\mathscr{L}_{\text{contr}}$)	69.2	-	**68.7**	**68.7**	**0.5**	78.5	-	**78.0**	**78.0**	**0.5**

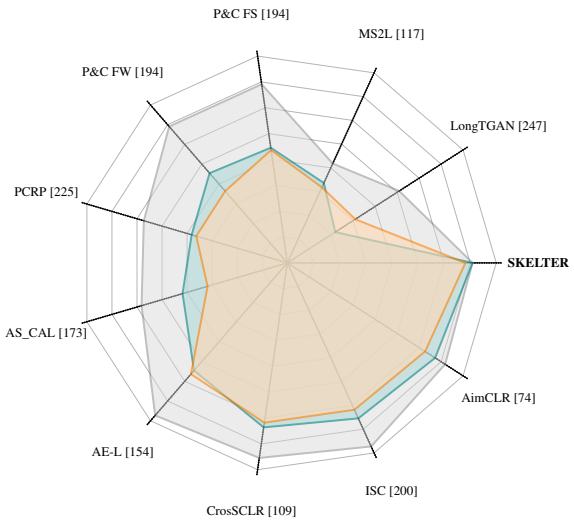
	NTU-120 [121] C-Subject					NTU-120 [121] C-Setup				
	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)	CLN	ROT	RM	AVG ACC (%)	Drop ↓ (%)
LongTGAN [247]	35.6	20.2	33.0	26.6	9.0	39.7	26.8	22.4	24.6	15.1
MS2L [117]	24.3	16.2	19.7	17.9	6.4	23.8	15.4	19.9	17.6	6.2
P&C FS [194]	40.5	25.0	30.7	27.8	12.7	42.4	23.8	24.9	24.3	18.1
P&C FW [194]	40.3	24.2	30.4	27.3	13.0	42.9	24.7	37.0	30.8	12.1
PCRP [225]	41.7	26.8	32.0	29.4	12.3	45.1	25.8	33.5	29.6	15.5
AS_CAL [173]	48.6	23.9	34.8	29.3	19.3	49.2	24.5	36.2	30.3	18.9
AE_L[154]	59.1	46.7	48.0	47.3	11.8	62.4	42.1	50.1	46.1	16.3
CrosSCLR [109]	67.9	49.4	52.8	51.1	16.8	66.7	53.4	56.7	55.1	11.6
ISC [200]	67.1	53.0	50.8	51.9	15.2	67.9	53.9	56.9	55.4	12.5
AimCLR [74]	**68.2**	53.9	53.1	53.5	14.7	**68.8**	54.2	57.2	55.7	13.1
SKELTER (Pure)	52.9	<u>54.1</u>	<u>54.6</u>	54.3	0	56.0	<u>55.3</u>	<u>57.7</u>	56.5	0
SKELTER (w/*RotHeads*)	52.9	**56.6**	-	56.6	0	56.0	**58.8**	-	58.8	0
SKELTER (*w/* $\mathscr{L}_{\text{contr}}$)	52.9	-	**59.0**	**59.0**	**0**	56.0	-	**61.0**	**61.0**	**0**

Table 5.6 Performance comparisons in terms of accuracy (%) between the SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-60 [181] and NTU-120 [121] are altered by: ROT and RM (see Section 5.2 for definitions) in terms of the average (AVG) accuracy and the Drop ↓ *w.r.t.* clean data (CLN) (the lower, the better). CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets. The results of SKELTER are given in three settings: (a) pure SKELTER, (b) SKELTER with the rotation head (RotHeads) and (c) SKELTER with $\mathscr{L}_{\text{contr}}$. The best results of each column are given in **bold** while the second best result is <u>underlined</u>.

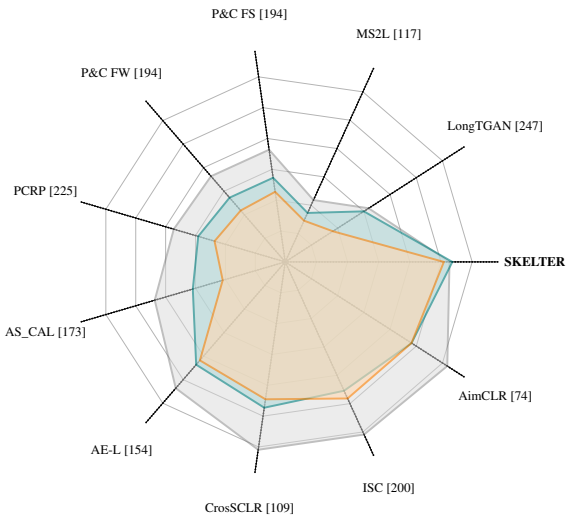
NTU-60 [181] C-Subject - *Train & Test set*



NTU-60 [181] C-View - *Train & Test set*



NTU-120 [121] C-Subject - *Train & Test set*



NTU-120 [121] C-Setup - *Train & Test set*

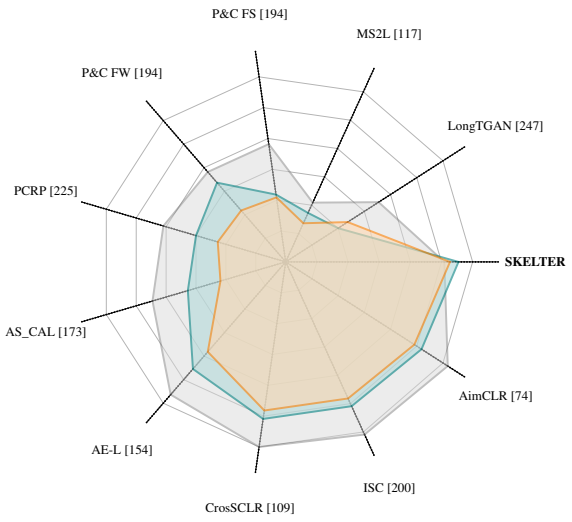


Figure 5.16 Kiviat plots in terms of Accuracy (%) between the SOTA U-HAR and SKELTER when **both** *training* and *testing* splits of NTU-60 [181] and NTU-120 [121] are altered by: ROT and RM (see Section 5.2 for definitions). Each ray line represents the accuracy results of each method (where the centre is the zero), and coloured lines and areas represent the Accuracy values *w.r.t.* the CLN (grey), ROT (blue) and RM (orange) applied. CLN stands for clean data, *i.e.,* usage of original data as supplied by the datasets.

5.10 Qualitative Results

Figures 5.17 and 5.18 shows the visualisations of a skeletal action sequence "Throw" picked from the NTU-60 [181] Cross-View split. As a reference for all illustrations, the original unaltered skeletons are represented in *blue* colour, their perturbed counterpart (by applying one of the perturbations from Section 5.2) in *red* colour, and the denoised skeletons obtained from the proposed SKELTER model in *green* colour.

Figure 5.17 depicts some of the proposed perturbations, which potentially could negatively affect the proposed method's performances and the state-of-the-art. Starting on a variety of perturbed data, the effectiveness of SKELTER can be seen through ***the smoothed denoised skeleton reconstruction*** of it even in case of heavy data perturbation.

As a concluding remark, Figure 5.18 shows how SKELTER reconstructs and denoise each sample accordingly as the sequence unfolds *w.r.t.* its temporal dimension.

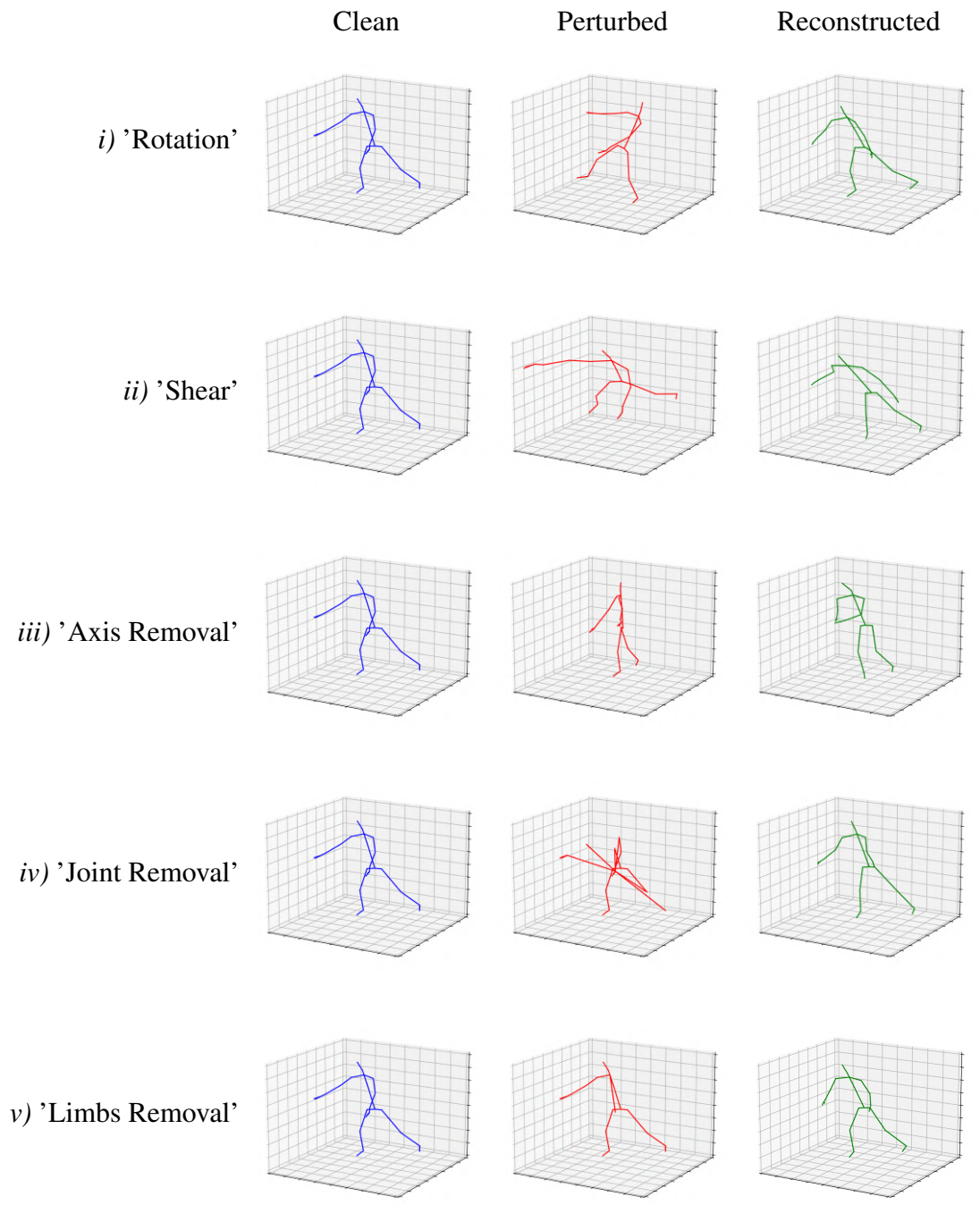


Figure 5.17 SKELTER reconstruction. Starting from the clean "Throw" skeleton action sequences (first column, blue), a perturbation is applied (middle column, red) and gives the obtained sequence as the input, which is then reconstructed (last column, green). Each row is a sample of different perturbations. *From first to the last row: 'i)'* rotated skeleton (along X, Y, and Z axes), *'ii)'* sheared skeleton, *'iii)'* 2D skeleton (all coordinated of X axis set to zero), *'iv)'* joint-corrupted skeleton (random joints coordinates set to zero), *'v)'* no-limb skeleton (the joints set coordinates of the left arm set to zero).

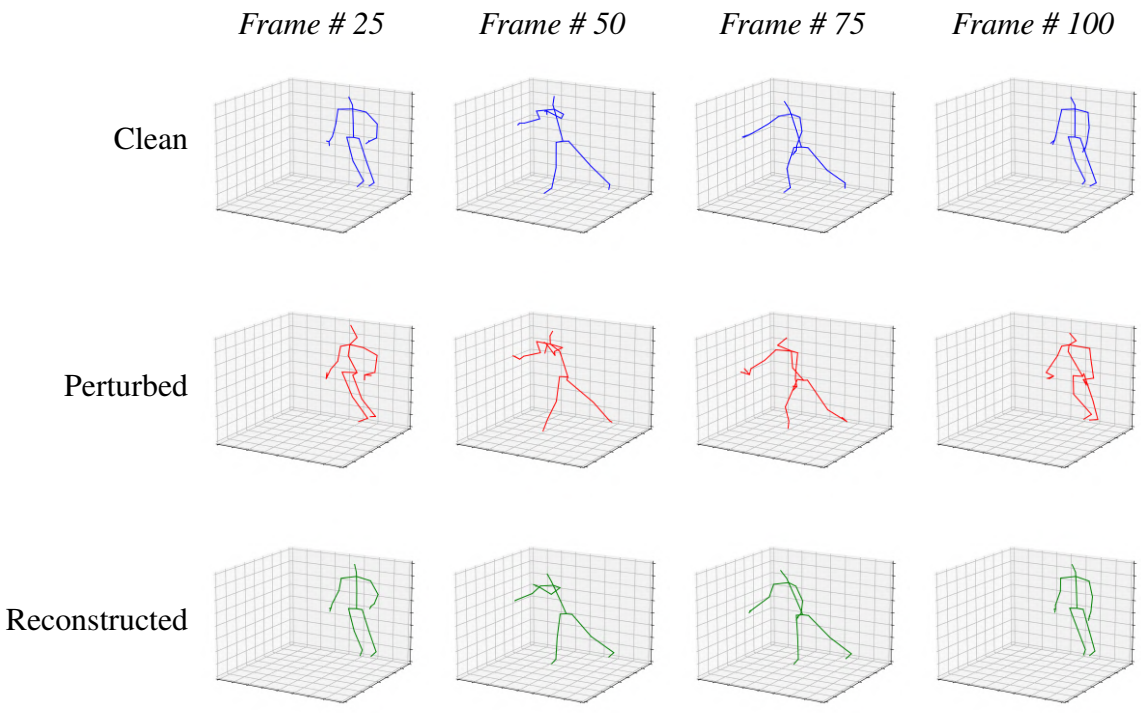


Figure 5.18 Original (blue), perturbed (red), and SKELTER-reconstructed (green) skeletal pose. As the data perturbation *Gaussian additive noise* is applied, each column represents one particular frame of the overall sequence. *Left to right:* frame #25, frame #50, frame #75, frame #100.

Sample pose from a real-life scenario [19] *Sample pose from NTU-60* [181]

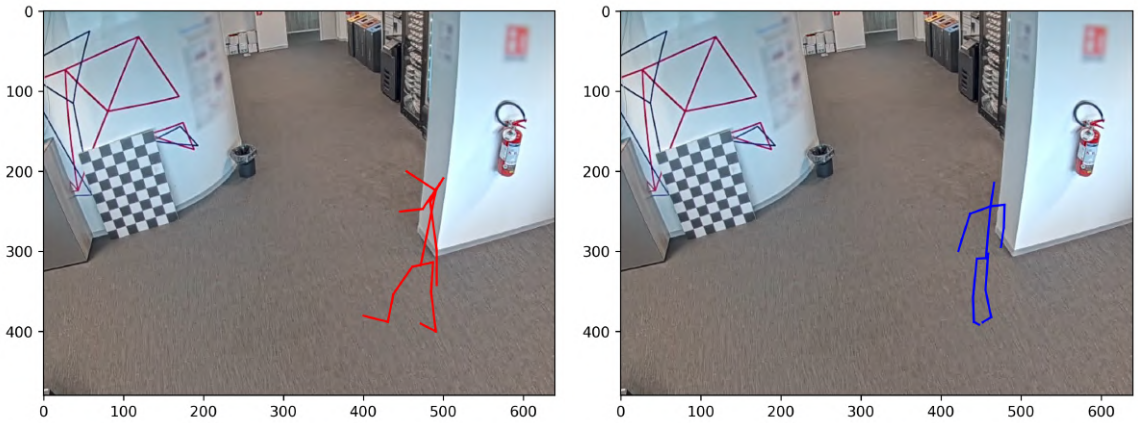


Figure 5.19 Graphical comparison between perturbed and real-life sample poses. ***Left:*** 2D skeleton pose estimated using OpenPose [19], from a sample captured from a CCTV video stream (red skeleton). Camera calibration and reference origin point estimated beforehand for the 3D-to-2D conversion of the perturbed dataset. All 2D poses were normalised and centred *w.r.t.* the reference point, which is set identically to the perturbed poses. ***Right:*** a sample from NTU-60 [181] (blue skeleton) after applying the world-to-camera projection, using camera parameters obtained earlier, making sure that both distributions of poses are compatible with each other. Axis values correspond to the pixel values of the recorded frame (*i.e.,* 640x480). In both cases, the RGB background is left for illustration purposes.

		Missing Joints (AVG %)	**Missing Limbs (AVG %)**	**MMD [201] ($p < 0.05$)**
Perturbed NTU-60 [181]	CLN	0.32	0.49	0.0095
	SUB	5.89	0.01	0.0078
	AR	0.21	0.30	0.0313
	JR	2.11	0.57	0.0157
	SHR	0.89	1.32	0.0191
	GN	0.24	0.45	0.0294
	LR	20.87	25.38	0.0009
	JO	0.64	0.49	0.0103
	Real-world	13.45	22.76	-

Table 5.7 Statistics between perturbed NTU-60[181] and real-world 2D poses. Values of missing joints and limbs are reported as the average percentage *w.r.t.* all joints of 2D poses. MMD refers to the Maximum Mean Discrepancy [201] between the real-world 2D poses and each distinct proposed perturbation of NTU-60[181].

5.11 Real-Life scenario - a case study

Section 5.1 sets the foundations of the overall claim of this chapter: devise an unsupervised model, U-HAR oriented, capable of handling data corruption of skeleton poses in any conditions which can be found in more practical scenarios. Subsequent sections proved the usefulness of SKELTER for this particular task. Still, an important question remained unanswered: if the proposed skeleton poses perturbations or alterations (described in Section 5.2) plausibly reflect data corruption which could be found in real-world scenarios. This section describes a case study about a simulated scenario, with a comparison between a perturbed dataset (*i.e.,* perturbed NTU-60[181]) and real-world 2D skeleton poses. The goal is to demonstrate that both data distributions can overlap each other, confirming the plausibility of proposed perturbation *w.r.t.* real data.

To achieve this, a set of 2D skeleton poses were captured from a CCTV video stream using OpenPose [19]. Recordings were made in an office scenario, where the original video stream was deleted later to maintain the privacy of people detected. This can be seen in Figure 5.19, where a clean office background is left only for visualisation purposes: the *left* pose represents a sample frame from the real-world poses captured, and the *right* pose represents a sample frame from the perturbed dataset. In addition, camera parameters and a reference origin point were recorded and estimated to ensure an equal comparison for both data distributions. As for the *perturbed dataset*, a world-to-camera projection had to be performed to convert its 3D poses into 2D poses, compatible in terms of the number of joints (keeping only a subset of 17 skeleton joints common to each other), their order and their pixel position *w.r.t.* camera parameters estimated beforehand. The reference origin point was necessary to keep all poses coming from both datasets aligned. In addition, for the perturbed dataset, to add variety and add realism, each pose was rotated along its Z-axis before performing the camera projection to ensure a similar behaviour naturally occurring in real-life scenarios (*i.e.,* rotations of people detected). As the last step, pose normalisation in unit-norm was applied for both datasets.

Table 5.7 reports some statistics related to the number of missing joints, missing limbs (*i.e.,* a group of joints) and the *Maximum Mean Discrepancy* (MMD). Missing joints and limbs refer to the averaged percentage value of each distinct joint which is missing (*i.e.,* zero-valued) for the former and the missing values of groups of joints which form one of the four limbs (*i.e.,* arms and legs). Results show that the Limbs Removal perturbation is the closest *w.r.t.* real-world 2D poses, simulating the high occurrence of missing entire body parts due to heavy occlusions instead of milder occlusions like single Joints Removal. Maximum mean

discrepancy (MMD) [201] is a kernel-based statistical test used to determine whether two given data distributions are identical. In addition to being used as a statistical test (as an integral probability metric), MMD can also be used as a loss or cost function in various machine learning algorithms (as a distance, or difference, between feature means). It is often used as a simpler discriminator because of its easy implementation and the rich kernel-based theory that underlies its principles. The kernel trick was used to estimate this measure, and a lower value denotes a statistically-significative overlap between the two data distributions. It was performed by comparing the real-world 2D poses with each, and distinct NTU-60 [181] dataset perturbation proposed in Section 5.2. In all cases, its value was below the null hypothesis $p < 0.05$, denoting the plausibility of such proposed data perturbation strategies, despite the semantic differences and type of motion involved.

5.12 Concluding remarks

Robust human action recognition is a fundamental capability in artificial intelligence systems, and this chapter shows that data perturbations and alterations can severely reduce the performance of SOTA approaches. First, several perturbations and alterations that could be commonly found when extracting skeletal data in realistic environments (*e.g.,* occlusions, geometrical distortions, noise, *etc.*) were introduced. Then, a novel framework, based on a transformer encoder-decoder and accepting 3D-skeletal data as the input, is presented. Additional losses grant to obtain robust representations against rotation variances and to provide temporal motion consistency. Indeed, results show that the current methods have a relevant drop in performance while the proposed method is less affected by such data perturbations and alterations. This confirms that the proposed approach might be prone to be better resistant to challenging realistic operational scenarios.

Chapter 6

Conclusions

In conclusion, this thesis presents novel approaches to human activity recognition using unsupervised learning techniques, both actions and emotions. The proposed methods address several operational limitations of previous approaches, including difficulty handling the temporal dimension, noise in skeletal data, and computational challenges. The following sections define the concluding remarks of each research topic addressed in this thesis, focusing on drawbacks, limitations and insights for future works.

6.1 Subspace Clustering

The results of the experimental analysis presented in Chapter 3 demonstrate the effectiveness of the proposed fully unsupervised pipeline for human action recognition (HAR). The pipeline, which combines subspace clustering methods based on the self-expressiveness property with covariance representation and temporal subspace clustering using dictionary learning and temporal Laplacian regularisation, was validated on eight different datasets with a wide variety of action types, the number of action classes, and experimental protocols. Across these benchmarks, the proposed pipeline consistently outperformed previous subspace clustering methods and, in some cases, even outperformed supervised approaches.

Drawbacks, limitations, and future works: One of the main drawbacks of the subspace clustering approach for unsupervised human action recognition (HAR) is its limited scalability. Due to the space complexity of the affinity matrix, which is required for the classification task and grows quadratically with the size of the dataset, the applicability of such algorithms is restricted to smaller datasets. This can limit the ability of the approach to capture the full

range of nuances in human actions and may impair its performance on more complex and diverse datasets. To address this issue, a promising direction for future work is to develop subspace clustering algorithms capable of handling large-scale or big-data regimes. Overall, the results of this chapter demonstrate the potential of the proposed pipeline for unsupervised HAR but also highlight the need for further research to address its scalability limitations and enable its application to more complex and diverse datasets.

6.2 *AE-L*: convolutional residual autoencoder

The experimental analysis results presented in Chapter 4 demonstrate the effectiveness of the proposed convolutional autoencoder with Laplacian regularisation (*AE-L*) method for unsupervised feature learning in the context of 3D skeleton-based action and emotion recognition. The proposed method was validated on large-scale benchmarks for both action and emotion recognition, showing superior performance compared to state-of-the-art unsupervised methods in various settings, including cross-subject, cross-view, and cross-setup. The incorporation of gradient reversing into the *AE-L* framework also resulted in improved invariance to camera viewpoint changes. These findings highlight the potential of unsupervised learning approaches for 3D skeleton-based action and emotion recognition and suggest that the proposed *AE-L* method represents a valuable contribution to the field, capable of learning more distinctive action and emotion features compared to the prior art.

Drawbacks, limitations, and future works: One of the major drawbacks of the existing unsupervised human action recognition (U-HAR) methods is that they have largely been evaluated on benchmark datasets recorded in controlled experimental settings. These datasets may not adequately capture the challenges and complexities that can arise in real-world scenarios, such as noisy data, severe occlusions, and errors in sensors or pose estimators. As a result, the performance of these methods may not generalise well to more realistic environments. To address this issue, a promising direction for future work is to evaluate the performance of U-HAR methods on datasets that more closely reflect real-world conditions. Additionally, it may be beneficial to investigate the robustness of U-HAR methods to errors or missing data, as this is a common issue that can arise in real-world scenarios.

6.3 SKELTER: transformer for real-world perturbed data

In conclusion, Chapter 5 has demonstrated the importance of the robustness of the proposed SKELTER in human action recognition (HAR) in need of approaches that can effectively handle data perturbations and alterations commonly found in realistic environments. A novel framework based on a transformer encoder-decoder and incorporating additional losses to promote rotation-invariant and temporal motion-consistent representations was presented to address this issue. The proposed approach was shown to be significantly less affected by data perturbations and alterations than state-of-the-art (SOTA) methods, indicating its potential to be more resistant to challenging real-life scenarios. Additionally, a systematic analysis of SOTA unsupervised HAR algorithms in the presence of perturbed data highlighted the need for noise-resistant models in these types of environments. Overall, this chapter's results demonstrate the SKELTER framework's potential as a solution for robust unsupervised HAR in challenging, in-the-wild settings.

Drawbacks, limitations, and future works: While the results of SKELTER, presented in Chapter 5, demonstrate the approach's effectiveness, there is still a significant need for further research and exploration of this topic since it is a relatively unexplored research topic, and has not yet been widely studied. Overall, the results of this work demonstrate the potential of the proposed approach for handling perturbed data and achieving robust results but also highlight the need for further research to fully understand and optimise this approach. To address this issue, a promising direction for future work is to put more research effort into developing and evaluating new *unsupervised* approaches for handling perturbed data in various applications. Such research will be beneficial not only for the computer vision community but also for a wide range of real-time and real-world applications where data perturbations are common.

References

[1] Amor, B. B., Su, J., and Srivastava, A. (2015). Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):1–13.

[2] Argyle, M. (2013). *Bodily communication*. Routledge.

[3] Aviezer, H., Trope, Y., and Todorov, A. (2012). Body cues, not facial expressions, discriminate between intense positive and negative emotions. *Science*, 338(6111):1225–1229.

[4] Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*.

[5] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11).

[6] Ben Tanfous, A., Drira, H., and Ben Amor, B. (2018). Coding kendall's shape trajectories for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2840–2849.

[7] Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*.

[8] Beyan, C., Karumuri, S., and Volpe, G. (2021). Modeling multiple temporal scales of full-body movements for emotion classification. *IEEE Transactions on Affective Computing*, pages 1–1.

[9] Beyan, C., Shahid, M., and Murino, V. (2018). Investigation of small group social interactions using deep visual activity-based nonverbal features. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 311–319.

[10] Beyan, C., Zunino, A., Shahid, M., and Murino, V. (2019). Personality traits classification using deep visual activity-based nonverbal features of key-dynamic images. *IEEE Transactions on Affective Computing*, 12(4):1084–1099.

[11] Bian, C., Feng, W., Wan, L., and Wang, S. (2021). Structural knowledge distillation for efficient skeleton-based action recognition. *IEEE Transactions on Image Processing*, 30:2963–2976.

[12] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402. IEEE.

[13] Bloom, V., Argyriou, V., and Makris, D. (2016). Hierarchical transfer learning for online recognition of compound actions. *CVIU*, 144:62–72.

[14] Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267.

[15] Bregonzio, M., Gong, S., and Xiang, T. (2009). Recognising action as clouds of space-time interest points. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1948–1955. IEEE.

[16] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[17] Burgoon, J. K., Manusov, V., and Guerrero, L. K. (2021). *Nonverbal communication*. Routledge.

[18] Calvo, R. A. and D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on affective computing*, 1(1):18–37.

[19] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299.

[20] Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

[21] Castellano, G., Villalba, S. D., and Camurri, A. (2007). Recognising human emotions from body movement and gesture dynamics. In *International conference on affective computing and intelligent interaction*, pages 71–82. Springer.

[22] Cavazza, J., Morerio, P., and Murino, V. (2019). Scalable and compact 3d action recognition with approximated rbf kernel machines. *Pattern Recognition*, 93:25–35.

[23] Cavazza, J., Zunino, A., San Biagio, M., and Murino, V. (2016). Kernelized covariance for action recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 408–413. IEEE.

[24] Chang, Y.-J., Chen, S.-F., and Huang, J.-D. (2011). A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570.

[25] Chen, T., Zhou, D., Wang, J., Wang, S., Guan, Y., He, X., and Ding, E. (2021a). Learning multi-granular spatio-temporal graph network for skeleton-based action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4334–4342.

[26] Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., and Hu, W. (2021b). Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368.

[27] Cheng, D. S. and Cristani, M. (2017). *Social Signal Processing for Surveillance*, page 331–348. Cambridge University Press.

[28] Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., and Lu, H. (2020). Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192.

[29] Cheng, K., Zhang, Y., He, X., Cheng, J., and Lu, H. (2021a). Extremely lightweight skeleton-based action recognition with shiftgcn++. *IEEE Transactions on Image Processing*, 30:7333–7348.

[30] Cheng, Y.-B., Chen, X., Chen, J., Wei, P., Zhang, D., and Lin, L. (2021b). Hierarchical transformer: Unsupervised representation learning for skeleton-based human action recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

[31] Cho, K. and Chen, X. (2014). Classifying and visualizing motion capture sequences using deep neural networks. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 122–130. IEEE.

[32] Cimen, G., Ilhan, H., Capin, T., and Gurcay, H. (2013). Classification of human motion based on affective state descriptors.

[33] Ciptadi, A., Goodwin, M. S., and Rehg, J. M. (2014). Movement pattern histogram for action recognition and retrieval. In *European conference on computer vision*, pages 695–710. Springer.

[34] Clopton, L., Mavroudi, E., Tsakiris, M., Ali, H., and Vidal, R. (2017). Temporal subspace clustering for unsupervised action segmentation. *CSMR REU*, pages 1–7.

[35] Code, P. W. (2022a). Skeleton Based Action Recognition Benchmarks on NTU RGB+D 120. https://paperswithcode.com/sota/skeleton-based-action-recognition-on-ntu-rgbd-1.

[36] Code, P. W. (2022b). Skeleton Based Action Recognition Benchmarks on NTU RGB+D 60. https://paperswithcode.com/sota/skeleton-based-action-recognition-on-ntu-rgbd.

[37] Cooper, H. and Bowden, R. (2007). Large lexicon detection of sign language. In *International Workshop on Human-Computer Interaction*, pages 88–97. Springer.

[38] Costeira, J. P. and Kanade, T. (1998). A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179.

[39] Coulson, M. (2004). Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of nonverbal behavior*, 28(2):117–139.

[40] Cowie, R., Sussman, N., and Ben-Ze'ev, A. (2011). Emotion: Concepts and definitions. In *Emotion-oriented systems*, pages 9–30. Springer.

[41] Crenn, A., Meyer, A., Konik, H., Khan, R. A., and Bouakaz, S. (2020). Generic body expression recognition based on synthesis of realistic neutral motion. *IEEE Access*, 8:207758–207767.

[42] Cristani, M., Del Bue, A., Murino, V., Setti, F., and Vinciarelli, A. (2020). The visual social distancing problem. *Ieee Access*, 8:126876–126886.

[43] Cristani, M., Raghavendra, R., Del Bue, A., and Murino, V. (2013). Human behavior analysis in video surveillance: A social signal processing perspective. *Neurocomputing*, 100:86–97.

[44] Dael, N., Goudbeek, M., and Scherer, K. R. (2013). Perceived gesture dynamics in nonverbal expression of emotion. *Perception*, 42(6):642–657.

[45] Dael, N., Mortillaro, M., and Scherer, K. R. (2012). The body action and posture coding system (bap): Development and reliability. *Journal of Nonverbal Behavior*, 36(2):97–121.

[46] Daoudi, M., Berretti, S., Pala, P., Delevoye, Y., and Bimbo, A. D. (2017). Emotion recognition by body movement representation on the manifold of symmetric positive definite matrices. In *International Conference on Image Analysis and Processing*, pages 550–560. Springer.

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

[48] de Gelder, B., Van den Stock, J., Meeren, H. K., Sinke, C. B., Kret, M. E., and Tamietto, M. (2010). Standing up for the body. recent progress in uncovering the networks involved in the perception of bodies and bodily expressions. *Neuroscience & Biobehavioral Reviews*, 34(4):513–527.

[49] Del Bue, A., Xavier, J., Agapito, L., and Paladini, M. (2011). Bilinear modeling via augmented lagrange multipliers (balm). *IEEE transactions on pattern analysis and machine intelligence*, 34(8):1496–1508.

[50] Deo, N. (2017). *Graph theory with applications to engineering and computer science*. Courier Dover Publications.

[51] Derbaix, C. and Pham, M. T. (1991). Affective reactions to consumption situations: A pilot investigation. *Journal of Economic Psychology*, 12(2):325–355.

[52] DMCD (2021). Dance motion capture database. http://dancedb.eu/.

[53] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, pages 65–72. IEEE.

[54] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

[55] Du, Y., Fu, Y., and Wang, L. (2015a). Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*, pages 579–583. IEEE.

[56] Du, Y., Wang, W., and Wang, L. (2015b). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of IEEE CVPR*, pages 1110–1118.

[57] Duric, Z., Gray, W. D., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M. J., Schunn, C., and Wechsler, H. (2002). Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. *Proceedings of the IEEE*, 90(7):1272–1289.

[58] Ekman, P. (1992). Are there basic emotions? *American Psychological Association*.

[59] Ekman, P. and Friesen, W. V. (1986). A new pan-cultural facial expression of emotion. *Motivation and emotion*, 10(2):159–168.

[60] Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11):2765–2781.

[61] Elmadany, N. E. D., He, Y., and Guan, L. (2018). Information fusion for human action recognition via biset/multiset globality locality preserving canonical correlation analysis. *IEEE Transactions on Image Processing*, 27(11):5275–5287.

[62] Evangelidis, G., Singh, G., and Horaud, R. (2014). Skeletal quads: Human action recognition using joint quadruples. In *2014 22nd International Conference on Pattern Recognition*, pages 4513–4518. IEEE.

[63] Fan, X. and Vidal, R. (2006). The space of multibody fundamental matrices: Rank, geometry and projection. In *Dynamical Vision*, pages 1–17. Springer.

[64] Fothergill, S., Mentis, H., Kohli, P., and Nowozin, S. (2012). Instructing people for training gestural interactive systems. In *SIGCHI Conference*, pages 1737–1746.

[65] Fourati, N. and Pelachaud, C. (2016). Perception of emotions and body movement in the emilya database. *IEEE Transactions on Affective Computing*, 9(1):90–101.

[66] Fourati, N., Pelachaud, C., and Darmon, P. (2019). Contribution of temporal and multi-level body cues to emotion classification. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 116–122. IEEE.

[67] Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

[68] Garcia, N. C., Morerio, P., and Murino, V. (2019). Learning with privileged information via adversarial discriminative modality distillation. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2581–2593.

[69] Gaur, U., Zhu, Y., Song, B., and Roy-Chowdhury, A. (2011). A "string of feature graphs" model for recognition of complex activities in natural videos. In *2011 International conference on computer vision*, pages 2595–2602. IEEE.

[70] Gear, C. W. (1998). Multibody grouping from motion images. *IJCV*, 29(2):133–150.

[71] Girdhar, R., Carreira, J., Doersch, C., and Zisserman, A. (2019). Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253.

[72] Global Newswire (2021). Virtual reality (vr) market to reach usd 84.09 billion by 2028. https://www.globenewswire.com/.

[73] Gui, L.-Y., Wang, Y.-X., Liang, X., and Moura, J. M. (2018). Adversarial geometry-aware human motion prediction. In *Proceedings of the european conference on computer vision (ECCV)*, pages 786–803.

[74] Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T., and Ding, R. (2022). Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 762–770.

[75] Gupta, P., Thatipelli, A., Aggarwal, A., Maheshwari, S., Trivedi, N., Das, S., and Sarvadevabhatla, R. K. (2021). Quo vadis, skeleton action recognition? *International Journal of Computer Vision*, 129(7):2097–2112.

[76] Hao, X., Li, J., Guo, Y., Jiang, T., and Yu, M. (2021). Hypergraph neural network for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 30:2263–2275.

[77] Harandi, M., Salzmann, M., and Porikli, F. (2014). Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1010.

[78] Hendrycks, D. and Dietterich, T. (2018). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

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

[80] Holden, D., Saito, J., Komura, T., and Joyce, T. (2015). Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*, pages 1–4. ACM.

[81] Hong, W., Wright, J., Huang, K., and Ma, Y. (2006). Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12):3655–3671.

[82] Hu, H., Lin, Z., Feng, J., and Zhou, J. (2014). Smooth representation clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3834–3841.

[83] Hu, J.-F., Zheng, W.-S., Lai, J., and Zhang, J. (2015). Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5344–5352.

[84] Hu, J.-F., Zheng, W.-S., Ma, L., Wang, G., Lai, J., and Zhang, J. (2018a). Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2568–2583.

[85] Hu, J.-F., Zheng, W.-S., Pan, J., Lai, J., and Zhang, J. (2018b). Deep bilinear learning for rgb-d action recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 335–351.

[86] Huang, Z., Wan, C., Probst, T., and Van Gool, L. (2017). Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6099–6108.

[87] Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Twenty-third international joint conference on artificial intelligence*.

[88] Ji, P., Salzmann, M., and Li, H. (2014). Efficient dense subspace clustering. In *Proceedings of IEEE WACV*, pages 461–468. IEEE.

[89] Ji, P., Zhang, T., Li, H., Salzmann, M., and Reid, I. (2017). Deep subspace clustering networks. In *Advances in Neural Information Processing Systems*, pages 24–33.

[90] Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211.

[91] Junejo, I. N., Dexter, E., Laptev, I., and Perez, P. (2010). View-independent action recognition from temporal self-similarities. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):172–185.

[92] Kacem, A., Daoudi, M., Amor, B. B., Berretti, S., and Alvarez-Paiva, J. C. (2018). A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):1–14.

[93] Karg, M., Samadani, A.-A., Gorbet, R., Kühnlenz, K., Hoey, J., and Kulić, D. (2013). Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Transactions on Affective Computing*, 4(4):341–359.

[94] Ke, Q., An, S., Bennamoun, M., Sohel, F., and Boussaid, F. (2017a). Skeletonnet: Mining deep part features for 3-d action recognition. *IEEE signal processing letters*, 24(6):731–735.

[95] Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. (2017b). A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297.

[96] Ke, Y., Sukthankar, R., and Hebert, M. (2007). Spatio-temporal shape and flow correlation for action recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

[97] Kocabas, M., Athanasiou, N., and Black, M. J. (2020). Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263.

[98] Koniusz, P., Cherian, A., and Porikli, F. (2016). Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European conference on computer vision*, pages 37–53. Springer.

[99] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[100] Kundu, J. N., Gor, M., Uppala, P. K., and Radhakrishnan, V. B. (2019). Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1459–1467. IEEE.

[101] Kwak, S., Han, B., and Han, J. H. (2011). Scenario-based video event recognition by constraint flow. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3345–3352.

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

[103] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[104] Lee, I., Kim, D., Kang, S., and Lee, S. (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1012–1020.

[105] Li, C., Cui, Z., Zheng, W., Xu, C., and Yang, J. (2018a). Spatio-temporal graph convolution for skeleton based action recognition. In *AAAI Conference on Artificial Intelligence*.

[106] Li, C., Hou, Y., Wang, P., and Li, W. (2017a). Joint distance maps based action recognition with convolutional neural networks. *IEEE Signal Processing Letters*, 24(5):624–628.

[107] Li, C., Zhong, Q., Xie, D., and Pu, S. (2017b). Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 597–600. IEEE.

[108] Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. (2018b). Unsupervised learning of view-invariant action representations. *Advances in neural information processing systems*, 31.

[109] Li, L., Wang, M., Ni, B., Wang, H., Yang, J., and Zhang, W. (2021). 3d human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4741–4750.

[110] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3595–3603.

[111] Li, S. and Fu, Y. (2013). Low-rank coding with b-matching constraint for semi-supervised classification. In *International Joint Conference on Artificial Intelligence*.

[112] Li, S. and Fu, Y. (2014). Learning balanced and unbalanced graphs via low-rank coding. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1274–1287.

[113] Li, S., Li, K., and Fu, Y. (2015). Temporal subspace clustering for human motion segmentation. In *Proceedings of IEEE ICCV*, pages 4453–4461.

[114] Li, S., Li, W., Cook, C., Zhu, C., and Gao, Y. (2018c). Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5457–5466.

[115] Li, W., Zhang, Z., and Liu, Z. (2010). Action recognition based on a bag of 3d points. In *Proceedings of IEEE CVPRW*, pages 9–14. IEEE.

[116] Liang, D., Fan, G., Lin, G., Chen, W., Pan, X., and Zhu, H. (2019). Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0.

[117] Lin, L., Song, S., Yang, W., and Liu, J. (2020). Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2490–2498.

[118] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. (2012). Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184.

[119] Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003. IEEE.

[120] Liu, J. and Shah, M. (2008). Learning human actions via information maximization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[121] Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. (2019). Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701.

[122] Liu, J., Shahroudy, A., Xu, D., and Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, pages 816–833. Springer.

[123] Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. (2017a). Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1647–1656.

[124] Liu, M., Liu, H., and Chen, C. (2017b). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362.

[125] Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020). Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152.

[126] Lo Presti, L., La Cascia, M., Sclaroff, S., and Camps, O. (2014). Gesture modeling by hanklet-based hidden markov model. In *Asian Conference on Computer Vision*, pages 529–546. Springer.

[127] Loghmani, M. R., Rovetta, S., and Venture, G. (2017). Emotional intelligence in robots: Recognizing human emotions from daily-life gestures. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1677–1684. IEEE.

[128] Lu, C., Feng, J., Lin, Z., Mei, T., and Yan, S. (2018). Subspace clustering by block diagonal representation. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):487–501.

[129] Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., and Yan, S. (2012). Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360. Springer.

[130] Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

[131] Mallick, T., Das, P. P., and Majumdar, A. K. (2014). Characterizations of noise in kinect depth images: A review. *IEEE Sensors journal*, 14(6):1731–1740.

[132] Markets and Markets (2021). Video surveillance market with covid-19 impact analysis. https://www.marketsandmarkets.com/Market-Reports/videosurveillance-market-645.html.

[133] Martinez, H. P., Yannakakis, G. N., and Hallam, J. (2014). Don't classify ratings of affect; rank them! *IEEE transactions on affective computing*, 5(3):314–326.

[134] Martinez, J., Black, M. J., and Romero, J. (2017). On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900.

[135] Matsumoto, D., Frank, M. G., and Hwang, H. S. (2012). *Nonverbal communication: Science and applications*. Sage Publications.

[136] Mauss, I. B. and Robinson, M. D. (2009). Measures of emotion: A review. *Cognition and emotion*, 23(2):209–237.

[137] Meng, F., Liu, H., Liang, Y., Tu, J., and Liu, M. (2019). Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition. *IEEE Transactions on Image Processing*, 28(11):5281–5295.

[138] Moon, H., Sharma, R., and Jung, N. (2012). Method and system for measuring shopper response to products based on behavior and facial expression. US Patent 8,219,438.

[139] Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. (2007). Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn.

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

[141] Ni, B., Wang, G., and Moulin, P. (2011). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1147–1153. IEEE.

[142] Nie, Q., Liu, Z., and Liu, Y. (2020). Unsupervised human 3d pose representation with viewpoint and pose disentanglement. In *Springer European Conference on Computer Vision (ECCV)*.

[143] Nie, Q., Wang, J., Wang, X., and Liu, Y. (2019). View-invariant human action recognition based on a 3d bio-constrained skeleton model. *IEEE Transactions on Image Processing*, 28(8):3959–3972.

[144] Niewiadomski, R., Hyniewska, S. J., and Pelachaud, C. (2011). Constraint-based model for synthesis of multimodal sequential expressions of emotions. *IEEE Transactions on Affective Computing*, 2(3):134–146.

[145] Niewiadomski, R., Mancini, M., Piana, S., Alborno, P., Volpe, G., and Camurri, A. (2017). Low-intrusive recognition of expressive movement qualities. In *Proceedings of the 19th ACM international conference on multimodal interaction*, pages 230–237.

[146] Noroozi, F., Corneanu, C. A., Kamińska, D., Sapiński, T., Escalera, S., and Anbarjafari, G. (2018). Survey on emotional body gesture recognition. *IEEE transactions on affective computing*, 12(2):505–523.

[147] Ohn-Bar, E. and Trivedi, M. (2013). Joint angles similarities and hog2 for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 465–470.

[148] Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of IEEE CVPR*, pages 716–723.

[149] Ortony, A. (2009). Affect and emotions in intelligent agents: Why and how? In *Affective information processing*, pages 11–21. Springer.

[150] Pang, J. and Cheung, G. (2017). Graph laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Transactions on Image Processing*, 26(4):1770–1785.

[151] Pantic, M. and Rothkrantz, L. J. (2003). Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390.

[152] Paoletti, G., Beyan, C., and Del Bue, A. (2022). Graph laplacian-improved convolutional residual autoencoder for unsupervised human action and emotion recognition. *IEEE Access*, "Accepted on December 2022".

[153] Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2021a). Subspace clustering for action recognition with covariance representations and temporal pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6035–6042. IEEE.

[154] Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2021b). Unsupervised human action recognition with skeletal graph laplacian and self-supervised viewpoints invariance. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*. BMVA.

[155] Park, S. and Aggarwal, J. (2003). Recognition of two-person interactions using a hierarchical bayesian network. In *First ACM SIGMM international workshop on Video surveillance*, pages 65–76.

[156] Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: a review. *Acm sigkdd explorations newsletter*, 6(1):90–105.

[157] Patron-Perez, A., Marszalek, M., Reid, I., and Zisserman, A. (2012). Structured learning of human interactions in tv shows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2441–2453.

[158] Pavllo, D., Feichtenhofer, C., Grangier, D., and Auli, M. (2019). 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762.

[159] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[160] Petrovich, M., Black, M. J., and Varol, G. (2021). Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995.

[161] Piana, S., Staglianò, A., Odone, F., and Camurri, A. (2016). Adaptive body gesture representation for automatic emotion recognition. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(1):1–31.

[162] Picard, R. W. (1997). *Affective Computing*. MIT Press, Cambridge, MA.

[163] Planalp, S. (1996). Varieties of cues to emotion in naturally occurring situations. *Cognition & Emotion*, 10(2):137–154.

[164] Planalp, S. (1999). *Communicating emotion: Social, moral, and cultural processes*. Cambridge University Press.

[165] Pollick, F. E., Paterson, H. M., Bruderlin, A., and Sanford, A. J. (2001). Perceiving affect from arm movement. *Cognition*, 82(2):B51–B61.

[166] Poppe, R. (2010). A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990.

[167] Presti, L., Sclaroff, S., and Rozga, A. (2013). Joint alignment and modeling of correlated behavior streams. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 730–737.

[168] Presti, L. L. and La Cascia, M. (2016). 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130–147.

[169] Rahmani, H. and Bennamoun, M. (2017). Learning action recognition model from depth and skeleton videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5832–5841.

[170] Rahmani, H., Mahmood, A., Huynh, D., and Mian, A. (2016). Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2430–2443.

[171] Rahmani, H., Mahmood, A., Huynh, D. Q., and Mian, A. (2014). Real time action recognition using histograms of depth gradients and random decision forests. In *IEEE winter conference on applications of computer vision*, pages 626–633. IEEE.

[172] Rahmani, H., Mian, A., and Shah, M. (2017). Learning a deep model for human action recognition from novel viewpoints. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):667–681.

[173] Rao, H., Xu, S., Hu, X., Cheng, J., and Hu, B. (2021). Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 569:90–109.

[174] Rehg, J., Abowd, G., Rozga, A., Romero, M., Clements, M., Sclaroff, S., Essa, I., Ousley, O., Li, Y., Kim, C., et al. (2013). Decoding children's social behavior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3414–3421.

[175] Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[176] Roffo, G., Cristani, M., Pollick, F., Segalin, C., and Murino, V. (2013). Statistical analysis of visual attentional patterns for video surveillance. In *Iberoamerican Congress on Pattern Recognition*, pages 520–527. Springer.

[177] Sadanand, S. and Corso, J. J. (2012). Action bank: A high-level representation of activity in video. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 1234–1241. IEEE.

[178] Scherer, K. R. (2005). What are emotions? and how can they be measured? *Social science information*, 44(4):695–729.

[179] Scherer, K. R. (2010). Emotion and emotional competence: conceptual and theoretical issues for modelling agents. *Blueprint for affective computing: A sourcebook*, pages 3–20.

[180] Seidenari, L., Varano, V., Berretti, S., Bimbo, A., and Pala, P. (2013). Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *Proceedings of IEEE CVPRW*, pages 479–485.

[181] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019.

[182] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905.

[183] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019a). Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921.

[184] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019b). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of IEEE CVPR*, pages 12026–12035.

[185] Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2020). Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545.

[186] Shi, Z. and Kim, T.-K. (2017). Learning and refining of privileged information-based rnns for action recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3461–3470.

[187] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee.

[188] Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. (2019). 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*, pages 1227–1236.

[189] Si, C., Jing, Y., Wang, W., Wang, L., and Tan, T. (2018). Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–118.

[190] Smaira, L., Carreira, J., Noland, E., Clancy, E., Wu, A., and Zisserman, A. (2020). A short note on the kinetics-700-2020 human action dataset. *arXiv preprint arXiv:2010.10864*.

[191] Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2018). Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *IEEE Transactions on image processing*, 27(7):3459–3471.

[192] Song, Y., Zhang, Z., Shan, C., and Wang, L. (2021). Efficientgcn: Constructing stronger and faster baselines for skeleton-based action recognition. *arXiv preprint arXiv:2106.15125*.

[193] Soo Kim, T. and Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 20–28.

[194] Su, K., Liu, X., and Shlizerman, E. (2020a). Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640.

[195] Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2020b). Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

[196] Tang, Y., Tian, Y., Lu, J., Li, P., and Zhou, J. (2018). Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5323–5332.

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

[198] Thangali, A., Nash, J., Sclaroff, S., and Neidle, C. (2011). Exploiting phonological constraints for handshape inference in asl video. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 521–528.

[199] The United Nations (2019). World population aging. https://www.un.org/en/development/desa/population/.

[200] Thoker, F. M., Doughty, H., and Snoek, C. G. (2021). Skeleton-contrastive 3d action representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1655–1663.

[201] Tolstikhin, I. O., Sriperumbudur, B. K., and Schölkopf, B. (2016). Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29.

[202] Tracy, J. L. and Randles, D. (2011). Four models of basic emotions: a review of ekman and cordaro, izard, levenson, and panksepp and watt. *Emotion review*, 3(4):397–405.

[203] Tracy, J. L. and Robins, R. W. (2004). Show your pride: Evidence for a discrete emotion expression. *Psychological science*, 15(3):194–197.

[204] Tran, K. N., Gala, A., Kakadiaris, I. A., and Shah, S. K. (2014). Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. *Pattern Recognition Letters*, 44:49–57.

[205] Tuzel, O., Porikli, F., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. In *European conference on computer vision*, pages 589–600. Springer.

[206] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[207] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[208] Veeriah, V., Zhuang, N., and Qi, G.-J. (2015). Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049.

[209] Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595.

[210] Vemulapalli, R. and Chellapa, R. (2016). Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4471–4479.

[211] Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68.

[212] Wang, H. and Wang, L. (2017). Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 499–508.

[213] Wang, H. and Wang, L. (2018). Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection. *IEEE Transactions on Image Processing*, 27(9):4382–4394.

[214] Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297. IEEE.

[215] Wang, L., Huynh, D. Q., and Koniusz, P. (2019a). A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29:15–28.

[216] Wang, L., Zhang, J., Zhou, L., Tang, C., and Li, W. (2015). Beyond covariance: Feature representation with nonlinear kernel matrices. In *Proceedings of the IEEE international conference on computer vision*, pages 4570–4578.

[217] Wang, M., Ni, B., and Yang, X. (2020). Learning multi-view interactional skeleton graph for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[218] Wang, P., Li, W., Li, C., and Hou, Y. (2018). Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 158:43–53.

[219] Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., and Chao, L. S. (2019b). Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822.

[220] Wen, Y.-H., Gao, L., Fu, H., Zhang, F.-L., and Xia, S. (2019). Graph cnns with motif and variable temporal block for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8989–8996.

[221] Wu, C., Wu, X.-J., and Kittler, J. (2019). Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of IEEE ICCVW*, pages 0–0.

[222] Xia, L., Chen, C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *Proceedings of IEEE CVPRW*, pages 20–27. IEEE.

[223] Xing, Y. and Zhu, J. (2021). Deep learning-based action recognition with 3d skeleton: A survey. *CAAI Transactions on Intelligence Technology*, 6(1):80–92.

[224] Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., and Zhang, W. (2020a). Deep kinematics analysis for monocular 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on computer vision and Pattern recognition*, pages 899–908.

[225] Xu, S., Rao, H., Hu, X., Cheng, J., and Hu, B. (2021). Prototypical contrast and reverse prediction: Unsupervised skeleton based action recognition. *IEEE Transactions on Multimedia*.

[226] Xu, S., Rao, H., Peng, H., Jiang, X., Guo, Y., Hu, X., and Hu, B. (2020b). Attention-based multilevel co-occurrence graph convolutional lstm for 3-d action recognition. *IEEE Internet of Things Journal*, 8(21):15990–16001.

[227] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.

[228] Yang, A. Y., Wright, J., Ma, Y., and Sastry, S. S. (2008). Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225.

[229] Yang, D., Li, M. M., Fu, H., Fan, J., and Leung, H. (2020). Centrality graph convolutional networks for skeleton-based action recognition. In *European Conference on Computer Vision (ECCV)*.

[230] Yang, X., Deng, C., Zheng, F., Yan, J., and Liu, W. (2019a). Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4066–4075.

[231] Yang, X. and Tian, Y. (2014). Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 804–811.

[232] Yang, X. and Tian, Y. L. (2012). Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 14–19. IEEE.

[233] Yang, Y., Saleemi, I., and Shah, M. (2012). Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1635–1648.

[234] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

[235] Yi, C., Yang, S., Li, H., Tan, Y.-p., and Kot, A. (2021). Benchmarking the robustness of spatial-temporal models against corruptions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[236] You, C., Li, C.-G., Robinson, D. P., and Vidal, R. (2016a). Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of IEEE CVPR*, pages 3928–3937.

[237] You, C., Robinson, D., and Vidal, R. (2016b). Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of IEEE CVPR*, pages 3918–3927.

[238] Zanfir, M., Leordeanu, M., and Sminchisescu, C. (2013). The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of IEEE ICCV*, pages 2752–2759.

[239] Zhang, H. and Yoshie, O. (2012). Improving human activity recognition using subspace clustering. In *International Conference on Machine Learning and Cybernetics*, volume 3, pages 1058–1063. IEEE.

[240] Zhang, J., Shum, H. P., Han, J., and Shao, L. (2018). Action recognition from arbitrary views using transferable dictionary learning. *IEEE transactions on image processing*, 27(10):4709–4723.

[241] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pages 2117–2126.

[242] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2019). View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1963–1978.

[243] Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., and Zheng, N. (2020a). Semantics-guided neural networks for efficient skeleton-based human action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1112–1121.

[244] Zhang, X., Wang, Y., Gou, M., Sznaier, M., and Camps, O. (2016). Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of IEEE CVPR*, pages 4498–4507.

[245] Zhang, X., Xu, C., and Tao, D. (2020b). Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14333–14342.

[246] Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., and Ding, Z. (2021). 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665.

[247] Zheng, N., Wen, J., Liu, R., Long, L., Dai, J., and Gong, Z. (2018). Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *AAAI Conference on Artificial Intelligence*.

[248] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*.

[249] Zunino, A., Cavazza, J., Volpi, R., Morerio, P., Cavallo, A., Becchio, C., and Murino, V. (2020). Predicting intentions from motion: The subject-adversarial adaptation approach. *International Journal of Computer Vision*, 128(1):220–239.