OPEN ACCESS

University of BRISTOL

This electronic thesis or dissertation has been downloaded from Explore Bristol Research, http://research-information.bristol.ac.uk

Author:
Dadashzadeh, Amirhossein

Title:
Learning Strategies for Parkinson's Disease Severity Assessment

General rights
Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy
Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

•Your contact details
•Bibliographic details for the item, including a URL
•An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Learning Strategies for Parkinson's Disease Severity Assessment

Amirhossein Dadashzadeh

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering

February 5, 2024

43560 words

Abstract

Parkinson's disease (PD) is a common neurodegenerative disorder that affects millions of people worldwide. Current clinical assessments of PD symptoms require trained raters and are subjective. Computer vision and machine learning can be used to automate PD assessments, reducing the reliance on trained raters and introducing a more objective measure. However, complexity of human movements, subtle motion differences, and scarcity of annotated data present challenges that this thesis attempts to address by developing novel deep learning frameworks to predict PD severity in videos.

First, to assess PD using RGB data, we propose an end-to-end model, built on a temporal segment framework to capture both spatial and long-term temporal structures. We enhance the performance of our model by incorporating a temporal attention mechanism. Motion boundaries are also explored as an extra input modality to assist in obfuscating the effects of camera motion. We evaluate this method on the PD2T dataset, which includes two PD motor function tasks performed by actual patients. Our results suggest that a deep learning-based approach to assess PD from only RGB data is not only feasible, but also effective.

Next, in response to the scarcity of annotated videos, we focus on self-supervised learning (SSL). Unlike traditional SSL methods which struggle with small pretraining data, our approach leverages an auxiliary pretraining phase with knowledge similarity distillation, enabling improved generalisation with significantly less data. We further introduce a novel SSL pretext task, Video Segment Pace Prediction or VSPP, to provide more reliable self-supervised representation. Our SSL framework shows state of the art performance on UCF101 and HMDB50 datasets under a low-data regime. Furthermore, this approach outperforms fully-supervised pretraining when evaluated on a new PD dataset (PD4T), which includes four different PD motor tasks.

Finally, this thesis presents a novel, parameter-efficient, continual pretraining workflow (PECoP) that significantly improves upon conventional fine-tuning techniques. Its primary objective is to enhance the transfer of knowledge gained from existing large-scale video datasets to AQA target tasks by updating only a small number of parameters in additional bottleneck layers (called 3D-Adapters) through self-supervised learning. Evaluating our method on PD4T, and three public AQA benchmarks (JIGSAWS, MTL-AQA, FineDiving), we show that PECoP can boost the robustness of recent state of the art AQA methods, by a considerable margin.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original, except where indicated by special reference in the text, and no part of the dissertation has been submitted for any other academic award.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED: DATE:

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Majid Mirmehdi and Dr. Alan Whone. Their constant support and great advice have been key to my research journey. I am also very thankful to my annual reviewer, Prof. Dima Damen, for her helpful feedback and suggestions.

I am grateful to all my colleagues in the ViLab[1] and MaVi[2] research groups: Abel, Arindam, Akinobu, Burak, Chen, Danier, Gabriele, Gavryel, Hanyuan, Jakub, Jing, Lama, Obed, Richard, Saptarshi, Simon, Shuchao, Spike, Shijia, Siddhant, Sasha, Xin, Will, and Zhifan. Their friendship and support have created a welcoming and inspiring environment that I have been fortunate to be a part of.

Lastly, I must give special thanks to the generous donors at the Southmead Hospital Charity, particularly to Caroline Belcher. Her support has not just helped my research move forward but has actually made it possible for me to pursue my Ph.D.

[1] Visual Information Laboratory
[2] Machine Learning and Computer Vision

Publications

The work described in this thesis has been presented in the following publications:

1. **Amirhossein Dadashzadeh**, Alan Whone, Michal Rolinski, and Majid Mirmehdi. Exploring motion boundaries in an end-to-end network for vision-based Parkinson's severity assessment. *10th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2021.

2. **Amirhossein Dadashzadeh**, Alan Whone, and Majid Mirmehdi. Auxiliary Learning for Self-Supervised Video Representation via Similarity-based Knowledge Distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

3. **Amirhossein Dadashzadeh**, Shuchao Duan, Alan Whone, and Majid Mirmehdi. PECoP: Parameter Efficient Continual Pretraining for Action Quality Assessment. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

To My Parents

Contents

List of Figures

List of Tables

Acronyms

A-VID Appearance Video Instance Discrimination. 28

ACP Appearance Consistency Perception. 28

AQA Action Quality Assessment. 4, 9

AT Adapter Tuning. 36

CL Contrastive Learning. 25

CNN Convolutional Neural Network. 5

CoRe Group-aware Contrastive Regression. 12

DCT Discrete Cosine Transform. 10

GART Group-aware Regression Tree. 13, 72

GCN Graph Convolutional Network. 20

GMM Gaussian Mixture Model. 18

HMM Hidden Markov Model. 18

KD Knowledge Distillation. 32

KDE Kernel Density Estimation. 18

KL Kullback-Leibler divergence. iv, 12

MAXL Meta Auxiliary Learning. 31

MUSDL Multi-Path Uncertainty-Aware Score Distributions Learning. 12

NLP Natural Language Processing. 36, 77

PD Parkinson's disease. 1, 9, 95

PECoP Parameter-Efficient Continual Pretraining. 78

PETL Parameter-Efficient Transfer Learning. 36

PT Prompt Tuning. 36

RKD Relational Knowledge Distillation. 33

RNN Recurrent Neural Network. 53

RSP Relative Speed Perception. 27

SCP Speed Consistency Perception. 28

SKD Similarity-based Knowledge Distillation. 33

SSKD Self-Supervised Knowledge Distillation. 32

SSL Self-Supervised Learning. 3, 6, 7, 9, 55

SVR Support Vector Regression. 10

TCNN Temporal Convolutional Neural Network. 19

TPT Temporal Parsing Transforme. 14

TSA Temporal Segmentation Attention. 14

UPDRS Unified Parkinson's Disease Rating Scale. 2, 39, 70

USDL Uncertainty-Aware Score Distribution Learning. 12

VPT Visual Prompt Tuning. 36

VSPP Video Segment Pace Prediction. 57

Chapter 1

Introduction

Parkinson's disease (PD) is the second most common neurodegenerative disorder after Alzheimer's dementia [126]. It affects around 10 million people worldwide, and is slightly more prevalent in males [1]. PD involves the substantial reduction of dopamine-producing neurons, particularly in the substantia nigra region, which is responsible for the control of body movement [120]. The characteristic motor features of this condition include slowness of movement (bradykinesia), stiffness (rigidity), tremor, and postural instability [173]. These symptoms create significant barriers for patients in their daily activities, from simple household duties to more complicated tasks such as driving, which significantly reduce the quality of their lives.

Regular clinical assessment and close monitoring of the signs and symptoms of PD are required to tailor symptomatic treatments and optimise disease control. Further, accurate quantification of disease progression is crucial in the design and efficacy testing of any drugs or interventions that are aimed at modifying or improving the course of the condition, as it provides valuable data for longitudinal studies.

Assessment of motor symptoms in PD patients is usually performed in controlled clinical settings. Here, the focus is mainly on evaluating the degree of rigidity and bradykinesia. Typically, the patient is asked to perform an elaborate series of specific physical tasks. These tasks may include actions such as opening and closing their hand in rapid succession, i.e., gripping and letting go, or walking at their usual pace for several meters, and so on. The tasks are then closely monitored and evaluated by a PD physician or a specifically-trained nurse who makes a professional evaluation based on their experience in clinical practice.

In more formal research or experimental settings, such as drug trials or academic research

Table 1.1: *UPDRS Scoring for Gait. This table provides a standardised framework for evaluating gait severity levels in individuals with PD.*

Score	Description
0	Normal: No problems.
1	Slight: Independent walking with minor gait impairment.
2	Mild: Independent walking but with substantial gait impairment.
3	Moderate: Requires an assistance device for safe walking (walking stick, walker) but not a person.
4	Severe: Cannot walk at all or only with another person's assistance.

studies, the clinical assessment is usually scored using a globally recognised scale known as the Unified Parkinson's Disease Rating Scale (UPDRS) [45]. This comprehensive rating system consists of 33 separate examiner-defined tests aimed at offering a granular evaluation of motor functions. An example of UPDRS scoring for the gait task is shown in Table 1.1. Clinicians typically quantify the severity of each action by assigning a numerical score that ranges from 0 (normal) to 4 (most severe). However, such a process of assessment and scoring is highly subjective and necessitates the expense of an available rater trained in PD assessment. Therefore, automating the PD assessment process may assist in eliminating these shortcomings by offering a more objective and potentially real-time method of evaluation.

In recent years, several techniques have been explored for the automated measurement of symptoms related to PD [40, 61, 63, 99, 101, 132]. These have often relied on wearable sensors, which can be both costly and intrusive. As an alternative, video technology provides a non-intrusive and scalable solution for detecting and quantifying these symptoms. Advances in deep learning and high-performance computing now allow for more precise analysis of human movements through video [80, 97, 114, 138, 166]. In video-based PD assessment, most of the current research have focused on methods that employ skeleton data to analyse human motion [48, 92, 96, 97, 125]. While promising, these skeleton-based approaches come with their own challenges. First, they often require a preprocessing step to extract meaningful skeletal features, adding extra computational cost to the assessment process. Second, skeleton data may not always accurately capture the subtle movements exhibited by PD patients, especially those movements that are crucial for an accurate severity assessment. For example, when assessing gait based on the UPDRS, crucial factors like stride amplitude, stride speed, height of foot lift, and quality of heel strike during walking are considered. Skeleton data may outline basic

movement trajectories but lack the granularity to capture these critical elements, which are important for distinguishing between mild and moderate cases.

Given these limitations, the focus of this thesis will be on exploring the potential of RGB-based video data for assessing PD severity through developing innovative deep learning strategies tailored for this task. These strategies are categorised as follows:

i) End-to-end supervised learning, with a focus on motion analysis for PD severity assessment; ii) Self-supervised representation learning to reduce dependency on large-scale annotated datasets; iii) Parameter-efficient continual pretraining to enhance model adaptability by updating a few bottleneck layers through Self-Supervised Learning (SSL).

While these strategies are validated extensively on PD tasks, their versatility extends to other domains within computer vision. Additional experiments confirm this by applying the methods to a range of tasks including action recognition and other AQA tasks, such as diving and surgical skill assessment.

1.1 Challenges

In this thesis, we will explore four main challenges associated with applying deep learning to video-based PD severity assessment.

Challenge 1: Complexity of Human Movements

One of the main challenges in this area is the complexity of human movements, an issue that becomes even more challenging when dealing with the symptoms of PD. Examining the gait task, it is evident that even this basic action demonstrates a variety of arm swings, stride lengths, and walking speeds among healthy people. These characteristics are not only unique to each person but can also change based on factors such as age, mood, or even the type of shoes they are wearing. When it comes to assessing the severity of PD, this complexity is amplified. PD is associated with specific gait alterations like shuffling steps, a decrease in arm swing, or more severe symptoms such as freezing, where the patient is unable to take the next step despite the intention to continue walking. These subtle differences in motion, ranging from small changes in stride to episodes of freezing, present a significant challenge for automated assessment of PD severity. The challenge lies in designing deep learning models that are capable of capturing both the spatial and temporal features of a given task, such as gait, while also being sensitive enough to detect subtle irregularities representative of different levels of PD severity.

Challenge 2: Presence of Camera Motion

The use of video data for clinical studies, such as the assessment of PD, can be hindered by the introduction of variability caused by camera motions. For instance, when evaluating gait, a camera that follows the subject can add 'motion noise' to the video data, making it hard to differentiate between the subject's actual movements and those caused by the camera's motion. This extra layer of complexity can disrupt vision-based models that are meant to measure the severity of symptoms, such as stride irregularities or freezing episodes, leading to less accurate results.

Challenge 3: Scarcity of Annotated Data

Deep learning models are known for their exceptional performance but require a large amount of annotated data to reach that level of accuracy. In a clinical setting, particularly when dealing with PD, gathering this kind of extensive dataset presents several challenges. The first challenge is the recording process itself, where patients are required to perform specific motor tasks such as finger tapping or leg agility. These tasks can be difficult and emotionally stressful for PD patients, complicating and lengthening the data collection process. Furthermore, capturing these tasks on video often requires specialised equipment and controlled environments to ensure high-quality data. Once the data is collected, the next step is annotation. Medical experts need to carefully analyse these videos to provide labels or annotations that serve as the ground truth for deep learning models. This step is not only time-consuming but also costly, as it often requires the expertise of highly trained clinicians.

Beyond these challenges, there are also ethical and legal considerations. Concerns about patient privacy, data security, and informed consent add another layer of complexity to the data collection process. Obtaining the necessary approvals and authorisations from participants, and ensuring the sensitive data is stored and handled safely can take a long time and involve a lot of paperwork. All these factors contribute to the difficulty in collecting large-scale, high-quality annotated data for PD assessment tasks. This limitation restricts deep learning models' ability to generalise effectively, affecting their performance on new patient data or different stages of the disease.

Challenge 4: Transfer of Knowledge to AQA Tasks

One common solution to address the scarcity of annotated data in Action Quality Assessment (AQA), particularly for tasks related to PD, involves initialising a model with weights pretrained on more generic, large-scale datasets, e.g. Kinetics-400 [71]. Although better than pretraining from scratch, this approach presents challenges. As illustrated in Figure 1.1, samples from different classes in a generic dataset used for action recognition

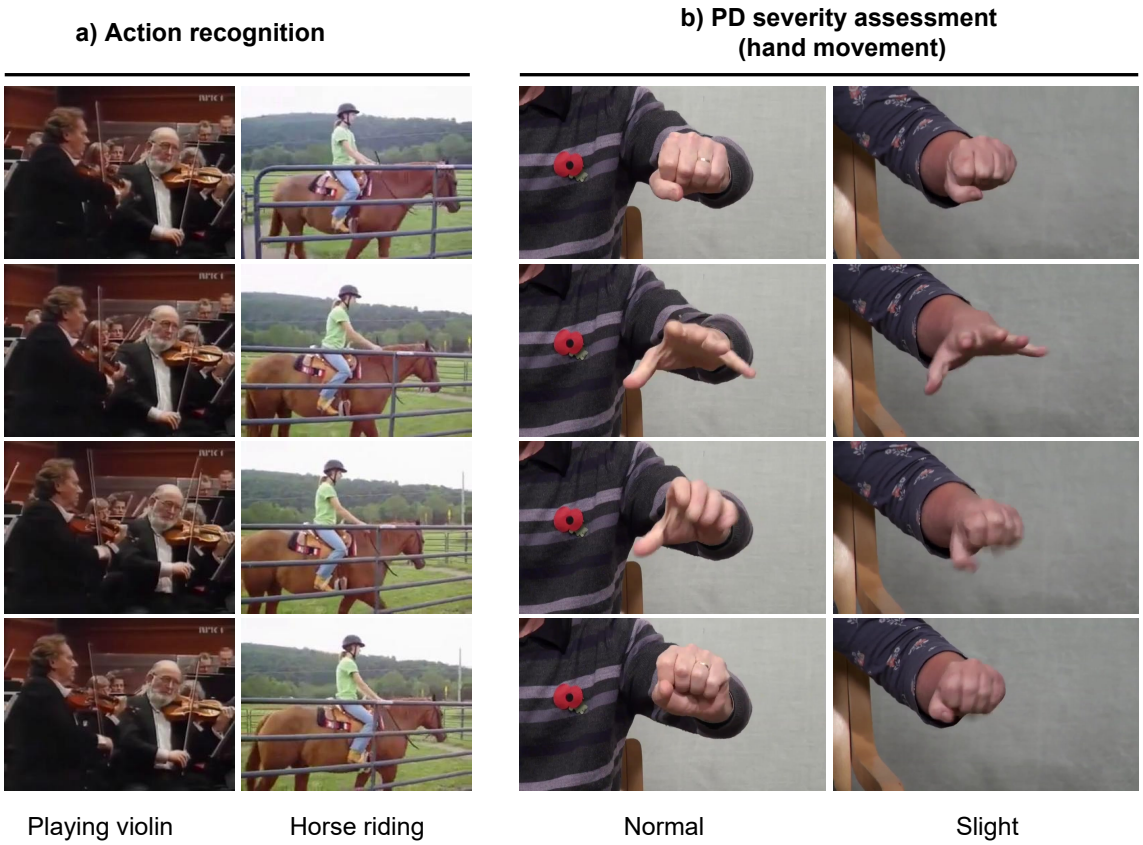


Figure 1.1: *Comparison of inter-class variances in action recognition and PD severity assessment (hand moveent task). For action recognition, classes like 'horse riding' and 'playing violin' exhibit clear temporal and spatial distinctions. In contrast, PD samples from different severity levels (e.g., Normal and Slight) demonstrate subtler differences, highlighting the intricate nature of PD severity assessment over action recognition.*

task display distinct temporal and spatial patterns. However, these may not align well with the specific needs of diagnosing or assessing PD severity. The nuanced differences in hand movements, particularly between Normal and Slight PD severity levels, highlight this mismatch. Therefore, there is a need to better adapt the knowledge gained from existing large-scale video datasets to PD tasks.

1.2 Contributions and Publications

1.2.1 Contributions

The main contributions of this thesis can be summarized as follows:

- A novel, end-to-end Convolutional Neural Network (CNN) architecture is proposed for evaluating the severity of PD motor states using only RGB video data aligned with the UPDRS. A distinctive feature of this proposed method is the integration

of motion boundaries to counteract camera motion effects, thus enhancing the accuracy of video-based PD severity prediction.

- An auxiliary pretraining stage based on similarity-based knowledge distillation is introduced to alleviate the dependency on large-scale generic datasets in self-supervised video representation learning, allowing for the use of PD target datasets instead of Kinetics-400.
- A simple, yet effective and novel SSL pretext task is presented which is more commensurate with video motion events than existing pretext tasks.
- A parameter efficient continual pretraining workflow is presented to better transfer the knowledge learned from existing large-scale video datasets to PD target tasks by only updating a small number of additional bottleneck layers (called 3D-Adapters) through SSL.
- A new AQA dataset, PD4T, is introduced, which includes 2,931 videos from 30 PD patients performing four key motor tasks: gait, hand movement, finger tapping, and leg agility. The dataset, captured at 25fps and clinically scored, aims to serve as a robust benchmark for the vision community.

1.2.2 Publications

The research detailed in this thesis has resulted in the following publications:

- **Amirhossein Dadashzadeh**, Alan Whone, Michal Rolinski, and Majid Mirmehdi. Exploring motion boundaries in an end-to-end network for vision-based Parkinson's severity assessment. *10th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2021.(Chapter 3)
- **Amirhossein Dadashzadeh**, Alan Whone, and Majid Mirmehdi. Auxiliary Learning for Self-Supervised Video Representation via Similarity-based Knowledge Distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022. (Chapter 4)
- **Amirhossein Dadashzadeh**, Shuchao Duan, Alan Whone, and Majid Mirmehdi. PECoP: Parameter Efficient Continual Pretraining for Action Quality Assessment. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. (Chapter 5)

1.3 Thesis Overview

This thesis is presented in 6 main chapters. Following this introduction,

- In Chapter 2, we provide a comprehensive survey of literature relevant to our research on PD severity assessment. We begin with an overview of AQA, covering its applications in both sports and healthcare. We then explore works in advanced learning strategies such as SSL, knowledge distillation, continual pretraining, and parameter-efficient transfer learning, all of which are relevant to our study.
- In Chapter 3, we present an end-to-end multi-stream deep learning configuration, adapting 3D CNNs to efficiently capture spatial and temporal features from RGB, optical flow, and motion boundaries. A sparse temporal sampling strategy enables capturing long-range temporal features, while attention units allow for focused analysis on key video segments. This method is then evaluated on two distinct PD tasks: hand movement and gait, showing its versatility and applicability.
- In Chapter 4, we tackle the limitations of self-supervised pretraining methods, particularly their poor generalisation capabilities when faced with small unlabeled datasets or significant domain shifts. To address these issues, we introduce a novel auxiliary pretraining phase, which employs a teacher-student framework for knowledge similarity distillation. We also propose a new pretext task, video segment pace prediction, to provide more reliable self-supervised representations. We then evaluate our proposed SSL framework on action recognition using widely recognised benchmarks UCF101 [134] and HMDB101 [75], as well as our newly introduced PD4T dataset, including real patients performing actions such as gait, finger tapping, hand movement, and leg agility.
- In Chapter 5, we present parameter-efficient continual pretraining, an innovative stage in the AQA transfer learning pipeline. We introduce 3D-Adapter, a lightweight bottleneck block inserted into pretrained 3D CNN architectures. The adapter fine-tunes for domain-specific spatiotemporal features via SSL while freezing the original model weights. This approach minimises computational costs and storage needs while effectively dealing with overfitting and catastrophic forgetting commonly encountered in continual learning scenarios. In this chapter, we evaluate our method on three public AQA benchmarks: MTL-AQA [114], JIGSAWS [42], and FineDiving [159], while also providing comparative results on the PD4T dataset.
- In Chapter 6, we start by providing a detailed review of the main objectives and

the significant contributions presented throughout this thesis. Next, we critically evaluate our major findings, underlining their strengths and limitations. Finally, the chapter suggests potential directions and avenues that future research could explore, building upon the groundwork laid in this study.

Chapter 2

Background

This chapter provides a background of the literature relevant to the research explored in this thesis. Section 2.1 offers an overview of AQA, exploring its applications in various domains such as sports and healthcare, but with a specific focus on assessing the severity of PD. Section 2.2 delves into SSL, highlighting various pretext tasks and contrastive learning methods and their application in AQA. Following this, Section 2.3 introduces auxiliary learning that enhances self-supervised pretraining performance by leveraging additional information. In Section 2.4, the focus shifts to knowledge distillation, with particular emphasis on similarity-based methods. Section 2.5 explores continual pretraining, another avenue for improving the generalisation capabilities of deep learning models, especially those employing SSL techniques. Next, Section 2.6 discusses parameter-efficient transfer learning, exploring methods that enable effective knowledge transfer with fewer parameters.

2.1 Action Quality Assessment (AQA)

In recent years, the field of AQA has gained increasing attention due to its vital role in a variety of real-world applications, such as sport event analysis [8, 115, 119, 138, 166], healthcare, physical rehabilitation [3, 11, 33, 96], skill assessment [30, 89], and more. Unlike action recognition, which identifies the type of action, AQA evaluates the quality of its execution. This involves focusing on finer details like slight variations in posture, timing, and fluidity, making AQA a more complex yet invaluable task than merely classifying or labeling an action. In this section, we explore in detail the various aspects of AQA, discussing its role in sports scoring, skill assessment, and in healthcare, specifically with regard to the evaluation of the severity of PD.

2.1.1 AQA for Sports Scoring

In the domain of sports, AQA mainly falls into two primary methodologies based on the type of input data: pose-based methods [46, 119], which focus on the study of body positions, and appearance-based methods [8, 113, 114, 124, 138, 166], which rely on visual elements captured through cameras. In early studies on pose-based AQA [46, 119], the process is usually divided into three main steps . First, the system tracks the location of key body parts such as hands and feet. Next, it gathers important features like position, speed, and direction from these tracked points. Finally, a score or grade for the quality of the action is calculated using either set rules or machine learning methods. For example, Pirsiavash et al. [119] introduce a regression-based method designed for the evaluation of action quality in Olympic sports, such as diving and figure skating. They initially capture the athletes' body poses, which are then encoded using the Discrete Cosine Transform (DCT). These transformed pose features serve as the input to a Support Vector Regression (SVR) model that subsequently predicts action quality scores. Their system not only predicts performance scores, but also generates constructive feedback for athletes, advising them on the specific body movements that could enhance their overall performance. Feedback examples for diving and figure skating actions generated by this work are visually presented in Figure 2.1. However, obtaining accurate pose data in the

Figure 2.1: *Feedback examples for divers in the first and second rows, and for skaters in the third and fourth rows. The red vectors guide the divers/skaters by indicating the direction in which they should move their bodies. The figure is adapted from [119].*

sport domain is difficult [150]. Often, the gathered information is incomplete due to the

athlete's body adopting complex positions, or because certain body parts are hidden from view. For example, when a gymnast flips, the arms and legs may overlap, making it difficult to get accurate data that can subsequently affect the final performance score. Moreover, focusing on pose data overlooks crucial visual elements; in the case of diving, for example, the size and shape of the water splash are essential for scoring but are not captured by pose-based methods.

Due to these limitations of pose-based methods, researchers have increasingly shifted their focus to appearance-based approaches [8, 113, 114, 124, 138, 166], which offer the advantage of utilising both spatial and temporal visual features. These methods have achieved considerable success in recent years.

As an early appearance-based approach, Parmar et al. [115] evaluate three different frameworks to assess athletic performance in sports such as diving, gymnastics vaulting, and figure skating. Initially, each framework employs a C3D [142] model to analyse 16-frame, non-overlapping segments of video, capturing spatio-temporal features. The subsequent phases involve various techniques for feature aggregation and score prediction. The first framework averages these features and employs a Support Vector Regression (SVR) for final score estimation. The second uses Long Short-Term Memory (LSTM) networks to capture long-term sequential patterns, while the third integrates both LSTM and SVR for a more comprehensive assessment.

In their later work, Parmar et al. [114] improve the scoring performance and the generalisation ability of their AQA model by employing multi-task learning, which integrates two additional tasks, action recognition and commentary generation, alongside the primary task of action score prediction. To capture features from various segments of a video, a C3D network is utilised. For the tasks of action assessment and recognition, these features are then averaged to create a unified video-level representation. In contrast, for the commentary generation task, a sequence-to-sequence approach is adopted, and the extracted features are individually input into its corresponding branch. Meanwhile, they introduced a new multi-task AQA dataset, named MTL-AQA, which consists of 1,412 diving samples to evaluate their approach.

Pan et al. [109] introduce a graph-based joint relation model for a more fine-grained analysis. This work emphasizes the importance of the interactions between neighbouring joints for accurately assessing actions. The study introduces two specialised graphs, one for spatial relationships and another for temporal dynamics, alongside two innovative modules: the joint commonality module and the joint difference module. These modules facilitate the understanding of general motions and differences in motion within body

parts.

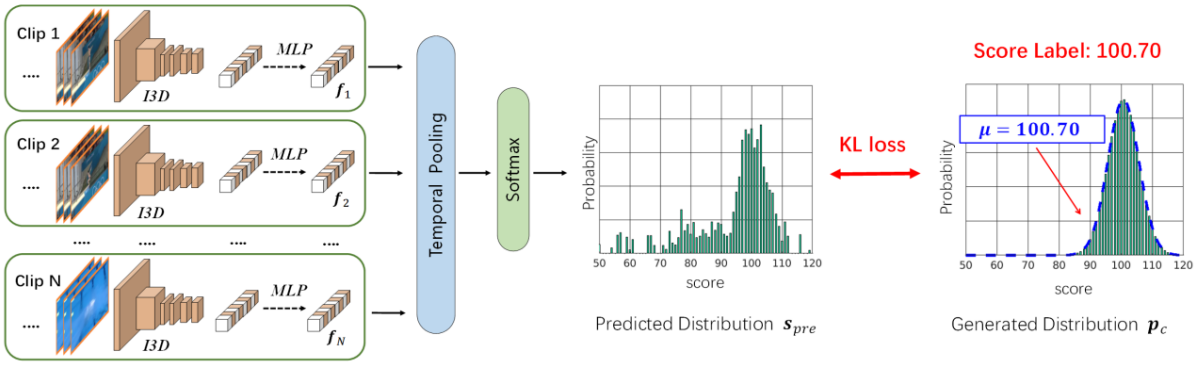


Figure 2.2: *Pipeline of uncertainty-aware score distribution learning, proposed in [138]. First, video frames are split into N segments and then processed using an I3D backbone [10] for feature extraction. Then, the extracted features pass through three fully-connected layers, get fused by temporal pooling, and are sent through a softmax layer to generate the predicted distribution. Finally, the KL loss between this predicted distribution and a Gaussian distribution derived from the score labels is optimised. The figure is adapted from [138].*

In previous AQA methods, regression algorithms are commonly used to predict action scores based on video data. These traditional approaches, however, neglect the inherent uncertainty and ambiguity present in score labels, often due to multiple judges or subjective evaluations. Addressing this limitation, Tang et al. [138] propose Uncertainty-Aware Score Distribution Learning (USDL), which differs from traditional methods by modeling an action's quality through a distribution of possible scores to better deal with the uncertainties often found in assessing action quality. The pipeline of USDL is illustrated in Figure 2.2.

For cases where fine-grained score labels are available, such as difficulty levels or multiple judges' scores, the authors extend USDL into a Multi-Path Uncertainty-Aware Score Distributions Learning (MUSDL) method. MUSDL disentangles the various components contributing to the final score, offering a more comprehensive handling of the inherent uncertainties in AQA evaluations. Both the USDL and MUSDL frameworks leverage KL to optimise the predicted score distribution against a Gaussian distribution that is generated from the real score labels. The architecture of MUSDL is shown in Figure 2.3.

Yu et al. [166] propose Group-aware Contrastive Regression (CoRe) framework to emphasise the importance of pairwise comparison of videos in assessing action quality. This is achieved by introducing a reference video, which belongs to the same category as the video under assessment and acts as a standard for comparing quality. CoRe uses a

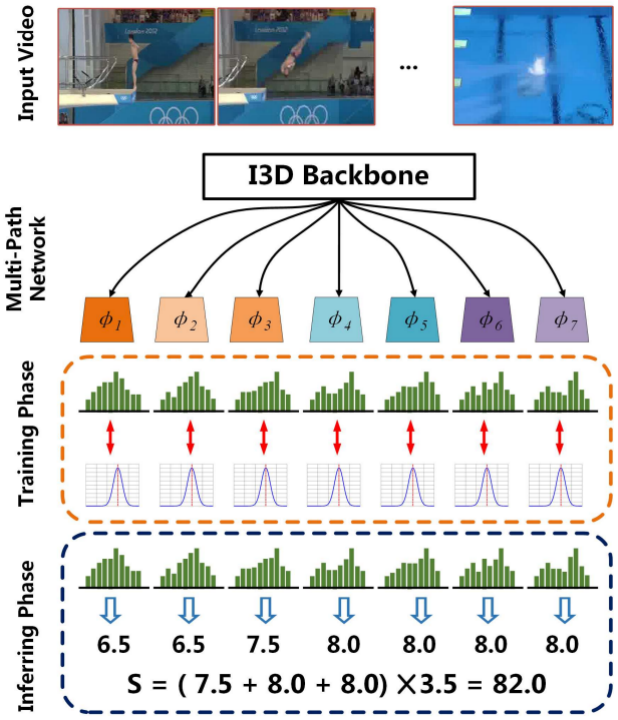


Figure 2.3: *Overview of MUSDL [138]. During the training phase, scores from K (e.g. K=7) judges are modeled as distinct Gaussian distributions, and a similar strategy is employed to train a model comprised of K sub-networks. In the testing phase, the final assessment is derived from the K predicted scores and the rule of the game. The figure is adapted from [138].*

hierarchical approach with a Group-aware Regression Tree (GART) that helps divide the task into smaller and easier-to-handle parts. Specifically, GART divides the relative score into several non-overlapping intervals, referred to as groups. A binary tree is then employed to progressively assign the relative score to one of these predefined groups. Subsequently, regression is performed within the group where the relative score is situated to predict the final score. To optimise the model, CoRe employs a unique objective function that combines classification and regression tasks. The classification part ensures that video pairs are accurately sorted into predefined groups, while the regression part refines the score predictions within those groups. An overall view of CoRe is shown in Figure 2.4.

Xu et al. [159] argue that understanding both high-level semantics and internal temporal aspects of actions in competitive sports is essential for accurate and interpretable AQA. To support this claim, they present FineDiving, a fine-grained dataset developed for various diving events. Unlike existing datasets, FineDiving includes detailed annotations on action procedures, thereby facilitating a more reliable and transparent approach to

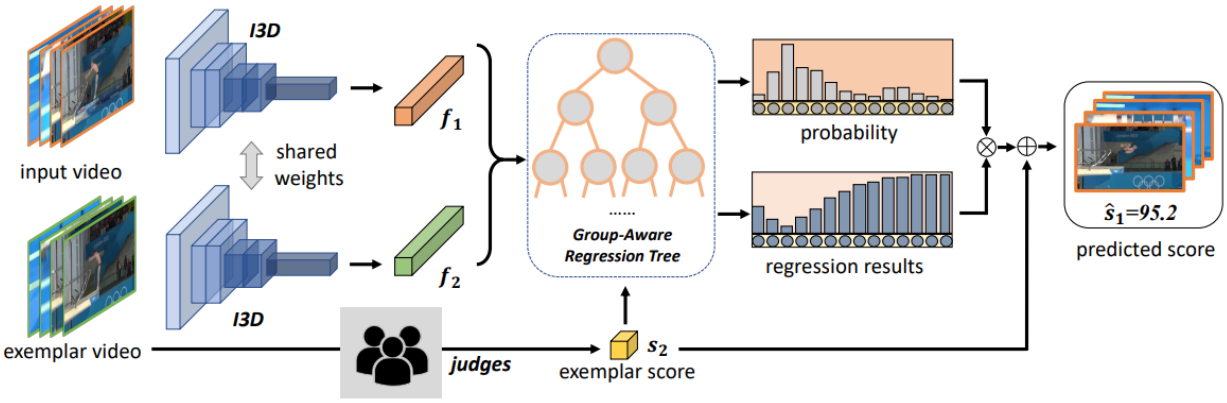


Figure 2.4: *The pipeline of group-aware contrastive regression method (CoRe [166]). First each input video is paired with an exemplar video. This video pair is then fed through the shared I3D backbone to extract spatio-temporal features, which are then combined with the reference score of the exemplar video. This combined feature set is sent through a group-aware regression tree to obtain the relative quality score between the input and the exemplar. In the inference phase, this process is repeated with multiple exemplars for a more robust final quality score for the input video, achieved by averaging the relative scores. The figure is adapted from [166].*

scoring in AQA. To exploit this fine-grained dataset, the authors introduce a procedure-aware method for AQA using a novel Temporal Segmentation Attention (TSA) module. Instead of traditional methods, their approach decomposes pairwise query and exemplar action instances into consecutive steps to capture diverse correspondences. The TSA module employs a procedure-aware cross-attention mechanism to learn embeddings and find semantic, spatial, and temporal matches between the query and exemplar actions. The method then performs fine-grained contrastive regression on these embeddings to derive a reliable scoring mechanism that quantifies step-wise quality differences between query and exemplar actions. The overall framework of TSA is illustrated in Figure 2.5. It is worth noting that TSA can only be applied to those AQA datasets that contain fine-grained scores (e.g. FineDiving).

To more effectively capture fine-grained intra-class variations in AQA tasks, Bai et al. [8] presents a novel framework that utilises Temporal Parsing Transforme (TPT). This framework decomposes holistic features into temporal part-level representations. Particularly, the model employs learnable queries to focus on atomic temporal patterns, thus allowing for the capture of key phases in actions. For instance, in the case of a diving action, these key phases could include the approach, take-off, and flight. This level of specificity contributes to more accurate quality assessments. To calculate the quality score, the authors take advantage of the contrastive regression framework proposed in [166], which they apply to the temporally ordered part representations. The use of con-

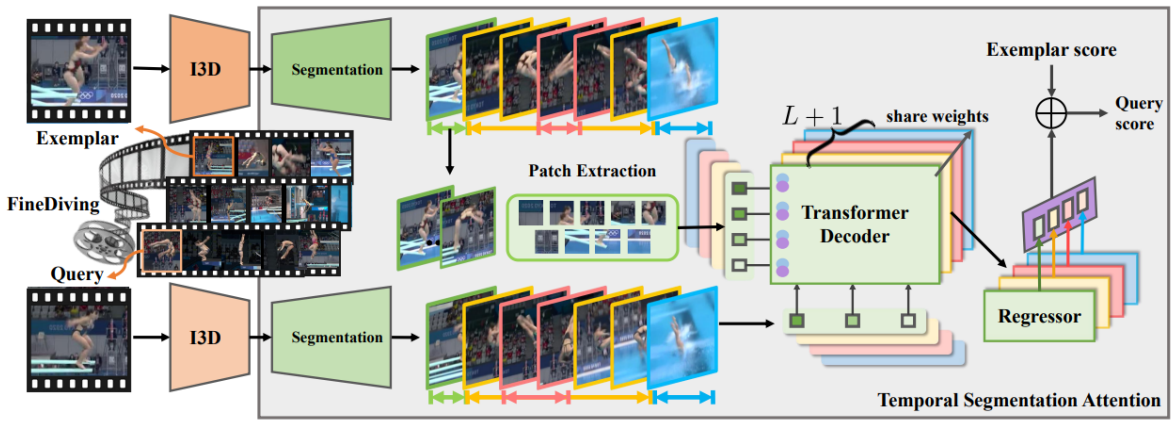


Figure 2.5: *The architecture of the procedure-aware action quality assessment proposed in [159]. With the use of pairwise query and exemplar instances, the method leverages I3D to capture spatial-temporal visual features. To evaluate action quality, a temporal segmentation attention module is introduced. This module performs tasks in a sequence: it first segments the action procedure, then engages in procedure-aware cross-attention learning, and finally carries out fine-grained contrastive regression. The temporal segmentation attention module is trained using step transition labels and action score labels. The figure is adapted from [159].*

trastive regression allows the model to differentiate quality more effectively based on these part-level features. Addressing the absence of temporal part-level labels in most of AQA datasets, the authors introduce two innovative loss functions. The first is a ranking loss that operates on the cross-attention responses of the transformer's decoder. This ensures that the learnable queries are aligned with the inherent temporal order of the action's phases. The second is a sparsity loss designed to make the part representations more discriminative, thereby enhancing the model's ability to differentiate between subtle variations in quality. The architecture of TPT is shown in Figure 2.6. Please note that the aforementioned AQA works (e.g. [8, 138, 159, 166]), detailed in this section, have been used as baselines in subsequent chapters of this thesis.

In summary, most of the state-of-the-art methods in assessing the quality of sport actions have demonstrated impressive performance on a variety of benchmarks. However, it is important to note that these approaches often rely on initialising their backbone model with large-scale datasets like Kinetics-400 [71]. This heavy reliance on large-scale datasets raises questions about the models' generalisability and transferability. In particular, the performance gains might not solely come from the method's effectiveness in AQA tasks. Instead, they could be a byproduct of features learned from a dataset that is quite different from the target AQA dataset. This highlights the need for methods that are effective and versatile across a range of domains. In Chapters 5, we address this issue by incorporating target AQA videos during the pretraining stage. Table 2.1 provides a

summary of the learning-based approaches proposed for sports assessment.

It is worth noting that, in Chapters 4 and 5, the AQA methods detailed above, including MUSDL [138], CoRe [166], TSA [159], and TPT [8], are leveraged as baselines, allowing for a comprehensive comparison with the approaches proposed in this thesis.

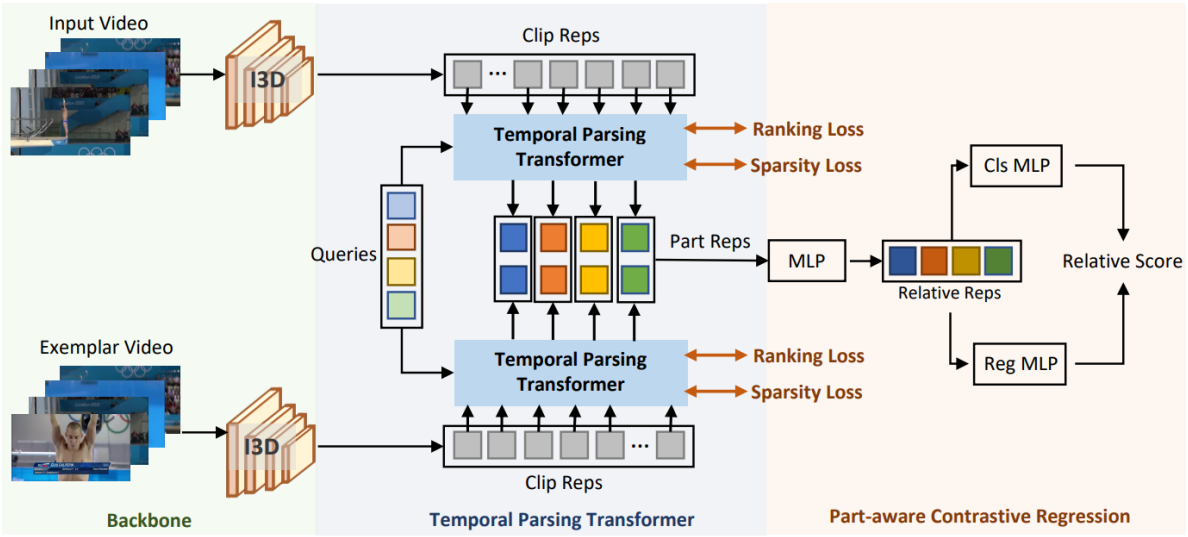


Figure 2.6: *Overview of TPT [8]. Clip-level representations are transformed into part-level temporal representations by temporal parsing transformer. Part-wise relative representations are initially calculated and subsequently fused to estimate the relative score by the part-aware contrastive regressor. A group-aware regression strategy is utilised following previous work [166]. During the training phase, the learning of part representations is guided by the employment of ranking loss and sparsity loss on decoder cross-attention maps. The figure is adapted from [8].*

2.1.2 AQA for Healthcare

In this subsection, we delve into the application of AQA within the healthcare domain, with a particular emphasis on studies related to the assessment of Parkinson's disease severity. This area has predominantly utilised wearable sensors as a means for data collection and analysis, as evidenced by several significant studies [63, 99, 101, 126, 128, 131]. For example, Jeon et al. [69] perform a comparative study of various machine learning algorithms, such as decision trees, support vector machines, discriminant analysis, random forests, and k-nearest-neighbor on data from a wrist-worn wearable device to classify hand tremor severity. Evaluated on 85 patients, the highest accuracy obtained was 85.6% by a decision tree classifier.

Seifert et al. [128] explore the use of radar micro-Doppler signatures for gait analysis across diverse applications, spanning from home security to medical diagnosis, rehabilitation, and assisted living. The objective of their study is twofold: to identify changes

Method	Year	Backbone	Input	Dataset
Pirsiavash et al. [119]	2014		2D Pose	MIT-Olympic [119]
Parmar et al [115]	2017	C3D	RGB	MIT-Olympic [119] UNLV [115]
Li et al. [83]	2018	C3D	RGB	MIT-Olympic [119] UNLV [115]
Parmar et al. [113]	2019	C3D	RGB	AQA-7 [113]
Parmar et al. [114]	2019	C3D	RGB	MTL-AQA [114] UNLV [115] MIT-Olympic [119]
Xu et al. [156]	2019	C3D	RGB	MIT-Olympic [119] Fis-V [156]
Pan et al. [109]	2019	I3D	RGB+ 2D Pose	AQA-7 [113]
Roditakis et al. [124]	2021	I3D	RGB	MTL-AQA [114]
Pan et al. [110]	2021	I3D	RGB+ 2D Pose	UNLV [115]
Tang et al. [138]	2020	I3D	RGB	AQA-7 [113] JIGSAWS [42] MTL-AQA [114]
Yu et al. [166]	2021	I3D	RGB	AQA-7 [113] JIGSAWS [42] MTL-AQA [114]
Farabi et al. [37]	2022	ResNet	RGB	MTL-AQA [114]
Xu et al. [159]	2022	I3D	RGB	FineDiving [159]
Zhang et al. [172]	2022	I3D	RGB	MTL-AQA [114]
Bai et al. [8]	2022	I3D	RGB	AQA-7 [113] JIGSAWS [42] MTL-AQA [114]

Table 2.1: *Overview of AQA methods mainly designed for sports performance evaluation.*

in gait patterns and tackle the intra-motion category classification challenge within gait recognition. To this end, they introduce new gait classification methods based on physical features, subspace features, and sum-of-harmonics modeling. Evaluations were carried out using K-band radar data from four test subjects, considering five unique gait classes for each participant, including standard walking patterns, pathological strides, and assisted ambulations.

While sensor-based methods can accurately capture human kinematics, their reliance on wearable devices, which can be expensive, cumbersome, and sometimes intrusive, limits their convenience and broad applicability in healthcare settings. In contrast, learning-based approaches offer a scalable, contactless, and non-intrusive solution, making them more convenient than sensor-based methods, as they only rely on cameras for data collection.

As an early work, Paiement et al. [108] present a learning-based method for the assessment of human movement quality in healthcare. Their approach is designed for online analysis and focuses on patients who walk on stairs. Utilising 3D skeletal data captured by a Kinect camera from a frontal view, the method consists of two key statistical models, pose and dynamic. The pose model quantifies the likelihood of standard body positions using a probability density function, while the dynamic model accounts for temporal sequences through a continuous-state Hidden Markov Model (HMM). During the inference phase, each individual frame in each sequence is categorised as normal or abnormal. This is based on how much the observed data deviates from the statistical models, assessed via a log-likelihood metric with an empirically determined threshold. In addition, the methodology involves initial steps to preprocess the skeleton data. These include normalisation procedures as well as dimensionality reduction techniques to handle the high-dimensionality problem of skeleton data.

Elkholy et al. [32] develop a system for assessing neuromusculoskeletal disorders in the elderly, specifically focusing on diseases like Parkinson's, relying on 3D skeletal data captured using depth cameras. They focus on three key features: asymmetry, velocity magnitude, and center-of-mass trajectory deformation. These features help to understand both the speed and the pattern of movements. In the training phase, two types of probabilistic models, Gaussian Mixture Model (GMM) and Kernel Density Estimation (KDE), are built based on the descriptors from normal sequences. During inference, the likelihood of a test sequence being normal or abnormal is computed using the trained GMM, and compared with a learned threshold. They also use a multiple linear regression model to give a score to abnormal movements, based on expert medical advice.

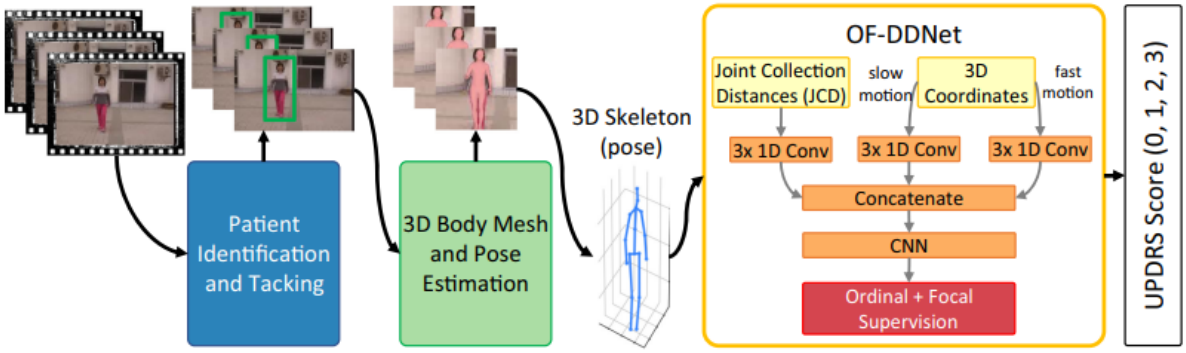


Figure 2.7: *Proposed framework by [96]. The subject is first tracked throughout the video, while other individuals, such as clinicians, are removed. Following that, the 3D body mesh of the identified participant is extracted, along with their skeleton. Finally, the proposed OF-DDNet model estimates the MDS-UPDRS gait score solely based on the 3D pose sequence. The figure is adapted from [96].*

Among deep learning-based methods, Liao et al. [85] propose a comprehensive framework tailored for assessment of home-based rehabilitation. The architecture employs autoencoder neural networks for dimensionality reduction of skeletal joint coordinates, followed by performance quantification and scoring mapping to produce movement quality scores. These scores serve as ground truth for training a deep neural network. This network manages the complexities of human movements by organising data into temporal pyramids and processing joint displacements of individual body parts via a series sub-networks. The architecture combines convolutional layers for spatial features with recurrent layers to capture temporal dependencies.

Lu et al. [96] introduce the first benchmark for classifying PD patients based on MDS-UPDRS [45] gait severity scores, utilising data from 30 research participants each assessed by a board-certified movement disorders neurologist. Their method operates in a series of steps: initially identifying and tracking the subject in the video, followed by the extraction of their 3D body skeleton from each frame. Then, a Temporal Convolutional Neural Network (TCNN) is trained on sequences of these 3D poses. The TCNN is based on a Double-Features Double-Motion Network [161] with a new hybrid ordinal-focal objective. The hybrid ordinal-focal objective in this work combines focal loss [86] and an ordinal loss [123] components, effectively handling data imbalances and leveraging the ordinality of MDS-UPDRS scores. An overview of this framework is shown in Figure 2.7. However, the main issue in [96] is its reliance on one expert's ratings, which could introduce bias into the model. To address this, Lu et al. [97] incorporate scores from three different neurologists, aiming for a more balanced and reliable model. Nonetheless, this addition of multiple raters introduces a source of noise and uncertainty. To manage

this, they propose a system known as rater confusion estimation. This system jointly learns the rater scoring noise and MDS-UPDRS score estimation with the ordinal focal neural network. Specifically, they create a learnable confusion matrix for each rater and optimise it while classifying the input videos using a modified version of ordinal focal strategy proposed in [96].

Turning to a different aspect of motor function, Guo et al. [48] explore the use of video-based evaluations to assess the severity of PD through hand movements. To better understand how the hand's joints work together, they use a Graph Convolutional Network (GCN) to look at the skeleton of the hand. The authors face two primary challenges: extracting fine-grained features and ensuring model stability. To address these issues, they introduce a tree-structure-guided GCN enhanced with group-sparse contrastive learning. This unique method capitalises on the natural tree structure of the human hand to create a sophisticated graph that captures key motion features from the fingertips to the palm. Additionally, the use of contrastive learning [55] allows the model to focus on the discriminative spatial-temporal motion features, rather than the minor differences between sequences that are due to confounding factors.

Liu et al. [92] focuse on PD tremor severity assessment. They address the challenge of capturing subtle and continuous tremors in different body parts (e.g. hand, leg, and jaw). The authors use Eulerian video magnification for preprocessing to amplify subtle tremors and introduce a model, global temporal-difference shift network, to focuse on the micro temporal changes caused by the tremors. To further improve the prediction accuracy, the model incorporates a global shift module, allowing each video segment to consider global temporal features.

In summary, most of the current methods for evaluating human movement in healthcare, particularly those related to PD, are dependent on 3D skeleton data. These methods often involve additional computational costs for preprocessing and do not capture some subtle important features for accurate PD severity assessment. This thesis instead focuses on assessment from only RGB data, which eliminates the need for preprocessing and can capture a richer set of features vital for accurate evaluations.

2.2 Self-Supervised Learning (SSL)

In recent years, deep neural networks, particularly CNNs, have shown remarkable success in various visual recognition tasks [14, 163]. These successes of CNNs have been largely dependent on large training sets of manually annotated data. However, collecting these large annotated datasets is both costly and time-consuming. Consider the action 'shaking

hands', which might only occur for a few seconds in an hour-long video. To train a model for this action using supervised learning, one has to sift through the entire video to manually identify and annotate those specific frames or crop them into a manageable range. The task becomes increasingly challenging and time-consuming as the volume of video data and variety of action classes grow. On the other hand, unsupervised learning methods that exploit unlabeled data provide much more scalable and flexible learning algorithms. Among unsupervised learning approaches, a prominent paradigm is self-supervised learning which provides a way for representation learning that does not need human-labeled data and has shown promise in both image and video domains. Particularly, the process of SSL starts by training the model on an unlabeled dataset using a learning objective designed to capture the underlying structure of the data. The pretrained model is then used as initialisation for the target dataset, where it is fine-tuned using the provided labeled samples.

In this subsection, we provide a comprehensive review of existing methods for SSL. We start by discussing pretext task-based SSL methods. Then, we move on to contrastive learning, another form of SSL. Finally, we review specific AQA methods that have successfully incorporated SSL into their frameworks.

2.2.1 Pretext Tasks

A pretext task is a SSL training goal that uses predefined tasks for the model to solve. This enables the model to learn useful representations for subsequent downstream tasks. The underlying concept is that if a model can solve a complex task that requires high-level understanding of its input, then it will acquire features that are more general [127]. The nature of these pretext tasks varies significantly between image and video domains. In the image domain, pretext tasks often deal with appearance statistics through techniques such as colorization [26, 67], ordering shuffled image patches [103, 106, 152], context prediction [28],and rotation classification [43]. On the other hand, in the video domain, pretext tasks extend beyond appearance statistics to also include temporal features. These may include techniques like temporal order prediction [39, 100, 157], jigsaw [4, 66] or video playback speed prediction [9, 148]. Next, we have a closer look at these specific techniques.

Temporal Order Prediction – The majority of early studies in self-supervised video representation learning focused on temporal order prediction [39, 78, 100, 157]. For example, Fernando et al. [39] propose a pretext task called odd-one-out in which the network takes multiple video sequences as input into its multi-branched architecture with shared weights. The goal is to identify the video sequence that has been sampled

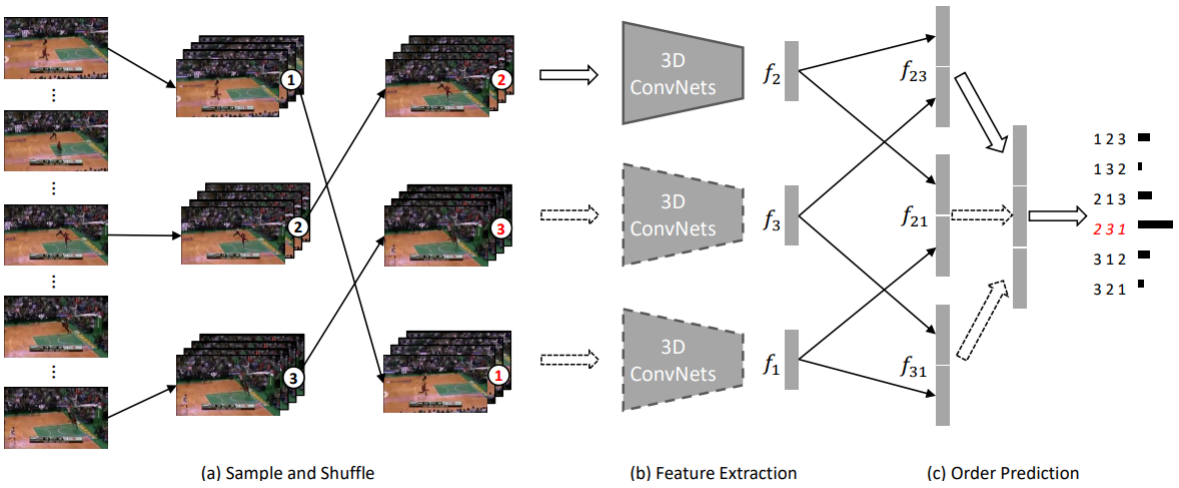


Figure 2.8: *Overview of clip order prediction framework proposed by [157]. (a) Sample non-overlapping video clips and shuffle them to a random order. (b) Employ the 3D ConvNets to extract the feature from all clips (c) The extracted features are pairwise concatenated, and fully connected layers are applied on top to predict the real order. The dashed lines indicate that the corresponding weights are shared among clips. The figure is adapted from [157].*

in an incorrect order. To identify this odd clip, the learning machine must compare all the video clips, identify the regularities among them, and select the one that exhibits irregularities. However, when relying on frame ordering, the difference between two frames might be insufficient to recognise a change in motion for some activities. To address this issue, Xu et al. [157] sample non-overlapping clips and shuffle them to a random order. Then a 3DCNN [90] is utilised to extract features from clips, and these features are processed to predict the actual order. This clip-based order prediction allows for better comparison because the dynamics of an action are maintained in a sub-clip. Figure 2.8 presents the overall framework, which is composed of mainly three procedures.

Jigsaw – Noroozi et al. [106] were the first to create a jigsaw task for images. In this task, an image is divided into 9 shuffled patches and the aim is to train the model to put these shuffled patches back in their original order. To do this, they introduce a unique neural network called a Context-Free Network, a type of siamese CNN that uses shared weights. During training, an image with a random permutation of the nine patches is fed to the network. Given that each image has nine patches, the total number of possible permutations is 9! = 362,880, making it highly unlikely that all permutations could be recognised. To limit this complexity, a Hamming distance measure was employed to ensure the task is appropriately challenging for a CNN, neither too difficult nor too easy.

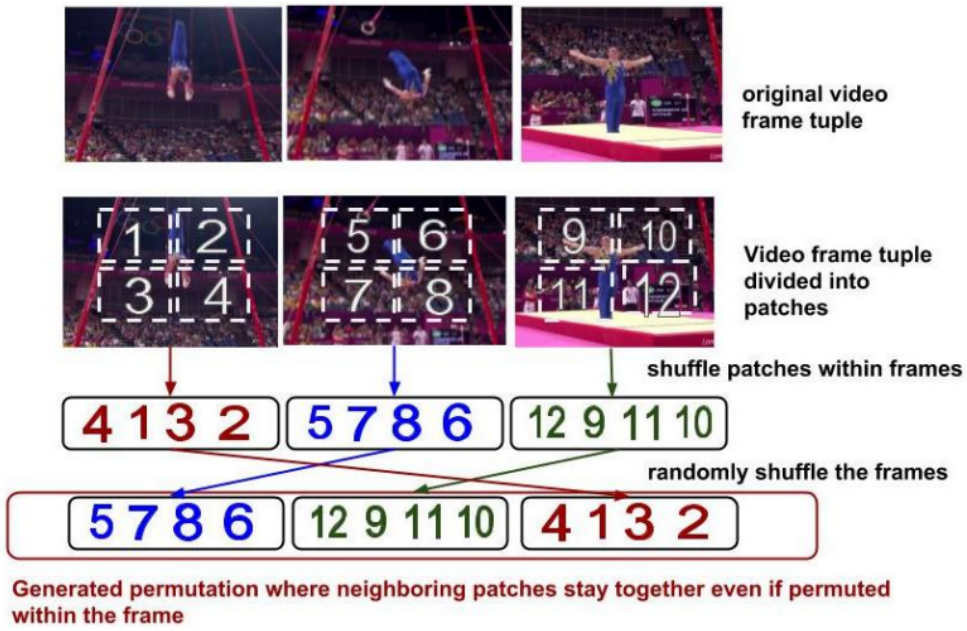


Figure 2.9: *The suggested permutation sampling strategy by [4], which involves randomly shuffling the patches within each frame of a tuple, followed by a permutation of the frames themselves. Given that each frame contains 4 patches, there are 4! = 24 distinct methods for rearranging these patches within a single frame. This process is repeated for all the frames in the tuple, and they finally choose the top N permutations based on Hamming distance. The figure is adapted from [4].*

A primary obstacle in adapting the jigsaw technique to videos is the rise in patch count, which subsequently leads to a larger set of permutations. Ahsan et al. [4] address this problem by dividing a video into clips of three frames, then split a video frame into 2×2 grid of patches which results in $3 \times (2 \times 2) = 12$ total patches per video (see Figure 2.9). Then, a multi-stream Siamese-like network (similar to [106]) is trained to predict both the spatial location of a patch within a frame and its temporal position over time. Note that before shuffling the frames themselves, the patches within each individual frame were initially shuffled.

Huo et al. [66] claim that directly solving 3D jigsaw puzzles is intractable due to the enormous number of possible permutations. As a solution, the authors develop Constrained Spatiotemporal Jigsaw, where the 3D puzzles are created in a constrained way to include large, continuous spatiotemporal cuboids. These cuboids serve as cues for the model to learn about spatiotemporal continuity. To make this task more manageable, they introduce four surrogate tasks that are more solvable, designed to train the model to be sensitive to spatiotemporal continuity on both local and global scales.

Video Playback Speed Prediction – Recently, estimating video playback speed has attracted much interest as a highly effective way to encourage the model to learn features

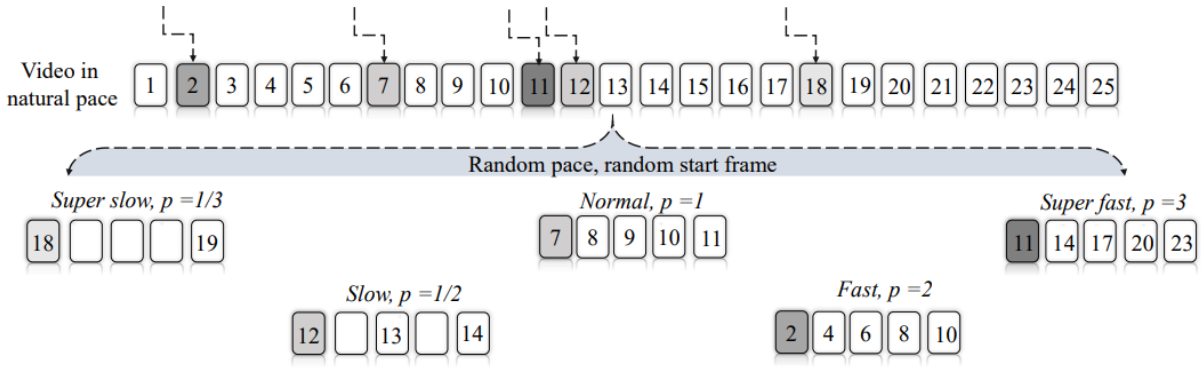


Figure 2.10: *Generating samples and speed labels from the pretext task proposed in [148]. Here, five different sampling paces are shown, ranging from super slow to slow, normal, fast, and super fast. The darker the initial frame appears, the faster the clip plays through. The figure is adapted from [148].*

(of moving objects) in videos [9, 16, 34, 65, 68, 148, 164].

Epstein et al. [34] design a method to predict normal video speed to detect an unintentional event in the video. SpeedNet [9] determines whether a given video clip is being played at normal or twice its original speed. Recently, Wang et al. [148] proposed VideoPace to predict the specific speed of each video clip which is randomly sampled at a different frame rate. The sampling strategy of this pretext task is shown in Figure 2.10. Formally, the authors represent pace sampling transformation as $g(x)$. For a given video x, they utilise $g(x|p)$ to generate the training clip x_e with an associated training pace p. The task of predicting the pace serves as a classification problem. Consequently, their neural network $f(x_e)$ is optimised using a cross-entropy loss L_{cls}:

$$L_{\text{cls}} = -\sum_{i=1}^{M} y_i \left(\log \frac{\exp(h_i)}{\sum_{j=1}^{M} \exp(h_j)} \right), \quad h = f(x_e) = f(g_{\text{pac}}(x|p)), \tag{2.1}$$

where M is the number of all the pace rate candidates.

Wang et al. [148] also prevent the network from taking shortcuts or cheating to accomplish their pretext task (VideoPace) by applying color jittering augmentation to each frame. However, one of the main limitations in considering playback speed alone is that video clips with different speed labels might appear similar to each other, e.g. when different athletes might perform the same sporting action at different speeds. In Chapter 4, this thesis proposes a novel pretext task to address the issue of inaccurate video speed labeling.

2.2.2 Contrastive Learning

Contrastive Learning (CL) is a discriminative method that differentiates between similar (often referred to as 'anchors' and 'positives') and dissimilar ('negatives') samples by attracting the similar ones closer and repelling the dissimilar ones in the feature space [76, 155]. A similarity metric measures the proximity between two feature vectors. In typical CL [155], a 'positive' sample is created from a single data point through data augmentation, while other images in the dataset are designated as 'negative' samples. However, this method demands substantial memory for sample storage and large batch sizes for effective training. To handle this issue, MoCo [55] employs an on-the-fly dictionary with the goal of aligning a query with its positive key encoding while maximising dissimilarity to the negative key encodings. Given a query embedding q and a set of keys embeddings $\{k_i\}_{i=0}^{K}$ in the dictionary queue, the contrastive loss for MoCo can be expressed as:

$$L_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}, \tag{2.2}$$

where τ is a temperature coefficient in learning. The sum is over one positive and K negative samples. This loss function tries to classify q as k^+ (positive key) through a softmax classification process.

As depicted in Figure 2.11, the MoCo framework essentially consists of two networks: an encoder network responsible for extracting the query q, and a momentum encoder network for generating the key feature vectors $\{k_i\}_{i=0}^{K}$. These keys are stored in a dynamic queue that operates on a First-In, First-Out basis. When a new mini-batch of image embeddings is processed, it pushes new keys to the head of the queue, simultaneously removing the same number of older keys from the tail.

After the contrastive loss is computed, the encoder network is updated through backpropagation. In contrast, the momentum encoder network is updated via a momentum-based rule which can be expressed as:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, \tag{2.3}$$

where θ_q and θ_k are parameters of the encoder and momentum encoder, respectivly. Here $m \in [0, 1)$ is the momentum coefficient. This slowly updating of the momentum encoder guarantees stable key representations.

To eliminate the need for negative samples in CL, Grill et al. [47] introduce BYOL. Unlike traditional approaches that emphasise dissimilarities, BYOL focuses on the similarities

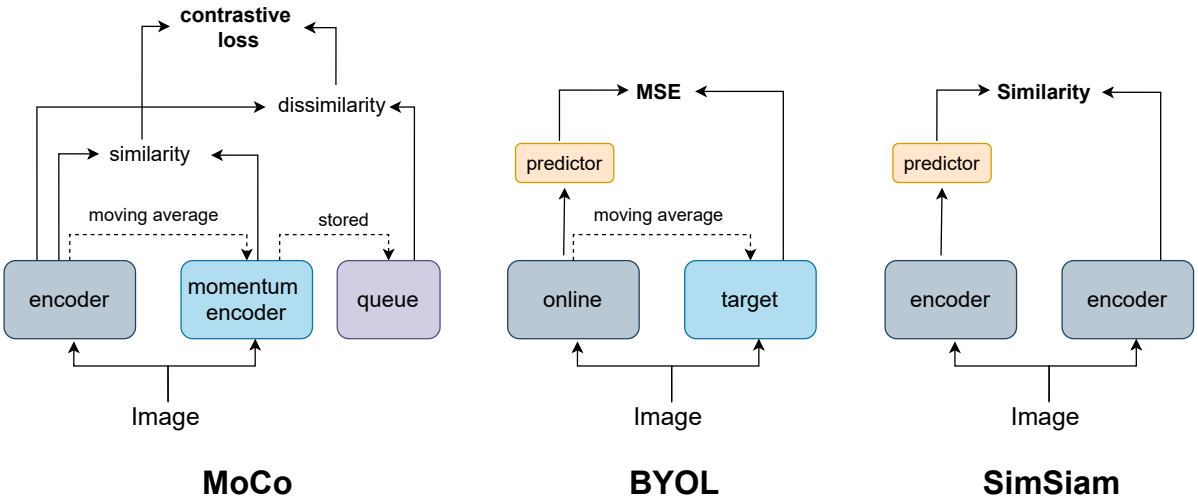


Figure 2.11: *Architecture details of three different contrastive learning approaches including MoCo, BYOL and SimSiam.*

between samples and their corresponding representations. As shown in Figure 2.11, it employs two encoders: an online network and a target network. These two networks share the same architecture, with the online network having an additional predictor head. During the training process, two augmented versions of the same image are inputted into the online and target networks to generate embedding vectors. The online network updates its encoder using these vectors, and these updated weights are then transferred to the target network as an exponential moving average [55]. Unlike traditional methods that use contrastive loss, BYOL utilises mean squared error to minimise the distance in similarities between embedding vectors made by the online target networks. The loss function in BYOL can be represented as follows:

$$L_{\theta,\xi} \triangleq \|q_\theta(z_\theta) - z'_\xi\|, \tag{2.4}$$

where θ and ξ are the parameters for the target and online networks respectively, $q_\theta(z_\theta)$ is the projection of the latent representation z_θ from the online network, and z'_ξ is the target representation.

Chen et al. [19] propose SimSiam to show that it is feasible to acquire a robust representation without the need for negative samples or a momentum encoder. The SimSiam model takes two augmented views and aims to maximise their similarity. These views undergo an encoding process via an encoder network that shares the weights between the two views. This encoder is constructed from a ResNet [54] backbone coupled with a projection MLP. For one of the view representations, an MLP predictor is employed, and then the stop-gradient operation is applied to another view representation to avoid

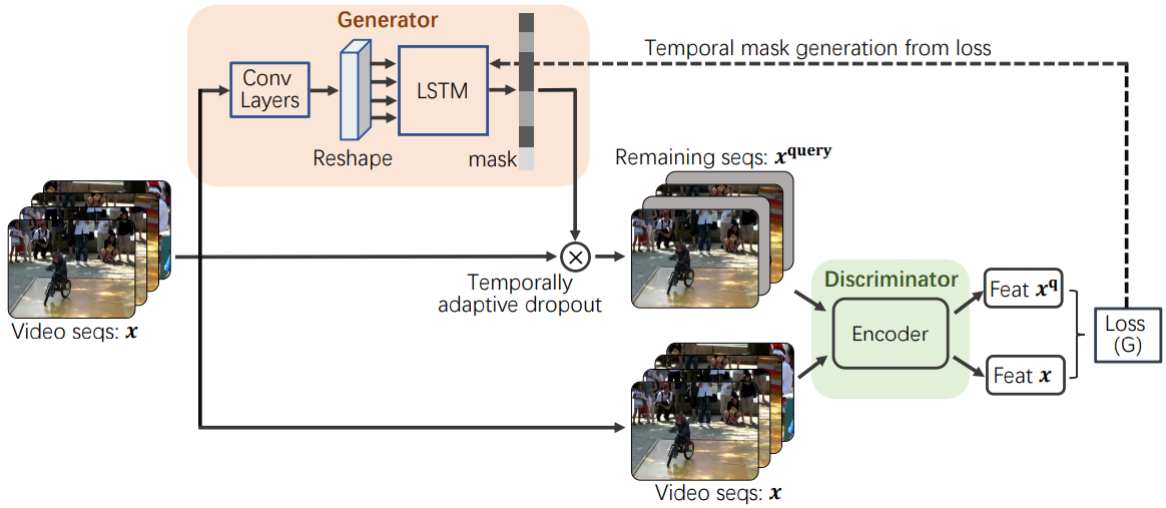


Figure 2.12: *An overview of temporally adversarial learning proposed by [111]. The figure is adapted from [111].*

collapse. The architecture of SimSiam is illustrated in Figure 2.11.

Due to the promising results of contrastive learning in the image domain, some works have extended contrastive methods to the video domain [16, 65, 111, 162], with the aim of capturing temporal dependencies in addition to spatial features.

VideoMoCo [111] builds upon MoCo to refine its application in video representation learning. Similar to MoCo, it follows the usage of a queue and a moving-average encoder. To make the encoder better at capturing temporal features, VideoMoCo utilises an adversarial learning approach comprising a generator (G) and a discriminator (D), as depicted in Figure 2.12. When provided with an input sample x in the form of a video clip, G selectively drops several frames to produce a new sample called x^{query}. The discriminator (D) then analyses features from both the original x and the modified x^{query}, calculating a similarity loss between them. This loss term is then used reversely to train G. During training iterations, G is learned to continuously attack D by removing different frames of x. Concurrently, D learns to defend this attack by encoding features that are resilient to temporal variations. The x^{query} is then used for contrastive learning. Furthermore, to deal with the discrepancy arising from the evolution of the momentum encoder, they propose a temporal decay to model key degradations in the memory queue. This helps the query sample to focus on the newest keys when computing the contrastive loss.

Chen et al. [16] introduce RSPNet to predict the relative speed between two video clips rather than the direct playback speed. As illustrated in Figure 2.13, the authors introduce two primary tasks: the Relative Speed Perception (RSP) task, which encourages

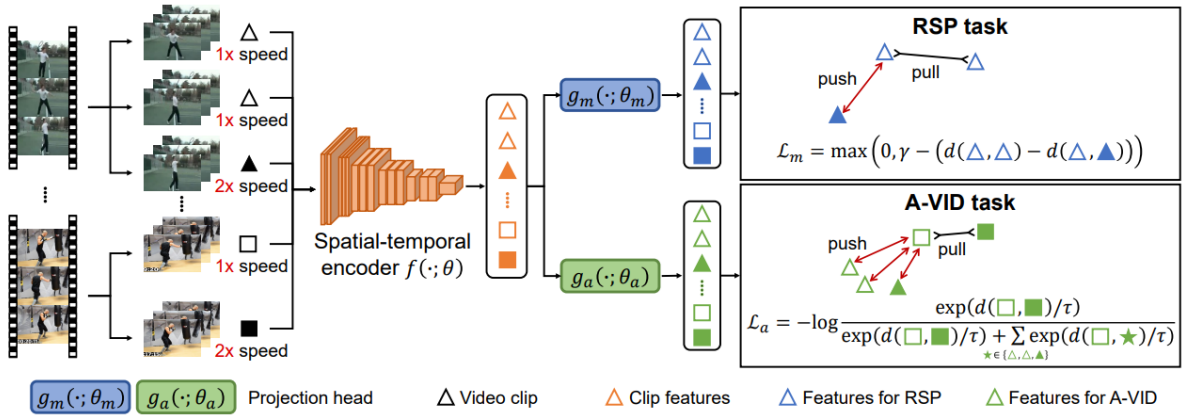


Figure 2.13: *Illustration of RSPNet proposed by [16]. Utilising a set of video clips at different speeds, an encoder $f(;\theta)$ followed by two projection heads (namely, gm and ga) is used to extract features for two tasks. In the first task, Relative Speed Perception (RSP), the aim is to identify the relative speed differences between clips. For the second task, Appearance-Focused Video Instance Discrimination (A-VID), the focus is on distinguishing video clips based on their visual content. Both tasks are formulate as a metric learning problem, and triplet loss $\mathcal{L}_m$ and InfoNCE loss $\mathcal{L}_a$ are used for the training process. The figure is adapted from [16].*

the model to capture motion features, and the Appearance Video Instance Discrimination (A-VID) task, designed to model appearance features. For the RSP task, they use a triplet loss to minimise the distance between two clips of the same video at the same playback speed and maximise the distance between two clips of the same video at different playback speeds. For the A-VID task, they extended the contrastive learning in image domain [55] to video. In particular, they sample two clips from the same randomly selected video v, and K clips from videos in subset $S\backslash v$. These clips are processed through a spatial-temporal encoder, followed by a projection mechanism. Then they compare the features generated for the two clips from the same video (considering them a positive pair) and contrast them against the features of the other clips (considering them negative pairs) using InfoNCE loss [107] method.

Different from RSPNet, which focuses on predicting relative speeds, Huang et al. [65] propose ASCNet that emphasises speed similarity. They introduce two innovative tasks: Appearance Consistency Perception (ACP) and Speed Consistency Perception (SCP). In the ACP task, two clips are sampled from the same video but with different playback speeds. The goal is to encourage the feature representations of these two clips to be closely aligned in the feature space. On the other hand, the SCP task focuses on minimising the feature distance between two clips sampled from two distinct videos that share the same playback speed. An overview of ASCNet is shown in Figure 2.14.

While aforementioned SSL techniques have achieved remarkable success in video repre-

sentation learning, their effectiveness falters when the pretraining dataset is limited or when there is a pronounced domain gap between the source (unlabeled data for pretraining) and target tasks (labeled data for fine-tuning). To address these challenges, in Chapter 4 an auxiliary learning stage to self-supervised pretraining is introduced. This auxiliary phase leverages similarity-based knowledge distillation and promises improved generalisation using considerably less video data for pretraining, e.g. Kinetics-100 instead of Kinetics-400.

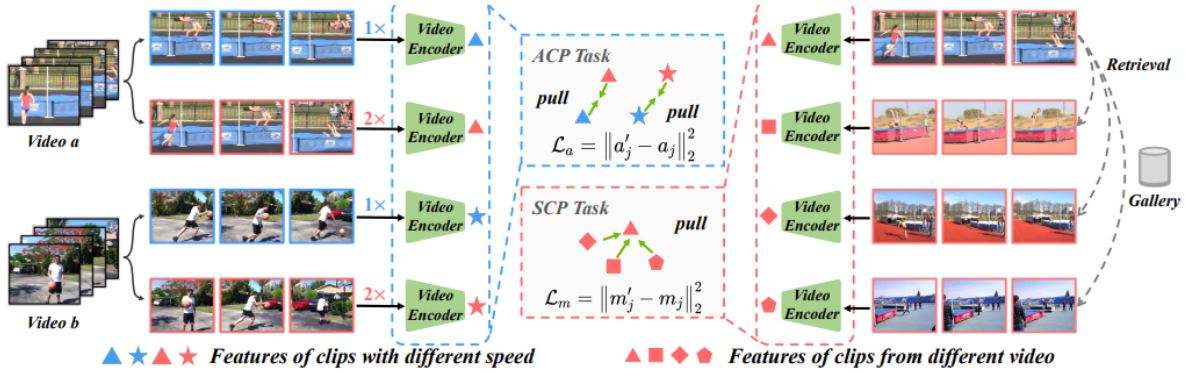


Figure 2.14: *Illustration of ASCNet framework [65]: A set of video clips played at different speeds (such as 1× and 2×) is processed through a video encoder f, which maps them into appearance and speed embedding space. In the context of the ACP task, the authors draw the appearance features from identical videos closer to each other. For the SCP task, they initially identify videos of the same speed that have similar content and then bring their speed features into closer alignment. The figure is adapted from [65].*

2.2.3 SSL for AQA

Although most of the state-of-the-art works in the AQA literature have focused on supervised learning approaches, a few have recently explored self-supervised learning [89, 124, 171]. In most of these studies, in addition to the traditional supervised regression loss, the framework is further equipped with an SSL loss during the finetuning stage to improve the performance without the need for additional annotations. For instance, Roditakis et al. [124] leverage Temporal Cycle Consistency (TCC) [31] embeddings to improve the accuracy of quality score estimation. Their method consists of two learning stages and a temporal alignment phase. In the first stage, TCC is employed to extract self-supervised embeddings which are subsequently used to align the video clips temporally. Then, the aligned video clips are fed into the second learning phase to evaluate the quality of actions. The second stage is supervised and incorporates features extracted by TCC and I3D backbones to learn the quality score of action. During this second training phase, only the I3D network undergoes optimisation, while the TCC model remains

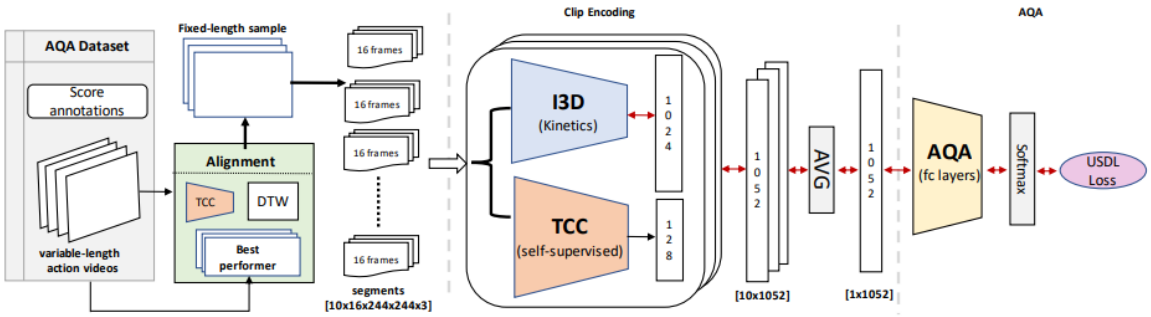


Figure 2.15: *The overall architecture of the self-supervised alignment for action assessment [124]. Every video sample is first aligned to a fixed-sized reference video that corresponds to the video of the best-performing individual within the training set. The aligned sequence is then divided into M separate segments. These segments are processed through two backbone networks: I3D and TCC. The features of both backbones are concatenated with average temporal pooling to create a clip-level representation, which is then used for quality score prediction. The figure is taken from [124].*

unchanged. This optimisation utilises the uncertainty loss function as described in [138]. Figure 2.15 provides an in-depth view of the suggested approach.

Zhang et al. [171] propose an adversarial self-supervised framework for semi-supervised AQA, aiming to learn with a limited set of labeled data. Their method leverages temporal patterns from unlabeled videos and aligns the representation distribution of labeled and unlabeled samples using adversarial learning. By introducing a masked segment feature recovery learning on unlabeled videos, the framework captures the temporal dependencies essential for AQA. Additionally, it aligns the feature representations between labeled and unseen unlabeled through a tripartite module system that includes masked segment feature recovery, action assessment, and representation distribution alignment, jointly trained to enhance semi-supervised learning efficiency.

In the domain of surgical skill assessment, Liu et al. [89] augment their supervised model with a self-supervised contrastive loss to enhance the capture of temporal dynamics in surgical videos. This self-supervised branch increases the model's ability to overcome the limitations posed by limited annotated data in surgical skill assessment tasks.

In Chapter 5, the advantages of SSL are leveraged during the *pretraining* stage of AQA frameworks. This approach introduces a level of domain-specific focus to the pretrained model, enabling more efficient handling of downstream AQA tasks.

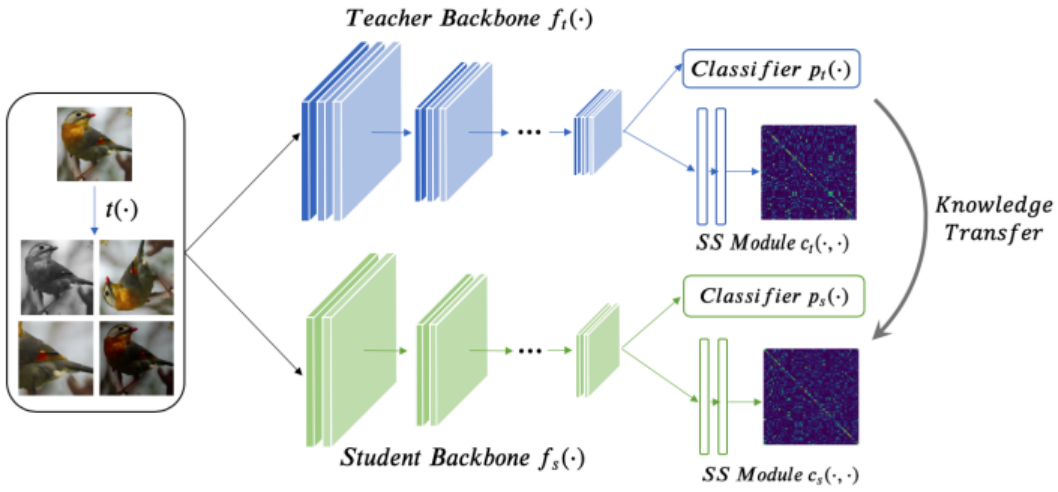


Figure 2.16: *The training architecture of SSKD [158]. Input images undergo specific transformations to get them ready for the self-supervision task. Both the teacher and student networks are composed of three components: the backbone $f(\cdot)$, the classifier $p(\cdot)$, and the SS module $c(\cdot,\cdot)$. The teacher's training is divided into two phases. In the first phase, $f_t(\cdot)$ and $p_t(\cdot)$ are trained through a classification task. The second phase focuses on fine-tuning $c_t(\cdot,\cdot)$ using a self-supervision task. During the student's training, the student is encouraged to mimic the teacher in terms of both the classification and self-supervision outputs, in addition to the standard label loss. The figure is adapted from [158].*

2.3 Auxiliary Learning

To assist a primary task to generalise better to unseen data, training through auxiliary learning is an effective approach [91, 104, 129]. By training on multiple tasks, the model gains the ability to learn extra features that it would not have acquired by focusing only on the primary task. This broadened feature set aids the model in performing better on unfamiliar data for the primary task. This is different from multi-task learning, where all tasks are considered equally important with the objective of optimising performance across them. In auxiliary learning, the focus is on optimising a single primary task, using the other tasks just as supplementary aids to boost its performance.

Auxiliary learning has been applied alongside a wide range of techniques, such as transfer learning [145], reinforcement learning [87], semi-supervised learning [169], and knowledge distillation [158]. Liu et al. [91] introduce a method called Meta Auxiliary Learning (MAXL) to enhance the generalisation performance of a primary supervised learning task without the need for manually labeling auxiliary data. The framework consists of two neural networks: a) a multi-task network for training both the primary and auxiliary tasks, as in standard auxiliary learning, and (b) a label-generation network for

automatically creating labels for the auxiliary task. The central concept of MAXL is to leverage the performance of the primary task, when trained with the auxiliary task in a given iteration, as a basis for enhancing the auxiliary labels in subsequent iterations. The authors empirically demonstrate that MAXL outperforms single-task learning and other baseline methods for generating auxiliary labels across multiple image datasets, all without requiring additional data.

In recent work, Xu et al. [158] introduce the Self-Supervised Knowledge Distillation (SSKD) framework, which harnesses SSL as an auxiliary task. SSKD integrates a lightweight auxiliary module within the teacher network to distill self-supervised signals, empowering the student model with a more structured understanding. The framework's model-agnostic design further adds the advantage of versatility, enabling the transfer of knowledge across various architectural designs. Notably, SSKD's effectiveness has been validated in scenarios like few-shot learning and environments with noisy labels. A detailed illustration of the SSKD methodology is provided in Figure 2.16.

In Chapter 4, it is demonstrated that the similarity information between the embedded feature points can serve as prior knowledge for self-supervised pretraining, enabling the learning of more generalised representations through pretext tasks, which serve as the primary task in this context. To capture this similarity information, a variation of knowledge distillation, called similarity-based knowledge distillation [140, 144], is employed as an auxiliary task.

2.4 Knowledge Distillation

Knowledge Distillation (KD) is a technique that aims to transfer the knowledge learned by a cumbersome, high-capacity model (referred to as the teacher) into a cheaper and faster model (known as the student) without losing too much generalisation power. This process allows the student model to inherit the generalisation capabilities of the teacher model but with the benefits of reduced computational complexity and latency. The concept was popularised by the seminal work of Hinton et al. [60], where they demonstrated that the dark knowledge contained in the soft probabilities of a teacher model could be transferred to a student model through the minimisation of the Kullback-Leibler (KL) divergence between their respective output distributions. In this way, the student model can learn how teacher network studied given tasks in a compressed form.

Formally, for any given input (image/video) x, the teacher network generates a vector of scores $s^t(x) = [s^t_1(x), s^t_2(x), \ldots, s^t_K(x)]$. These scores are subsequently turned into

probabilities using $p_k^t(x) = \frac{e^{s_k^t(x)}}{\sum_j e^{s_j^t(x)}}$. As trained neural networks often produce sharp probability distributions, Hinton et al. [60] suggest softening these distributions with temperature scaling:

$$\tilde{p}_k^t(x) = \frac{e^{s_k^t(x)/\tau}}{\sum_j e^{s_j^t(x)/\tau}}, \tag{2.5}$$

where $\tau > 1$ is a temperature hyperparameter. Similarly, the student network generates a softened probability distribution, $\tilde{p}_k^s(x)$. The overall loss L for the student is a linear combination of the standard cross-entropy loss L_{cls} and a knowledge distillation loss L_{KD}:

$$L = \lambda L_{\text{cls}} + (1 - \lambda) L_{\text{KD}}, \tag{2.6}$$

where $L_{\text{KD}} = -\tau^2 \sum_k \tilde{p}_k^t(x) \log \tilde{p}_k^s(x)$, and λ is a hyperparameter, with a typical choice being $\lambda = 0.9$ [21, 60]. Since the introduction of this concept, numerous advances have been made in exploring various aspects of KD. For example, Zagoruyko et al. [168] propose attention transfer that focuses on the feature maps of the network as a mechanism of transferring knowledge as opposed to the output logits [60]. In [165], the transfer of knowledge was achieved using the Flow of Solution Procedure, which involves calculating the Gram matrix of features across various layers. Heo et al. [59] propose an activation transfer loss metric to distill the activation boundaries formed by hidden neurons from the teacher to the compact student network. Using a double distillation objective, Zhang et al. [170] train a network for new classes and then merge it with the network focused on previous classes. The final model suffers less from forgetting the old classes, while maintaining high accuracy in recognising the new classes.

Similarity Based Knowledge Distillation – Similarity-based Knowledge Distillation (SKD) methods [2, 36, 112, 117, 139, 140, 144] train a student to mimic the similarity score distribution inferred by the teacher over data samples. Most early works in SKD use a supervised loss during distillation [112, 117, 144]. For example, Park et al. [112] propose Relational Knowledge Distillation (RKD) which introduces distance-wise and angle-wise distillation losses to preserve the structural relationships, such as relative distances and angles between data points, as learnt by the teacher model. RKD can be seen as an extension of traditional knowledge distillation, complementing it by emphasising the importance of the relationships between data points in the embedding space. Their experiments across multiple tasks — metric learning, image classification, and few-shot learning — demonstrate that RKD not only improves the performance of student models

but also can potentially enable them to surpass the teacher models.

Tung et al. [144] proposes a novel SKD method, based on the observation that semantically similar inputs yield similar activation patterns in a trained neural network. Utilising this insight, their method computes pairwise similarity matrices from output activation maps for a given mini-batch of images. Then a distillation loss is defined based on these matrices to guide the training of the student network, encouraging it to mimic the activation patterns of the teacher network.

There are a number of works that employ SKD in the context of contrastive learning [2, 35]. Abbasi et al. [2] propose a method for compressing deep SSL models, specifically focusing on transferring the discriminative power of a teacher model to a smaller student model. In this approach, the teacher's model parameters are kept frozen, serving as a fixed reference. The key idea is to transfer the probability distribution of the similarity between anchor and query points from the teacher's embedding space to the student's. The similarity measure employed is cosine similarity, converted into a probability distribution using a SoftMax operator. An overview of this method is shown in Figure 2.17. This work ([2]) also explores variations such as using the teacher's memory bank for the student and caching the teacher's embeddings to save computational cost.

While [2] relies on a pretrained frozen teacher model, Tejankar et al. [139] propose an iterative SKD method, known as ISD, where the teacher model continues to learn similarity score distributions during training. ISD can be considered a more relaxed version of contrastive learning methods such as BYOL [47]. Unlike BYOL, which compares a query image only with a differently augmented version of the same image, ISD compares the query image with other random images. More precisely, the ISD teacher network measures the similarity of the query image to a set of anchor points stored in a memory bank. It then converts this similarity into a probability distribution over neighboring examples. This knowledge is transferred to the student network, enabling it to mimic the same neighborhood similarity. As a result, ISD allows the embedding of the query image to vary, as long as its neighborhood similarity remains consistent. This is in contrast to BYOL, which aims to keep the embedding of the query image unchanged when subjected to augmentations. In this work, it should be noted that the teacher model is updated using a momentum method to be the running average of the student, similar to MoCo [55]. The student is also updated based on KL divergence loss.

In Chapter 4, an auxiliary learning stage for SSL is introduced. Although this stage employs an architecture similar to ISD [139], it is designed to meet different objectives, (i) extending ISD to extract representations from video data rather than images,

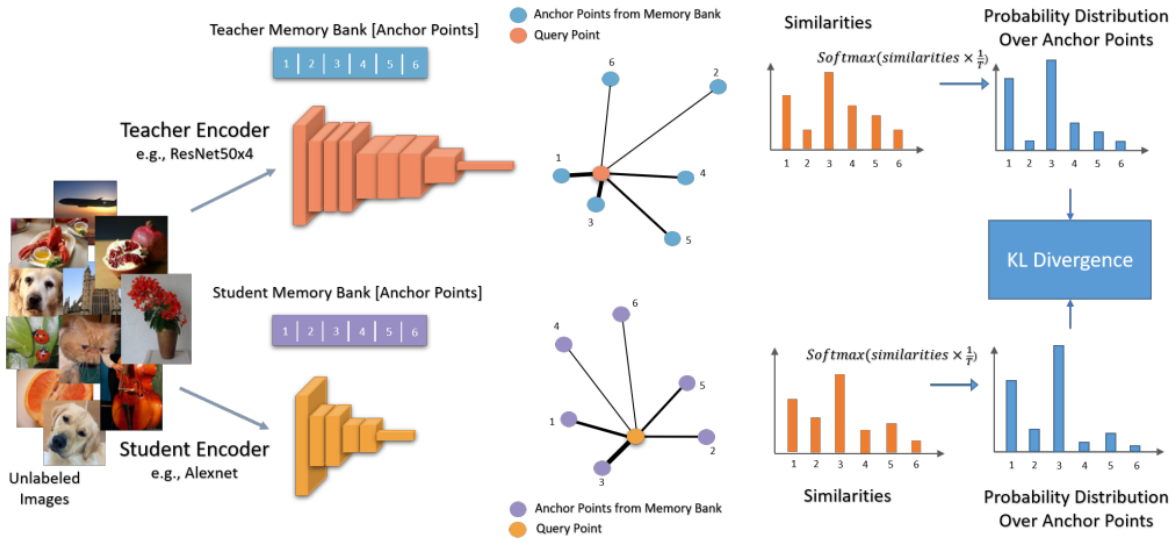


Figure 2.17: *An overview of compression framework proposed in [2]. The aim is to transfer the self-supervised teacher's knowledge to the student model. Each image is compared with a random set of data points, called anchors, to produce a set of similarity measures. These measures are then translated into a probability distribution over the anchors, representing each image through its nearest neighbours. As the aim is to transfer this knowledge to the student, an equivalent distribution from the student is also obtained. The final step involves training the student to reduce the KL divergence between the two distributions. The figure is taken from [2].*

and (ii) employing distilled similarity representations as auxiliary knowledge for various self-supervised pretraining methods, rather than using them directly in downstream tasks.

2.5 Continual Pretraining

In contrast to the traditional approach in transfer learning that follows domain-general pretraining (usually over ImageNet or Kinetics-400), continual pretraining can enhance learning via in-domain self-supervised pretraining to handle domain shift problems [7, 49, 118, 122, 154, 160]. As an early contribution, Gururangan et al. [49] show the importance of an additional pretraining phase with in-domain data to improve their target task performance on text classification.

In the image domain, Reed et al. [122] verify that models continually pretrained on datasets that are progressively more similar to the target data can speed up convergence and increase robustness, while being particularly helpful when the target training data is limited. To implement this strategy, they introduce a pretraining paradigm called HPT. Initially, HPT starts with base pretraining on a general-domain dataset, such as Ima-

geNet. Optionally, a subsequent phase, called source pretraining, utilises domain-specific datasets. The authors select these domain-specific datasets from a pool of general-domain data using a task-aware search strategy [74]. Finally, HPT performs pretraining on the target dataset, known as the target pretraining. It is worth noting that all phases of pretraining use MoCo-V2 [20] as the self-supervised training method.

Azizi et al. [6] show that models pretrained using SSL with natural images often outperform those pretrained in a supervised manner for medical image classification. They find that further SSL pretraining with domain-specific medical images yields the best performance. In a more recent study [7], they also reveal that a combination of both supervised pretraining on large-scale generic dataset (e.g. ImageNet) and intermediate contrastive SSL [18] on domain-specific medical data enhances the efficacy and robustness of the model for various medical imaging tasks.

In Chapter 5, continual pretraining is investigated for the first time in the video domain, specifically focusing on the AQA task. Instead of applying continual pretraining to the entire model parameter set, advantage is taken of parameter-efficient transfer learning. This approach reduces the cost of storage and model pretraining on in-domain data, while preserving the knowledge obtained through the initial pretraining on domain-general data.

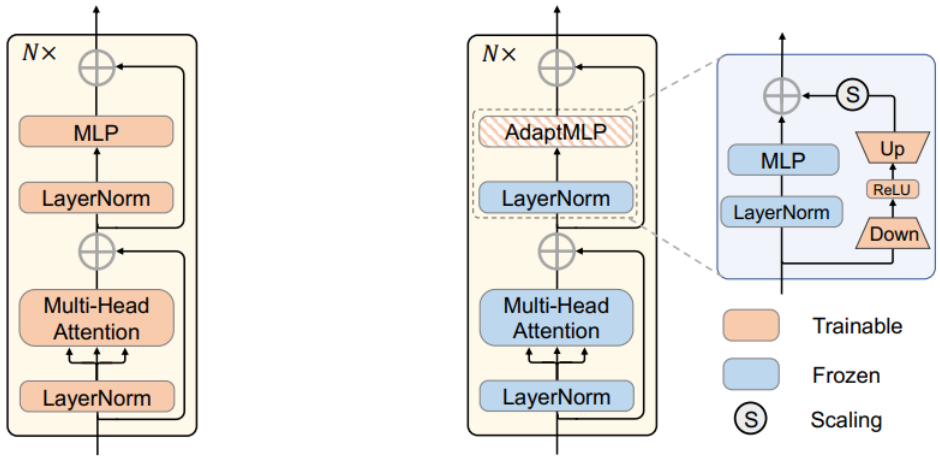
2.6 Parameter-Efficient Transfer Learning

Parameter-Efficient Transfer Learning (PETL) has become an essential approach for adapting pretrained models to new tasks, focusing on the fine-tuning of only a minimal number of parameters. This strategy enables faster adaptation to new tasks and enhances computational efficiency. This approach not only allows for quicker adaptation to new tasks, but also improves computational efficiency. Recently, two main categories of PETL approaches have been proposed, Prompt Tuning (PT) and Adapter Tuning (AT).

In PT [70, 82, 93, 136], a small number of learnable prompt vectors, also known as soft prompts, are prepended to the input embeddings of the model. Only these added tokens need to be fine-tuned for each downstream task. Compared to conventional fine-tuning, this approach offers almost the same level of performance but needs 1000 times less storage space for parameters [82].

Ju et al. [70], for the first time, introduce the concept of Visual Prompt Tuning (VPT) by adapting PT from Natural Language Processing (NLP) to vision transformers. They investigate two versions of VPT: (i) appending a series of learnable parameters to the

input of each transformer layer, termed as VPT-Deep; (ii) solely inserting the prompt parameters into the input of the first layer of Transformer encoder, termed as VPT-Shallow. In their experiments, VPT-Deep surpasses all other parameter-efficient tuning approaches across various tasks, making it the best fine-tuning strategy in storage-limited scenarios. On the other hand, although not as effective as VPT-Deep, VPT-Shallow still delivers significant improvements over head-oriented tuning methods (e.g. linear), making it a worthwhile choice under severe storage constraints.



(a) Full fine-tuning. (b) AdaptFormer fine-tuning

Figure 2.18: *Comparison of conventional fine-tuning and fine-tuning via AdaptFormer [17]. In AdaptFormer approach, the original MLP block of the vision transformer [29] is replaced with AdaptMLP. The AdaptMLP contains two branches: the left one being a frozen branch and the right one being a trainable down → up bottleneck module. Similar with the adapter architecture designed for NLP, this module employs a down projection followed by a ReLU activation and an up operation. The figure is adapted from [17].*

Similar to PT, AT was initially proposed for NLP tasks within transformer architectures [27, 53, 62, 64]. An adapter introduces a bottleneck mechanism by down-projecting the input $h \in \mathbb{R}^d$ to a lower-dimensional m-dimensional space ($m < d$). A non-linear function $g(\cdot)$ is applied, followed by an up-projection back to d-dimensions. A residual connection is added, yielding the final output as:

$$h \leftarrow h + g(hW_{\text{down}})W_{\text{up}}, \tag{2.7}$$

where $W_{\text{down}} \in \mathbb{R}^{d \times m}$ and $W_{\text{up}} \in \mathbb{R}^{m \times d}$ are the matrices used for down- and up-projections, respectively. For PETL, these adapter modules are inserted between the layers of the pretrained model, and only these new modules are fine-tuned while the rest of the model remains frozen.

Beyond NLP, AT has also been adapted for computer vision tasks. Chen et al. [17] presented AdaptFormer, designed to efficiently tailor a pretrained vision transformer model [29] for scalable image and video recognition tasks (see Figure 2.18). Similarly, in [13], a Conv-Adapter architecture was proposed, employing convolution layers to adapt 2D CNNs.

To our knowledge, no work has yet explored the utility of AT in 3D CNNs, presenting an opportunity that this thesis fulfills in Chapter 5.

2.7 Conclusions

This chapter provided a detailed overview of the relevant literature that serve as the basis for this thesis. Since the core focus of the thesis is on AQA, we initially reviewed works in this field, including both sports scoring and human movement assessment techniques in healthcare, particularly for PD. Due to the limited amount of labelled data in AQA tasks, and even more so for PD, we turned to SSL strategies. In this context, we covered a variety of approaches, including pretext tasks and contrastive learning methods to address the data limitation issue. Finally, the chapter looked at other learning methods aimed at enhancing the efficacy of SSL. These ranged from auxiliary learning and knowledge distillation to continual pretraining and parameter efficient transfer learning. By combining these methods, we aim to build a better and more efficient SSL system to assess the severity of PD.

Chapter 3

Supervised Learning with 3D CNNs and Motion Boundaries

3.1 Introduction

In this chapter, we introduce a novel, end-to-end approach for assessing the severity of Parkinson's disease motor states in clinical settings using only video data. Our method is based on UPDRS, a commonly used clinical assessment tool, to make the evaluation process accurate and efficient. The work in this chapter was published in [22].

One of the challenges of using video data for clinical studies is the variability introduced by camera motions. These movements can interfere with the assessment, making it difficult to obtain an accurate measure of a patient's condition. To overcome this, we propose to use motion boundary features [25], calculated using optical flow algorithms, as a way to stabilise the video input. This helps the model focus on the patient's movements rather than getting confused by unrelated camera motions.

Our model is built on a multi-stream deep learning configuration that uses not only RGB video frames but also optical flow and the aforementioned motion boundary features. In this framework, we employ the I3D CNN [10] to efficiently learn spatial and temporal features from these multiple input streams. We also model long-range temporal structure in the patient's actions since assessing only a few moments of an action could result in different scores by a rater, e.g. rapid hand opening and closing sequences may be very similar in part, but in one case the hand may fail to keep up a consistent amplitude and speed of movement towards the end of the sequence due to fatigue (as occurs in PD) or may start badly at the start of the sequence but get better as the action evolves. To

this end, we adopt a sparse temporal sampling strategy (as proposed in [149]) to train our network. This allows for stacks of a few consecutive frames from different segments of the input video to be processed by the 3D CNN independently at inference time and their final scores averaged only at the end (see Figure 3.2).

Inspired by the success of 'attention', now commonly used in deep learning networks, e.g. for human action recognition [95, 116], we engage attention units which assign individual attention weights over each temporal feature vector. Unlike spatial attention [151, 153] which emphasises specific regions within a frame, our model benefits from temporal attention by selectively emphasising the segments of a video that contain pivotal classification information. This approach reflects clinical practices where assessments are often based on momentary actions, such as an interruption or hesitation during the hand movement task. Thus, the network is designed to prioritise critical temporal sequences over spatial features, which is particularly effective for capturing the dynamic and progressive nature of Parkinson's disease symptoms.

In this chapter, we use a dataset (named PD2T) collected from 25 clinically diagnosed PD patients who underwent UPDRS assessments of their motor function after withholding symptom improving dopaminergic medication overnight, focusing on the rapid hand opening and closing and gait components. We train and test our model via a subject-level N-fold cross validation scheme to evaluate its performance and compare against other popular deep learning architectures – in particular to demonstrate the importance of the use of motion boundaries.

In Section 3.2, we present PD2T dataset. Section 3.3 elaborates on our proposed approach, detailing the end-to-end deep learning framework we have developed for this purpose. Section 3.4 focuses on the experiments we conducted to validate our approach, including both the setup and the results. Finally, Section 3.5 offers a conclusion and summarise our key findings in this chapter.

3.2 Dataset

3.2.1 PD2T dataset

The PD dataset used in this chapter (called PD2T) contains video data from 25 PD patients tested longitudinally at 8 week intervals over time. Subjects were between the ages of 41 to 72 years and performed UPDRS tasks and their scores were assigned by trained clinical raters. Videos were captured at 25fps at a resolution of 1920×1080, using a single RGB camera (SONY HXR-NX3). Our dataset consists of 1058 videos spanning

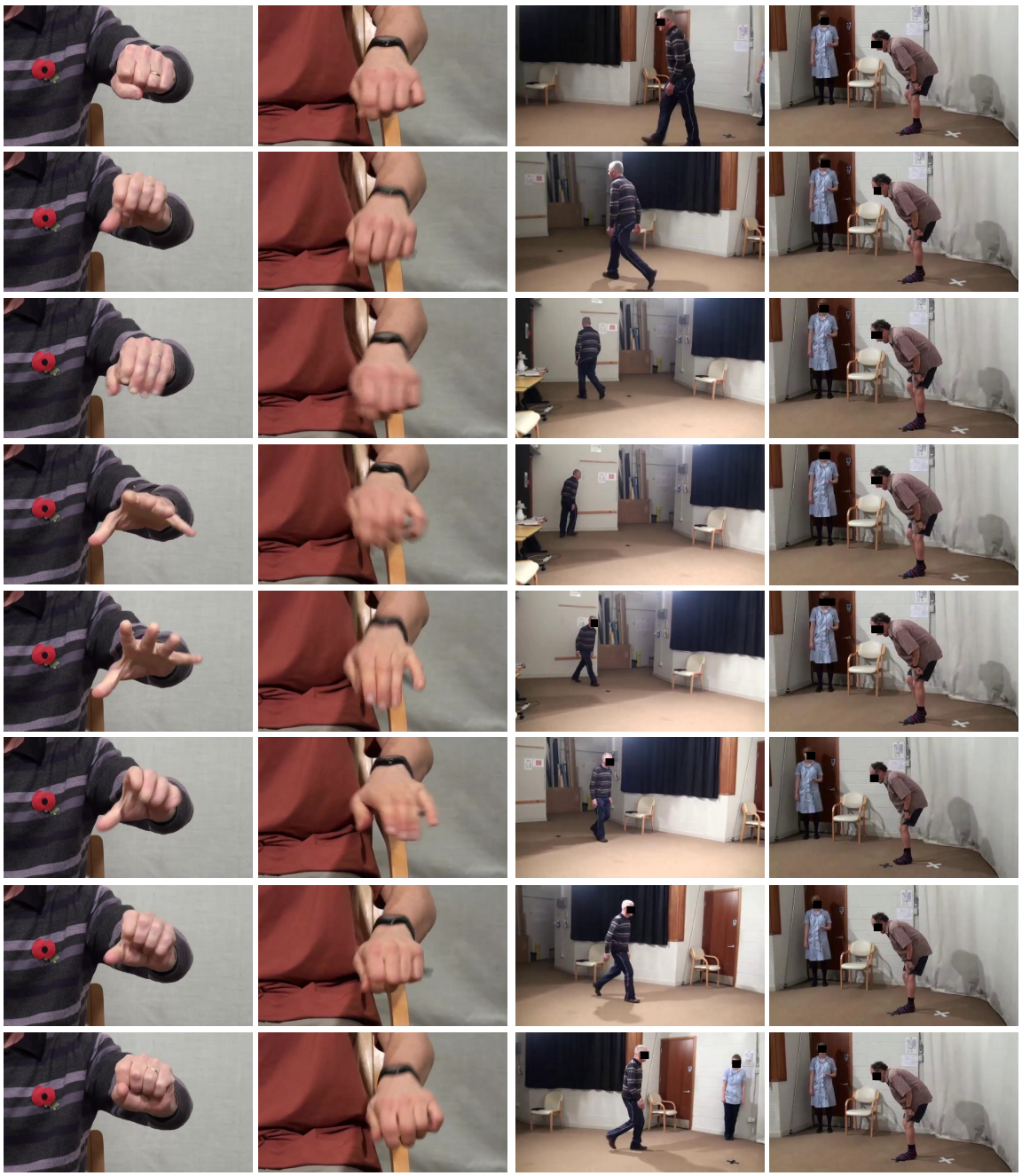


Figure 3.1: *Sample Frames from our PD dataset: The first two columns represent hand movements with varying levels of severity, showing the intricacies of fine motor skills affected by PD. The last two columns illustrate gait patterns in patients, also with different levels of severity. All videos in this dataset are from actual PD patients and were recorded at Southmead Hospital in Bristol, UK, within a clinical setting, over the course of several months as part of a clinical study.*

two different UPDRS tasks: hand movement and gait. In the first task, the patients had to open and close their hand (each hand separately) 10 times, as fully and as quickly as possible. The second task is gait analysis in which the patients walked 10 metres at a

comfortable pace and then returned to their starting point. Table 3.1 shows the number of videos in each of our score classes, as well as their minimum/maximum number of frames for each UPDRS task. To facilitate a more effective training and evaluation process of our machine learning models, we have simplified the categorisation of UPDRS scores into three labels: Normal (0), Mild (1-2), and Severe (3-4). This categorisation was strategically chosen to consolidate the dataset into clinically significant groups that reflect the gradations of severity in PD, thus enabling the model to learn with a more balanced distribution of classes. Furthermore, to further mitigate the impact of class imbalance during model training, we employed a focal loss function, as detailed in Section 3.3.3. Figure 3.1 shows sample frames from our dataset, PD2T, selected from four subjects with different PD severity levels performing hand movement and gait tasks.

3.2.2 Data Protection and Ethical Approval

PD2T dataset has been collected, stored, and used in strict compliance with all relevant data protection laws and guidelines. This compliance includes ensuring patient confidentiality and the secure handling of personal data. All participants in the study provided informed consent, being fully aware of the nature and purpose of the research.

To ensure the privacy and rights of the participants, any identifiable information has been anonymised in the dataset, making it impossible to directly or indirectly identify individual subjects through the data.

In addition to these measures, the data has been securely stored in the University of Bristol's Research Data Storage Facility (RDSF). This facility provides an additional layer of security, ensuring that only authorised people can access the data. Please note PD2T has full ethics approval from the relevant committees, ensuring all research conforms to the highest ethical standards.

Table 3.1: *Details of PD2T dataset.*

	Hand movement		**Gait**	
	#video	**#frame min/max**	**#video**	**#frame min/max**
Normal (0)	180	131/312	171	473/980
Mild (1-2)	500	123/717	180	580/5007
Severe (3-4)	24	202/1210	3	1367/3012

3.3 Proposed Approach

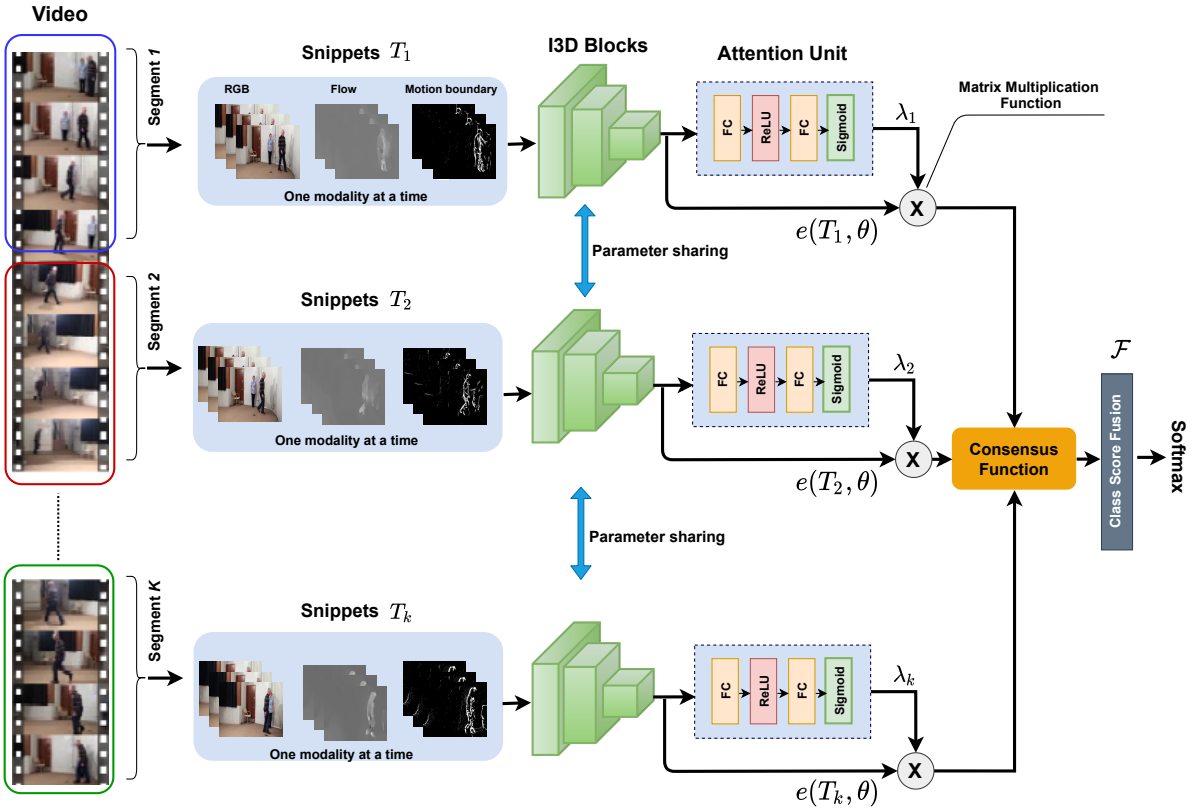


Figure 3.2: *Architecture of the proposed method for PD severity assessment task. The whole model can be trained in an end-to-end manner by only one loss function. The main steps are as follows: (i) Extracting spatial and temporal feature representations from K video snippets using a single I3D network that shares all of its weights with the other branches. (ii) Computing an attention weight for each video snippet by an attention unit. (iii) Weighting each feature vector by its corresponding attention weight before being forwarded to the consensus function, (iv) Using a Softmax layer to output class score predictions. Note that at every training and testing process, the network takes one input modality amongst RGB, optical flow and motion boundaries.*

Our aim is to learn an end-to-end, deep learning model for movement disorder severity assessment in Parkinson's patients, without resort to joint data or elaborate annotations. Given a video from a patient in the clinic performing a UPDRS test task, such as hand opening and closing, our model exploits the motion information in the scene to predict a score depending on how well the task was carried out. Our only annotation is the UPDRS score for the test, as determined by an expert clinical neuroscience rater. Figure 3.2 illustrates an overview of our network and approach. In the following, we explain the details of our method and its training procedure.

3.3.1 Network architecture

Our proposed network architecture consists of three main components: sparse temporal sampling, backbone network, and attention unit. Next, we provide detailed descriptions and justifications for each component.

Sparse Temporal Sampling – In order to facilitate efficient training of our model, we adopt a sparse temporal sampling technique inspired by [149]. As illustrated in Figure 3.2, the first phase involves breaking the video into K distinct segments. From each of these segments, a short snippet is randomly selected, forming a sparse representation of the entire video, represented by K snippets $\{T_i, i = 1..K\}$. Each snippet is produced in three formats: RGB, flow, and motion boundaries. Then, similar to [90], we apply a 3D CNN as the backbone of this framework to directly learn spatial and temporal features from video snippets.

Backbone – While 3D CNNs are inherently designed to capture spatio-temporal features in video data, they come with their own set of challenges, including a large number of parameters and an increased risk of overfitting. To mitigate these issues, we employ I3D [10] as the 3D CNN backbone of our framework. I3D inflates 2D convolution filters from Inception V1 architecture [137] to 3D, effectively reducing the number of parameters while maintaining depth. This innovative inflation technique enables the I3D model to capture spatio-temporal features more efficiently, while preserving the architectural benefits of the original Inception V1. The architecture of the I3D model is shown in Figure 3.3.

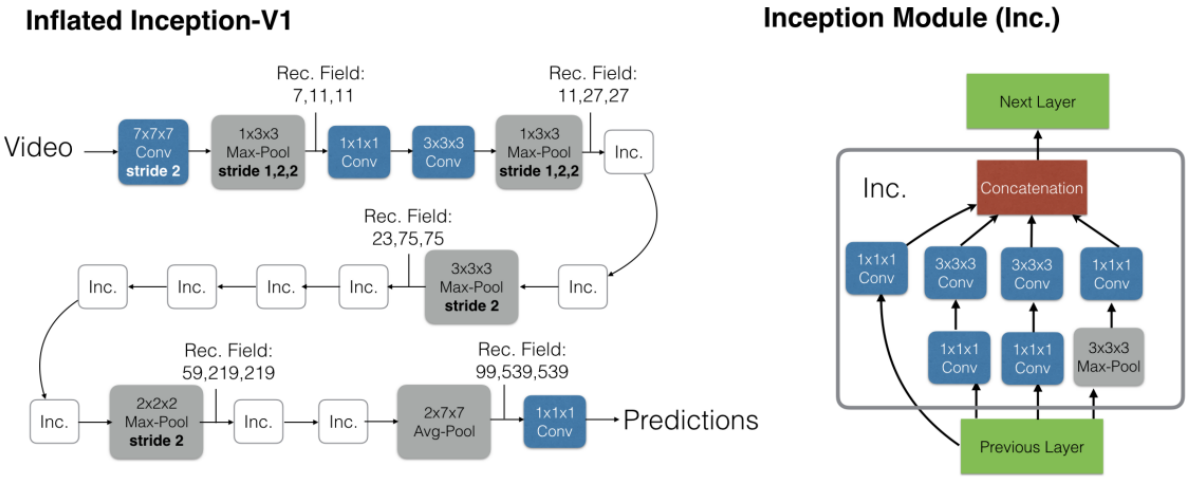


Figure 3.3: *Architecture of I3D Model: The left diagram shows the overall I3D model flow and the right diagram provides a detailed view of the Inception modules (Inc.). The figure is taken from [10].*

Attention Unit – The spatial and temporal feature maps of the last convolutional layer of I3D for each video snippet feed into an attention mechanism. The architecture of the attention unit comprises two fully connected (FC) layers separated by a ReLU activation function to introduce non-linearity. This is followed by a Sigmoid activation function. The output of the Sigmoid function is the attention weight λ ($0.0 \leq \lambda \leq 1.0$) for each video snippet. This is based on the attention module proposed in [105].

The role of these attention weights is to modulate the feature maps so that the model can focus on more informative parts of the video when making a decision. Specifically, in the forward pass of the system, the encoded, attention-weighted features are used to modulate the global average pooling and therefore compiled via the consensus function $C(.)$ to produce class score fusion $\mathcal{F}$ of length M over K video snippets,

$$\mathcal{F} = C(.) = \frac{\Sigma_{i=1}^{K}(\lambda_i e(T_i, \theta))}{K}, \tag{3.1}$$

where $e(.)$ is the encoding function and θ are the network parameters. A Softmax on $\mathcal{F}$ then provides the probability distribution p of the UPDRS class scores of the video clip, i.e.

$$p = \frac{\exp \mathcal{F}_i}{\Sigma_{j=1}^{X} \exp \mathcal{F}_j} \ . \tag{3.2}$$

3.3.2 Motion Boundaries

Previous works, such as [10, 133, 149], have shown the importance of using optical flow in deep learning-based human action recognition. Optical flow computes the motion between two frames, giving a dense map of apparent motion patterns. It effectively captures how and where objects move in videos, which is important to recognise actions [133]. A significant limitation of optical flow is its representation of absolute motion. This means it captures all movements in the frame, including those caused by camera motions. [12]. Wang et al. [149] proposed to use warped flow [147] to cancel out the camera motion. However, warped flow did not results in a better performance than normal optical flow in their work. Moreover, computing this modality can be computationally very expensive [147].

To address this problem, we need a new input stream that better encodes the relative motion between pixels. Our solution involves using motion boundaries. Originally introduced for human detection tasks in [25], motion boundaries is designed to remove constant motion from the scene. This inherently removes the effect of camera move-

ments, allowing the model to focus more on the relevant motion within the frame, such as the specific action of a patient with PD.

In a similar fashion to [25], we compute motion boundaries simply by a derivative operation on the optical flow components, as shown in Figure 3.4. Formally, let $u_x = \frac{\partial u}{\partial x}$ and $u_y = \frac{\partial u}{\partial y}$ represent the x and y derivatives of horizontal optical flow, and $v_x = \frac{\partial v}{\partial x}$ and $v_y = \frac{\partial v}{\partial y}$ represent the x and y derivatives of vertical optical flow respectively. Then, for any frame j,

$$B_u^j = f(u_x^j, u_y^j), B_v^j = f(v_x^j, v_y^j), \tag{3.3}$$

where B_u represents the motion boundary in horizontal optical flow u, and B_v represents the motion boundary in vertical optical flow v, and f is a summing function. It is clear that, for a video clip with N frames, $(N-1)*2$ motion boundary frames are computed.

3.3.3 Class imbalance

In PD2T dataset used in this study (see details in Table 3.1), the number of videos belonging to UPDRS scores 3 and 4 is significantly lower than those belonging to the other classes. Therefore, we have a class imbalance problem which can lead to a model biased towards the classes with large number of samples.

In order to mitigate this problem, we apply two strategies. In the first, we group the scores into three classes: score 0 for normal subjects - i.e. patients who are at very early stage of PD and may still have one unaffected upper limb, score 1-2 for subjects with mild symptoms, and score 3-4 for subjects with severe symptoms. In the second strategy, we utilise an extended version of the normal class entropy loss, called focal loss [86], to train our multi-class classification task.

While the traditional cross-entropy loss has been standard for classification tasks, it can be problematic when there is a class imbalance in the dataset. In such cases, the majority class can dominate the training, resulting in sub-optimal learning for the minority class. The focal loss addresses this challenge by giving more emphasis to the harder, misclassified examples and reducing the contribution from easily classified instances. This ensures that the model does not become biased towards the dominant class. In the context of our multi-class PD severity assessment, where some severity levels might be less common, the introduction of focal loss ensures a balanced learning approach across all

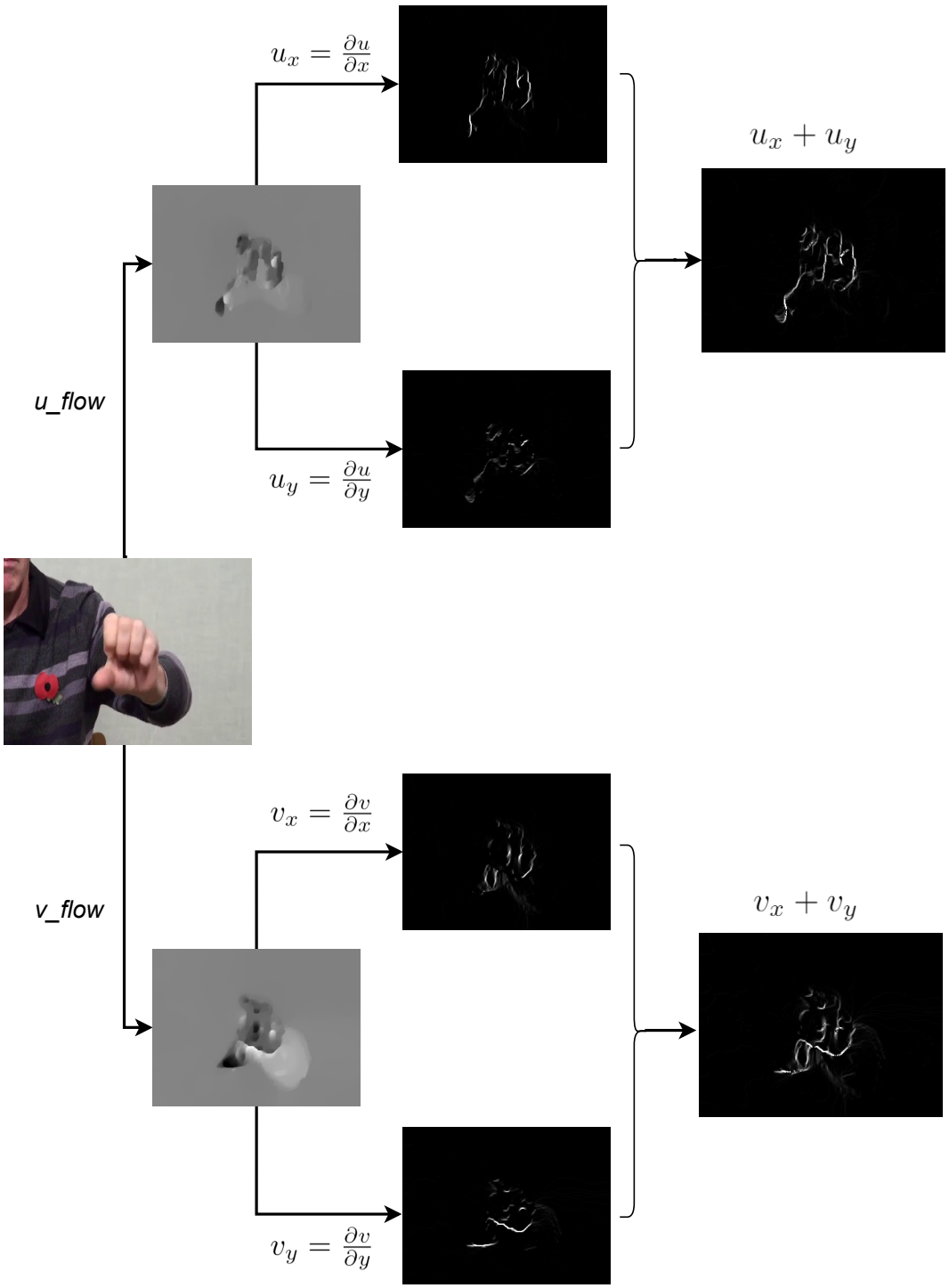


Figure 3.4: *Motion boundary computation from optical flow components u and v. For each flow component, we compute two motion boundaries via derivatives for the horizontal and vertical flow components. Then the final motion boundaries are obtained by their sum. It is clear that optical flow contains constant motion in the background which is removed after computing motion boundaries.*

classes. Our loss function is expressed as

$$\mathcal{L}(y, p) = -\alpha(1-p)^{\gamma} y \log p \ , \tag{3.4}$$

The loss function combines two essential elements: the modulating factor, $\alpha(1-p)^{\gamma}$, and the conventional cross-entropy term. The parameter γ adjusts the rate at which easy samples are down-weighted. Specifically, for samples that the model classifies with high confidence (large p), their contribution to the loss is minimal. However, for challenging instances that are often misclassified (small p), the modulating factor amplifies their impact on the overall loss, compelling the model to focus more on them. The factor α offers a way to weight each class in the loss computation, a feature that becomes critical in scenarios with pronounced class imbalances. When $\alpha = 1$ and $\gamma = 0$, focal loss essentially becomes the standard cross-entropy loss.

3.4 Experiments

In this section, we provide the experimental setup and our detailed ablation study of various aspects of our model. Finally, we compare our model with the state-of-the-art models, primarily those based on 3D CNNs, proposed for human action recognition.

3.4.1 Experimental Setup

Implementation Details – The input videos were reduced to a resolution of 340×256 pixels. We used Pytorch to implement our models and TV-L1 [167] for computing optical flow fields. The focal loss (Eq. 3.4) parameters were set to $\alpha = 0.5$ and $\gamma = 2$ for all experiments. We applied Adam optimization with a learning rate of 0.00001, and batch size 2 to optimize our model parameters. Dropout was applied with a ratio of 0.7 before the output layer of our I3D network. All models were trained for 120 epochs using one Nvidia RTX 2048TI GPU under Cuda 10.1 with cuDNN 7.6.

Training and Testing Details – Each video was split into $K = 4$ equal segments along the temporal axis. Preserving chronological order, we randomly sampled 32 frames within each video segment as a snippet. The length of our snippets is relatively larger than the length of snippets used in [149]. We verified empirically that for our PD task sampling these larger snippets can provide more application-specific motion characteristics to our network.

Since in the training step all I3D models share their parameters, our trained model behaves like the original I3D network [10] during testing. Therefore, we did not use

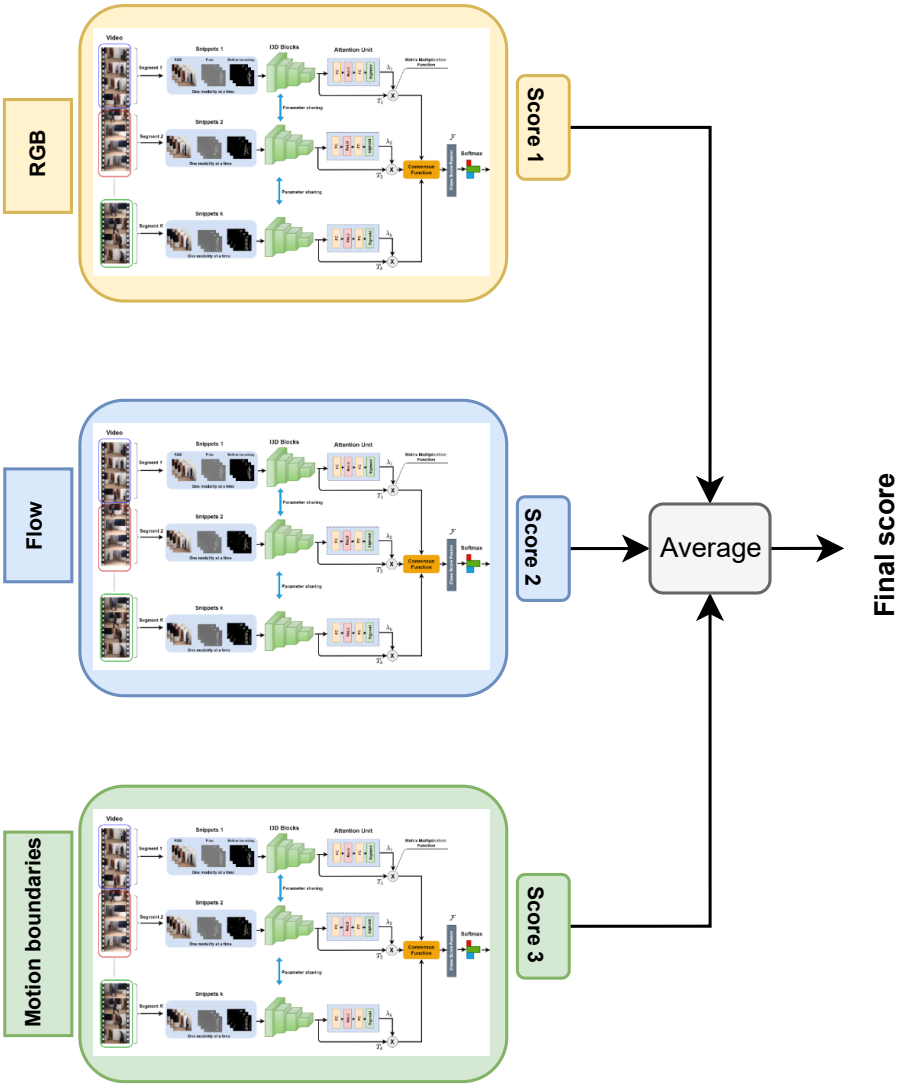


Figure 3.5: *An overview of our multi-stream configuration. We train our model with each different input modality separately and then use a late fusion approach at test time to average over all predicted scores.*

temporal sampling when testing our model, allowing us to draw fair comparison with other models who also tested without temporal sampling, such as [10, 38, 133]. In particular, during inference we used 64 non-sampled snippets per video, each containing 16 consecutive frames. The prediction scores of all these snippets were then averaged across each or combined modalities to get a video-level score (as illustrated in Figure 3.5). Note, this follows the same approach as [10] where RGB and Flow were averaged at test time.

Given the constrained size of our dataset, we initialised I3D by the weights pretrained on Kinetics [72], a large-scale, high-quality dataset. This simple transfer learning approach provides our network with generic and rich spatial and temporal features, making it less

Table 3.2: *F_1 score results of our proposed network for both hand movement and gait tasks with different input modalities, with and without attention units. The last column shows the average results across both tasks. All results are given in %.*

	Hand Movement		**Gait**		**Average**	
Input Modalities	+att.	−att.	+att.	−att.	+att.	−att.
RGB	68.4	65.2	74.8	73.7	71.6	69.4
Flow	71.0	68.6	76.5	74.1	73.7	71.3
Motion Boundaries	**72.3**	70.0	76.8	76.5	**74.5**	73.2
RGB + Flow	69.9	68.8	76.2	75.1	73.0	71.9
RGB + Motion Boundaries	70.4	70.1	75.4	72.3	72.9	71.2
Flow + Motion Boundaries	71.7	71.7	**77.1**	76.2	74.4	73.9
All Modalities	71.1	70.2	**77.1**	75.1	74.1	72.6

dependent on our smaller dataset for feature extraction and reducing the potential for overfitting. Furthermore, we incorporated a set of data augmentation techniques for all frames within each training snippet. These included scale jittering, which adjusts the scale of the image to introduce variability; corner cropping, a technique that involves cropping the frame from its corners to diversify the viewpoint; and horizontal flipping, which mirrors the video frames.

Evaluation Metrics – We used 5-fold cross validation for 5 batches (given our 25 patients). This approach yields an unbiased evaluation, since each patient's data undergoes both training and validation phases multiple times. Cross-validation is also important in scenarios with limited data, as it maximises the training and validation utility of each sample. We use F_1 score to report the model's performance, which is computed over the average validation scores.

3.4.2 Results including Ablation Study

Choice of Input Modalities – The results presented in Table 3.2 offer a comparative evaluation of different input modalities. For the Hand Movement task, using Motion Boundaries alone yields the highest F_1 score at 72.3%, which represents an increase of 3.9% and 1.3% over the RGB and Flow modalities, respectively. This suggests that Motion Boundaries effectively capture the characteristic movements that are critical for this task. However, when Motion Boundaries are combined with RGB and Flow, there is a

Table 3.3: *Comparison of our method with different state-of-the-art architectures. MBs is for Motion Boundaries and all results are given in %.*

Model	RGB	Flow	MBs	Hand Movement	Gait	Average
Two-Stream [133]		✓	✓	60.3	56.7	58.5
TSN [149]		✓	✓	70.1	75.7	72.9
I3D [10]		✓	✓	69.1	73.1	71.1
SlowFast [38]	✓			67.1	66.9	67.0
TSN + SlowFast	✓			68.4	68.9	68.6
Proposed Method w/o Focal loss			✓	70.7	75.7	73.2
Proposed Method		✓	✓	71.7	**77.1**	74.4
Proposed Method			✓	**72.3**	76.8	**74.5**

nuanced improvement for the RGB to 70.4% but a more noticeable enhancement for the Flow to 71.7%. The relatively modest increase when combining modalities could imply that while Motion Boundaries add value, the network may not be fully capitalising on the combined feature set, potentially due to the added complexity or noise introduced by the additional modalities. For the gait task, characterised by more pronounced dynamic movement spatiotemporally, the combination of all modalities performs well, yet it is the Flow combined with Motion Boundaries that achieves the peak performance at 77.1%. Overall, the inclusion of Motion Boundaries demonstrates their merit as an additive modality; however, the results also indicate that the expected synergistic effect of combining modalities does not always translate to improved performance, possibly due to the increased complexity and noise in the data which may not be optimally managed by the model.

Effect of Attention – To study the influence of the attention units, we perform all our experiments with and without them. As seen in Table 3.2, in all experiments for hand movement and gait tasks, our model achieves better accuracy *with* the attention units. For example, when considering all modalities, attention improves the average accuracy across both tasks by 1.5%, with a notable increase of 2.0% for the gait task alone. Again, even without attention units, Motion Boundaries play a significant role in improving the results over other modalities.

Performance of Other Architectures – Table 3.3 provides the F_1 percentages of other architectures adapted to provide a UPDRS score for our application. We used the

same data augmentation strategy with focal loss to train all models. All the network weights were initialised with pretrained models from Kinetics-400, except for the SlowFast network, as one of the properties of this model is training from scratch without needing any pretraining. Although we examined the performance of these architectures for all possible input modalities, we only report here their best results, again except for the SlowFast network, as this model is only based on RGB input. Thus, for example for I3D [10], its best result is when using Flow and Motion Boundaries. As shown in the table, our proposed approach performs better than these popular networks for both hand movement and gait tasks.

Effect of focal loss – The importance of using our focal loss (Eq. 3.4) is also shown in Table 3.3 where the performance of our method when using a categorical cross-entropy loss results in an average drop of $\downarrow$1.3% compared to the full focal-loss based result of 74.5%.

3.5 Conclusions

In this chapter, we introduced an end-to-end network aimed at assessing the severity of PD using video data. We focused on two key tasks of the UPDRS: hand movement and gait. Our approach builds upon an inflated 3D CNN trained by a temporal sampling strategy to effectively capture long-term patterns without a significant computational burden. Furthermore, we incorporated an attention mechanism along the temporal dimension to ensure our model prioritises the most relevant segments for a more accurate severity assessment. Recognising the challenges posed by constant camera motion, we proposed the use of motion boundaries as a viable input modality to suppress constant camera motion and showed its effect on the quality of the assessment scores quantitatively. Our results show the potential of our proposed model and highlight the value of integrating advanced techniques, such as attention mechanisms and motion boundaries, for improved accuracy in PD assessments.

While the proposed framework showed promise for Parkinson's disease assessment, there are notable limitations. First, due to the distinct training requirement for each modality, the computational cost can be considerably high, which can be a problem, especially in settings that may lack the budget for such resources. Another limitation of our approach is that it is unable to handle several UPDRS tasks in one training process, and hence we need to train and evaluate our model on each task separately. Additionally, our approach is inherently supervised, and it requires large amounts of labelled data for accurate performance. In real-world scenarios, acquiring such annotated data can be both time-

consuming and expensive, posing challenges in scalability and adaptability to diverse clinical environments. To address these limitations, future research could explore the integration of Recurrent Neural Network (RNN), particularly Long Short-Term Memory (LSTM) networks. LSTMs are well-suited for processing sequential data and could potentially enable more efficient handling of multiple tasks concurrently. This approach might also reduce the dependency on extensive labelled datasets. A hybrid model that combines the strengths of 3D CNNs and LSTMs may offer a more computationally efficient and potentially more accurate framework for assessing the severity of Parkinson's disease.

Chapter 4

Auxiliary Learning for Self-supervised Video Representation Learning

4.1 Introduction

Deep learning models generally require large amounts of well-annotated data to achieve high levels of performance and generalisation. However, when it comes to specialized tasks like Parkinson's disease, preparing a large-scale annotated dataset is challenging for two main reasons: i) the acquisition of such datasets is inherently expensive and time-consuming, especially since clinicians are needed to provide detailed and accurate annotations; ii) ethical concerns around patient privacy and data security need additional procedures and paperwork.

In Chapter 3, we developed a model to address the challenges of our sparsely labeled PD dataset by initialising it with supervised pretrained weights from a generic, large-scale dataset (Kinetics-400). While this approach is better than training from scratch, deploying such pretrained models might be sub-optimal for AQA tasks (e.g. PD assessment) due to the domain/task discrepancy between action recognition and AQA. For example, in PD action performance scoring, one or two interruptions in the regular rhythm of a patient's movement while performing an action can result in a different quality score. This contrasts with the source pretraining task, where subtle or even more pronounced differences in performing an action should not affect the action classification. In view of these obstacles, this chapter aims to address a crucial question: Is it possible to utilise the inherent, unlabeled information in our target video data itself to facilitate learning?

This concern leads us to explore SSL methods, which eliminate the cost of annotating large-scale datasets by acquiring high-level semantic visual representations from unlabeled data [16, 39, 55, 73, 148]. However, it is essential to note that despite the efficiency of SSL in utilising unlabeled data, there are still challenges to be addressed. The performance improvements attributed to recent SSL techniques are contingent upon the availability of extensive unlabeled datasets, such as Kinetics-400 [16, 39, 55, 73, 148]. This scale-dependency represents a obstacle when applying these methods to low-resource environments such as PD severity assessment.

In addition, the computational and memory requirements for these SSL methods are quite high. For example, it takes around two weeks to train MoCo [55] on Kinetics-400 for 300 epochs with two Nvidia RTX 2080TI GPUs. In fact, such computational costs place many state-of-the-art SSL approaches only in the realms of huge corporations who have such powerful resources [16, 47, 111], and this further becomes a subject of ethical fairness as well as carbon emission footprints [135].

A few recent works have addressed the issue of efficient pretraining for *image-based* tasks [57, 91, 121], but there is only Lin et al. [88]'s work for *video-based* tasks which improves the generalisation performance of a contrastive learning-based method [141] under a meta-learning paradigm. However, their method is not suited to most of the state-of-the-art works that use transformation-based pretext tasks [39, 68, 98, 146, 148].

Our motivation is to develop a *task-agnostic pretraining process that alleviates the dependency on large-scale datasets* for self-supervised video representation learning, while ensuring the model generalises well and still contains rich information. To achieve this, we propose an auxiliary pretraining stage based on knowledge distillation. Incorporating this stage enhances the adaptability and generalisation capabilities of our primary SSL stage, particularly when using smaller-scale datasets (e.g. PD dataset/Kinetics-100 instead of Kinetics-400). Please note that both the auxiliary and primary stages operate on the same dataset.

Figure 4.1 illustrates the difference between our framework and existing SSL methods, such as [16, 39, 68, 98, 148]. We employ a slowly progressing teacher model to iteratively distill knowledge to the student, our self-supervised model, by evaluating the similarity information of an augmented view of a query video clip to a large queue of random clips as anchors and transferring that information to the student. We believe that this Similarity-based Knowledge Distillation (SKD) approach leverages the inherent structure and continuity present in video data, which is pivotal for learning robust video representations. By focusing on similarity, the model is encouraged to understand and

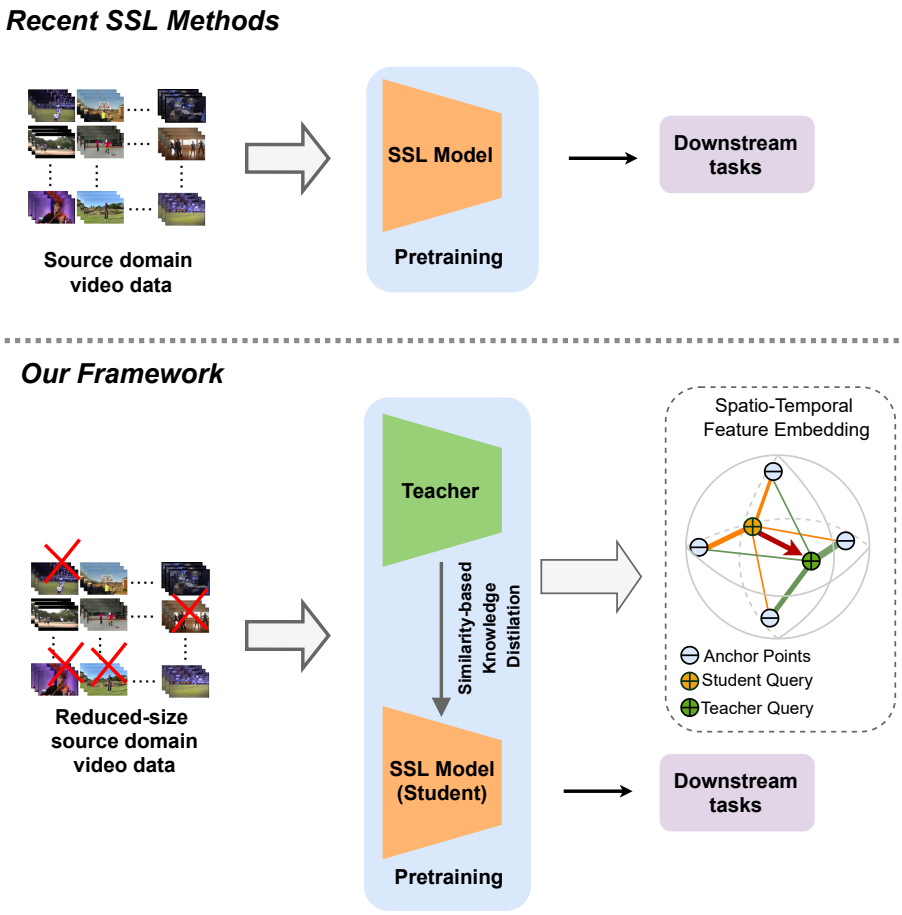


Figure 4.1: *High-level overview of of our framework and recent SSL methods – while recent methods encourage their model to solve a pretext task from scratch, our SSL model benefits from an implicit similarity-based knowledge, distilled by a teacher model, before solving the pretext task. However, the question that we pose in this chapter is: can we use an implicit knowledge of this type to improve the generalisation ability of self-supervised approaches?*

encode temporal dynamics and contextual nuances, which are critical in video understanding.

To our knowledge, this work represents the first implementation of SKD in the domain of video-based self-supervised learning. While SKD has recently been adopted in image-based applications, such as for contrastive learning [139, 140] and model compression [2, 36], its application in analysing video data is a novel approach. It is important to highlight that, although our SKD architecture is inspired by [139], our objective differs significantly. We utilise distilled similarity representations as auxiliary knowledge to enhance various self-supervised pretraining methods, rather than directly applying them to downstream tasks. To refer to this aspect of our work, we use auxSKD.

To support the operation of the proposed approach on temporal features in the video

domain, we apply temporal augmentations, in addition to spatial augmentations, to generate different transformed versions of a query video. Such temporal transformations are the same as the pretext transformations used in the primary pretraining stage. Their application at this stage allows our teacher to impart knowledge which matters most in the primary pretraining stage.

Also in this chapter, we propose a new pretext task for video representation learning, namely Video Segment Pace Prediction (VSPP) . While recent video playback rate prediction methods randomly sample training clips at different paces or speeds [9, 148], we sample training clips where only a randomly selected segment of the video has a randomly selected speed and the other segments of the video retain their natural pace. VSPP then requires the learner model to predict the playback speed of this randomly selected segment and its temporal location in the input training video. We advocate that by solving this pretext task, our model can strengthen its awareness of the natural pace of the clip and deal with the imprecise video speed labeling problem [16]. The work in this chapter was published in [23].

In Section 4.2 we elaborate on our proposed SSL framework that includes an auxiliary pretraining stage and a new pretext task, designed to make video representation learning more efficient in low-resource settings. Section 4.3 presents experiments in the action recognition domain, using UCF101 [134] and HMDB51 [75] datasets. Section 4.4 shifts focus to PD severity assessment, where we conduct experiments on various PD tasks and introduce our PD4T dataset, which includes four different PD motor functions: gait, finger tapping, hand movement, and leg agility. The conclusions are given in Section 4.6.

4.2 Proposed Method

Our goal is to reduce the pretext training computational burden by developing an auxiliary pretraining phase that assists the primary pretext task to learn as efficient generalised self-supervised video representation as possible on a reduced-size source dataset. To achieve this, we take inspiration from Similarity-based Knowledge Distillation which is used in recent works [2, 36, 139, 140]. We illustrate our full self-supervised pretraining framework in Figure 4.2.

4.2.1 Auxiliary Learning via auxSKD

Our auxiliary learning framework consists of a teacher $\mathcal{T}$ and a student $\mathcal{S}$ with the same architecture followed by a fully-connected layer, as the projection, to map the

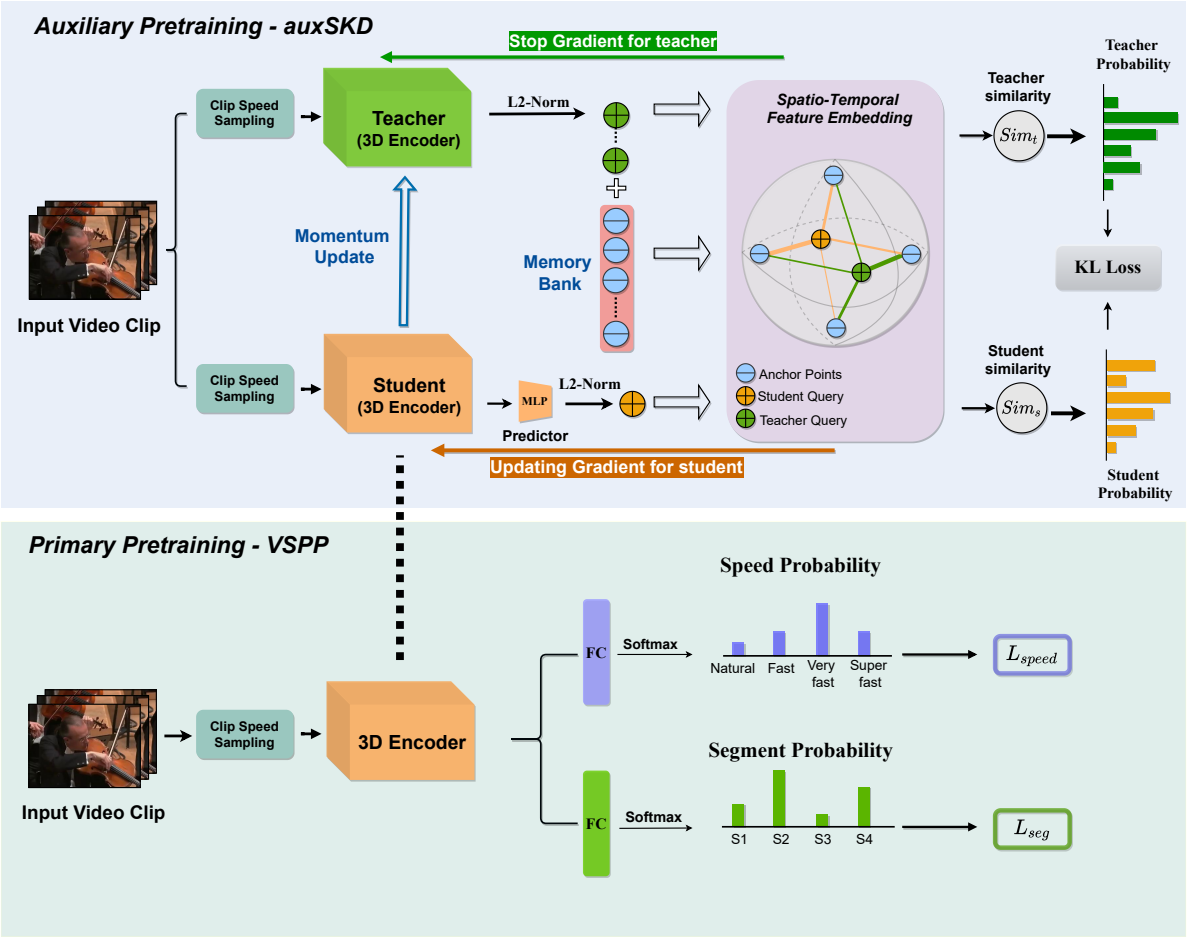


Figure 4.2: *The self-supervised learning pretext training scheme is supported by an Auxiliary Pretraining task (auxSKD - see top region) that provides a similarity knowledge distillation process via a teacher-student configuration. In this configuration, both the teacher and the student 3D encoders are initialised and trained from scratch. Our teacher encoder is updated using momentum as a moving-average of the student weights. We train the student via gradient update by minimising the KL divergence between the two probabilities from the teacher and the student for a transformed version of input video v, computing its similarity over anchor points. Note that in each iteration our encoders randomly take a different transformed input via our clip speed sampling process (see section 4.2.2). In the primary pretraining task (see bottom region), the student is ready to solve our VSPP task on input clips with segments that include changed pace.*

representations into a lower dimension space. We follow BYOL [47] and use a MLP predictor layer on top of the student model to establish an asymmetric architecture between the teacher and student [1]. We randomly initialise both models from scratch equally. The student model and its predictor layer are updated by back-propagation

[1]This design choice mitigates the risk of 'collapsed solutions', where diverse input representations become indistinguishably similar, leading to poor performance on downstream tasks.

while momentum update [55] is applied in the teacher model to be a running average of the student.

At each iteration our pretext task transformation VSPP is applied twice to a raw video instance v to generate two video clips v_1^* and v_2^* independently, with the goal of maximizing their similarity in our teacher-student framework. Then, given feature encodings $(\mathcal{T}(v_1^*), \mathcal{S}(v_2^*))$ and predictor function MLP(.) for $\mathcal{S}$, we perform L_2 normalisation such that $z^{\mathcal{T}} = \mathcal{T}(v_1^*)/\|\mathcal{T}(v_1^*)\|_2$ and $z^{\mathcal{S}} = \text{MLP}(\mathcal{S}(v_2^*))/\|\text{MLP}(S(v_2^*))\|_2$.

Similar to [55, 139], we consider a memory bank of H feature vectors (or anchors) $x_i^{\mathcal{T}} = [x_1^{\mathcal{T}}, ..., x_H^{\mathcal{T}}]$ obtained from the teacher model under a simple FIFO strategy. Specifically, at each iteration, we enqueue the feature vectors of the current batch extracted from the teacher model and dequeue the earliest instances. Next, we calculate the similarity of the teacher's embedding $z^{\mathcal{T}}$ to all feature vectors in the memory bank and apply Softmax to obtain a probability distribution,

$$p_i^{\mathcal{T}} = -\log \frac{\exp(\text{sim}(z^{\mathcal{T}}, x_i^{\mathcal{T}})/\gamma^{\mathcal{T}})}{\sum_{j=1}^{H} = \exp(\text{sim}(z^{\mathcal{T}}, x_j^{\mathcal{T}})/\gamma^{\mathcal{T}})} , \tag{4.1}$$

where $p_i^{\mathcal{T}} = [p_1^{\mathcal{T}}, ..., p_H^{\mathcal{T}}]$ is the probability of teacher query $z^{\mathcal{T}}$ for the i-th anchor point, sim(.,.) measures the similarity between L_2 vectors, and $\gamma^{\mathcal{T}}$ is the temperature value for the teacher's model.

Similarly, we calculate the student similarity distribution $p_i^{\mathcal{S}} = [p_1^{S}, ..., p_H^{S}]$ over anchor points, with

$$p_i^{\mathcal{S}} = -\log \frac{\exp(\text{sim}(z^{\mathcal{S}}, x_i^{\mathcal{T}})/\gamma^{\mathcal{S}})}{\sum_{j=1}^{H} = \exp(\text{sim}(z^{\mathcal{S}}, x_j^{\mathcal{T}})/\gamma^{\mathcal{S}})} . \tag{4.2}$$

Here $\gamma^{\mathcal{S}}$ is the temperature value for the student's model. Finally, the loss is measured by the Kullback–Leibler (KL) divergence as

$$\mathcal{L}(\mathcal{T}, \mathcal{S}) = \sum_i \text{KL}(p_i^{\mathcal{T}} \parallel p_i^{\mathcal{S}}) . \tag{4.3}$$

Note that during training, the teacher network's weights are initialised randomly and then they evolve gradually as a running average of the student using momentum with the update rule $\theta_{\mathcal{T}} \leftarrow m\theta_{\mathcal{T}} + (1-m)\theta_{\mathcal{S}}$, where $m \in [0, 1)$ is the momentum hyperparameter to ensure smoothness and stability, and $\theta_{\mathcal{T}}$ and $\theta_{\mathcal{S}}$ are the teacher and student model parameters respectively. Pseudo-code for our auxSKD training is provided in Algorithm

Input: Teacher model $\mathcal{T}(.,\theta_{\mathcal{T}})$ and student model $\mathcal{S}(.,\theta_{\mathcal{S}})$, videos $V = \{v_i\}_{i=1}^{N}$, and memory bank $\{x_i^{\mathcal{T}}\}_{i=1}^{H}$.

Output: Trained student model weights $\theta_{\mathcal{S}}$.

Randomly initialise $\mathcal{T}(.,\theta_{\mathcal{T}})$ and $\mathcal{S}(.,\theta_{\mathcal{S}})$.
while not max epoch **do**
    Randomly sample a video v from V.
    Sample two clips v_1^* and v_2^* from v using VSPP (Section 4.2.2).
    Compute student query features $z^{\mathcal{S}}$ from clip v_1^* using student model $\mathcal{S}(.,\theta_{\mathcal{S}})$.
    Update teacher parameters using momentum: $\theta_{\mathcal{T}} \leftarrow m\theta_{\mathcal{T}} + (1-m)\theta_{\mathcal{S}}$.
    Compute teacher query features $z^{\mathcal{T}}$ from clip v_2^* using teacher model $\mathcal{T}(.,\theta_{\mathcal{T}})$.
    Calculate $p_i^{\mathcal{T}}$ and $p_i^{\mathcal{S}}$ using Eq. 4.1 and Eq. 4.2.
    Add the teacher's embedding $z^{\mathcal{T}}$ into the memory bank $\{x_i^{\mathcal{T}}\}_{i=1}^{H}$.
    Pop-out the earliest sample from the memory bank $\{x_i^{\mathcal{T}}\}_{i=1}^{H}$
    Optimize student model using KL divergence loss, Eq. 4.3.
end while

Algorithm 4.1: *Training auxSKD*

4.1.

4.2.2 Primary Pretext Task Learning via VSPP

A SSL pretext task encourages the neural network to learn a representation from unlabelled data which contains high-level abstractions or semantics. In the video pace prediction approach of Wang et al. [148], each training clip is randomly sampled at a different pace and their pretext task then identifies the pace for each clip. While this is an effective approach, it means each clip is treated as if its pace is its natural speed.

We propose that each clip should contain within it one segment where the pace has been (randomly) altered. Our assumption is similar to [9, 16, 148] in that the network can only represent the underlying video content through efficient spatiotemporal features if it succeeds in learning the pace reasoning task, however, we build on [148]'s proposal through a more intricate, yet simple, within-video pace alteration task. Our proposed VSPP pretext task requires our model to temporally explore a video clip and predict the index and speed of a segment within a clip which is sampled at a different speed rate.

Given a video clip v_i comprising N frames, we generate video $v_i^* = \{I_0, I_1, ..., I_{K-1}\}$ of size $K<N$, comprising Z segments, such that K/Z number of frames in segment ζ are sampled at pace λ, where both ζ and λ are randomly selected from $1 \leq \zeta \leq Z$ and $1 \leq \lambda \leq Q$ respectively, and Q is the highest possible speed rate. Note, when $Z = 1$ the sampling strategy is similar to [148]. In this work, we select $Z = 4$ and $Q = 4$ to allow a significantly wide variation of starting locations and sudden speed rate changes to provide more precise self-supervision signals. Specifically, our approach results in a change of speed rate in only one segment of the clip while the rest of the clip (before and after) retains its natural rate (see Figure 4.3). This strategy allows the network to better find the difference between natural speed (changes which happen gradually) and altered speed (changes which happen suddenly) in a clip, alleviating imprecise video speed labeling issues [16].

To have a random speed rate λ for the ζ^{th} segment, beginning at frame I_b and ending at frame I_e, then

$$\begin{aligned} I_b &= f_r + ((\zeta - 1) * \frac{K}{Z}) + (\lambda - 1), \\ I_e &= I_b + \lambda * (\frac{K}{Z} - 1) \ , \end{aligned} \tag{4.4}$$

where f_r is the r^{th} frame of the original video v_i which is randomly selected during sampling to generate a more diverse video clip v_i^* at each iteration.

Summary of Our Overall Method – Given a 3D encoder, such as an R(2+1)D or R3D-18, and a video dataset $V = \{v_i\}_{i=1}^N$, we perform our auxiliary learning stage using our flavour of SKD (see Section 4.2 and Algorithm 4.1) based on a KL loss (Eq. 4.3). Following this auxiliary pretraining stage, the student model enters the primary pretraining stage to solve our VSPP pretext task through two simultaneous sub-tasks: i) predicting the speed rate λ in the ζ^{th} segment of v^*, ii) predicting the temporal location

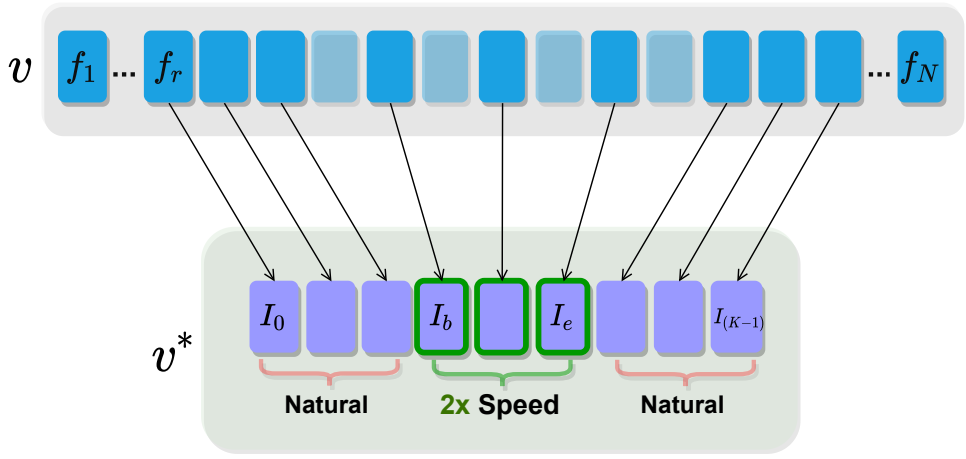


Figure 4.3: *Changing the natural speed of one random segment of a video clip for the pretraining stage - the VSPP pretext task learns where in v^* this occurs and at what speed change. In this example $\lambda = 2$ and $\zeta = 2$.*

of the segment in v^* which is sampled at a different speed, i.e. predicting index ζ. Then by jointly optimizing these two tasks, the final self-supervised loss is defined as

$$\mathcal{L} = \alpha \mathcal{L}_{speed} + \beta \mathcal{L}_{seg}, \tag{4.5}$$

where α and β are balancing weights (empirically found to work best in our experiments when $\alpha = \beta = 1$). $\mathcal{L}_{speed}$ and $\mathcal{L}_{seg}$ are cross-entropy losses.

4.3 Experiments on Action Recognition

4.3.1 Datasets

We conducted our experiments on four datasets, two for pretraining, Kinetics-400 [72] (K-400) and Kinetics-100 [16] (K-100), and two for downstream action recognition, UCF101 [134], HMDB51 [75].

Kinetics-400 is an extensive dataset created for the task of action recognition, which has been widely adopted in the computer vision community. This dataset, gathered from YouTube, includes a diverse range of 400 human action classes. These include, but are not limited to, physical exercises such as running and jumping, sports actions such as playing cricket or basketball, and also intricate activities such as playing various musical instruments or engaging in different cooking techniques. Each action class is represented by approximately 600 video clips, which have been trimmed to approximately 10 seconds. Altogether, the dataset contains around 240,000 video clips, providing a substantial volume of data to train robust deep learning models. The videos are labelled with

a single action class, despite the possibility of containing multiple discernible actions. Kinetics-400's structure and diversity make it an ideal benchmark for models intended to understand and categorise human actions in video data, offering challenges in terms of intra-class variance and inter-class similarity.

Kinetics-100 is created by selecting 100 classes from Kinetics-400, each with the smallest file sizes in the training set, and comprises around 33K videos. We use K-400 and K-100 as pretraining datasets to validate our proposed approach's performance on a reduced-size dataset and promote less dependency on large-scale datasets for self-supervised representation learning in action recognition tasks.

UCF101 is a widely-used dataset for action recognition, consisting of 13,320 video clips that span 27 hours of video data, distributed across 101 diverse action classes. The dataset covers a broad spectrum of activities, encompassing everything from daily life actions such as brushing teeth and hand-washing, to sports activities like basketball and volleyball, and even unique actions such as horse riding and trampoline jumping. These clips have been extracted from realistic, user-uploaded videos on YouTube, which inherently contain camera motion and cluttered backgrounds, presenting a substantial challenge for accurate action recognition. This dataset is divided into three training/testing splits and we follow prior works [16, 148] to use training split 1 for self-supervised pretraining and train/test split 1 for fine-tuning and evaluation.

HMDB51 consists of 6,849 video clips sourced from a variety of platforms such as movies and web content. These clips are categorised into 51 different action types, such as jump, kiss, and laugh, with each category having a minimum of 101 clips. For evaluation, the dataset employs three different train/test splits. In every split, there are 70 clips for training and 30 for testing within each action category. Here, we use split 1 for our downstream task evaluation, similar to [16, 148].

4.3.2 Implementation Details

Backbone Networks – We choose two different backbones R(2+1)D [143] and R3D-18 [52], as our 3D encoder, which have been widely used in recent state-of-the-art self-supervised video representation learning methods [16, 65, 111].

R3D-18 is a specific configuration of 3D Residual Network (ResNet [54]) designed for video analysis tasks. This architecture consists of 18 layers with residual connections, and it performs 3D convolutions on both the spatial and temporal dimensions of the video.

R(2+1)D, short for Residual 2.5D, is also an extension of ResNet that separately han-

dles spatial and temporal dimensions of video data. It decomposes 3D convolutions into a 2D spatial convolution for each video frame and a 1D temporal convolution across frames (see Figure 4.4). This decoupling makes the architecture computationally cheap while effectively capturing both spatial and temporal features.

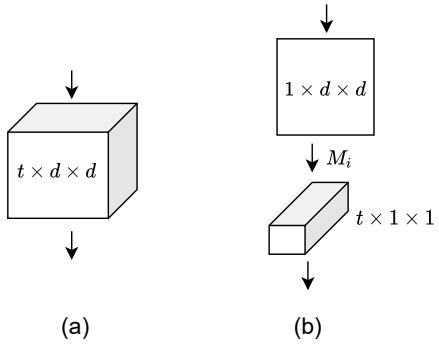


Figure 4.4: *(2+1)D versus 3D Convolution. (a) A 3D convolution utilises a filter sized $t \times d \times d$, with t representing the temporal dimension, and d denoting the spatial dimensions, both width and height. (b) Conversely, a (2+1)D convolutional architecture separates the process into an initial spatial 2D convolution and a subsequent temporal 1D convolution. The quantity of 2D filters (M_i) is determined in such a way that the parameter count in (2+1)D setup is equivalent to that of the full 3D convolutional framework. The figure is taken from [143].*

Default Settings – We run all experiments under PyTorch on two GeForce RTX 2080Ti GPUs with a batch size of 30. We use SGD as our optimiser with momentum of 0.9 and weight decay of 5e-4.

Pretraining Stages – Following [148], for both auxiliary and primary stages, we pretrain our models for 20 epochs with an initial learning rate of 1×10^{-3}. The learning rate is decreased by 1/10 every 6 epochs. For data augmentation, we randomly crop the video clip to 112 $\times$ 112 and then apply horizontal flip and color jittering to each video frame. Following [5], for UCF101 we apply (10x more iterations at) 90K iterations per epoch for temporal jittering. In our auxiliary pretraining stage, we use a predictor head for the student encoder comprising a 3-layer MLP with hidden dimension 1024, and output embedding dimension 128. We do not use a predictor for the teacher and only set its output dimension (projection head) to 128. We follow MoCo [111] and set the size of the memory bank to 16384 and set the momentum value of the encoder update to 0.999. We also use the same temperature for both teacher and student model at 0.02. To use the student encoder for the primary stage, the weights of the convolutional layers are retained after auxiliary pretraining and we drop the projection layer and predictor to replace them with two randomly initialised FC layers corresponding to the segment speed and index outcomes of our VSPP pretext task (see Figure 4.2). We select our

parameters empirically.

Fine-tuning – During fine-tuning, we transfer the weights of the convolutional layers to the human action recognition downstream task, while the last FC layer is randomly initialised. We fine-tune the network on UCF101 and HMDB51 for 25 epoches with labelled videos and apply cross-entropy loss. We use the same data augmentation and training strategy as the pretraining stage except for the initial learning rate which is set to 3×10^{-3}, similar to [148].

Evaluation Settings – We follow the common evaluation protocols on video representation learning [65, 148] to assess the performance of our proposed approach. For action recognition, we sample 10 clips uniformly from each video in the test sets of UCF-101 and HMDB-51. Then for each clip, we only simply apply the center-crop. To find the final prediction, we average the Softmax probabilities of all 10 clips from the video.

4.3.3 Evaluations

Comparison on K400 Pretraining – For completeness sake, and to illustrate how our proposed method fares when pretrained on K400, we present comparative results in Table 4.1 for top-1 accuracy on both UCF-101 and HMDB-51 datasets, along with the pretraining settings for all methods, i.e. backbone architecture, input size, pretraining dataset, and number of epochs. In Rows 1-4, we show a mix of methods that operate on temporal manipulations at different input sizes and on different backbones for reference. Rows 5-10 allow more like-for-like comparisons of recent, popular works in SSL video representation learning based on K-400 pretraining, R3D-18 backbone and almost consistent image sizes across the techniques. ASCNet achieves the most superior results with a combined appearance and speed manipulation approach. In Rows 11-15, where pretraining is on K-400 on the R(2+1)D architecture, RSPNet and VideoMoCo come 1st and 2nd-best alternatively on the two test datasets, while our approach exceeds CEP on both.

Comparison on K100 Pretraining – The results on Rows 16-29 of Table 4.1 represent the essence of our contributions, in that we aim to reduce the dependence of SSL methods on large pretraining datasets, for example by replacing K-400 with K-100 for pretraining. We apply our auxSKD stage to two other transformation-based pretext tasks, i.e. VCOP, VideoPace, and also to one contrastive task, i.e. RSPNet, to exhibit the flexibility of our method. For VCOP and VideoPace, we train their auxSKD with video clips sampled based on VCOP and VideoPace's own sampling strategies, as proposed in [148] and [157] respectively. To integrate auxSKD into the RSPNet framework, we train it with

4.3 Experiments on Action Recognition

Row	Self-Supervised Methods: Method	Network	Input Size	Pretrain	#ep.	Top 1 accuracy: UCF101	HMDB51
1	Shuffle&Learn [100] [ECCV, 2016]	Alexnet	256 × 256	UCF101	-	50.2	18.1
2	OPN [78] [ICCV, 2017]	VGG	80 × 80	UCF101	-	59.8	23.8
3	VCOP [157] [CVPR, 2019]	C3D	112 × 112	UCF101	-	65.6	28.4
4	SpeedNet [9] [CVPR, 2020]	I3D	224 × 224	K-400	n/a	66.7	43.7
5	VideoPace [148] [ECCV, 2020]	R3D-18	112 × 112	K-400	18	63.7	27.9
6	VideoMoCo [111] [CVPR, 2021]	R3D-18	112 × 112	K-400	200	74.1	43.6
7	RSPNet [16] [AAAI, 2021]	R3D-18	112 × 112	K-400	200	74.3	41.8
8	ASCNet [65] [ICCV, 2021]	R3D-18	112 × 112	K-400	200	**80.5**	**52.3**
9	CEP[162] [BMVC, 2021]	R3D-18	224 × 224	K-400	50	75.9	36.6
10	Ours	R3D-18	112 × 112	K-400	40	67.9	32.6
11	VideoPace [148] [ECCV, 2020]	R(2+1)D	112 × 112	K-400	18	77.1	36.6
12	VideoMoCo [111] [CVPR, 2021]	R(2+1)D	112× 112	K-400	200	78.7	**49.2**
13	RSPNet [16] [AAAI, 2021]	R(2+1)D	112 × 112	K-400	200	**81.1**	44.6
14	CEP[162] [BMVC, 2021]	R(2+1)D	224 × 224	K-400	50	76.7	37.6
15	Ours	R(2+1)D	112 × 112	K-400	20	77.6	40.4
16	VCOP [157] [CVPR, 2019]	R(2+1)D	112 × 112	K-100	200	71.4	32.1
17	VCOP [157] + auxSKD	R(2+1)D	112 × 112	K-100	200	72.6	32.5
18	VideoPace [148] [ECCV, 2020]	R(2+1)D	112 × 112	K-100	18	73.8	36.2
19	VideoPace [148] + auxSKD	R(2+1)D	112 × 112	K-100	18	76.1	38.6
20	RSPNet [16] [AAAI, 2021]	R(2+1)D	112 × 112	K-100	200	74.7	37.4
21	RSPNet [16] + auxSKD	R(2+1)D	112 × 112	K-100	200	75.5	39.0
22	Ours	R(2+1)D	112 × 112	K-100	20	**76.3**	**39.6**
23	VCOP [157] [CVPR, 2019]	R3D-18	112 × 112	K-100	200	58.2	25.2
24	VCOP [157] + auxSKD	R3D-18	112 × 112	K-100	200	60.7	28.4
25	VideoPace [148] [ECCV, 2020]	R3D-18	112 × 112	K-100	18	57.5	23.5
26	VideoPace [148] + auxSKD	R3D-18	112 × 112	K-100	18	60.9	27.1
27	RSPNet [16] [AAAI, 2021]	R3D-18	112 × 112	K-100	200	60.2	32.6
28	RSPNet [16] + auxSKD	R3D-18	112 × 112	K-100	200	61.9	**33.4**
29	Ours	R3D-18	112 × 112	K-100	20	**62.9**	33.0

Table 4.1: *Comparative performance results on UCF101 and HMDB51 when pretraining on K400, and most importantly, on the reduced-size dataset K-100 (shaded region) to emphasise the power of our proposed approach. Note auxSKD refers to our proposed auxiliary pretraining stage using similarity-based knowledge distillation.*

the video transformation proposed in [16] and then transfer all the convolutional layer weights to its query encoder and initialise the projection head and key encoder randomly from scratch.

Rows 16-22 relate to the networks with a R(2+1)D backbone. VCOP+auxSKD improves on VCOP by ↑1.2% and ↑0.4% on UCF101 and HMDB51 respectively, while VideoPace+auxSKD similarly surpasses VideoPace alone by ↑2.3% and ↑2.4%. Note these are very close performances to when VideoPace is pretrained on K-400 (cf. Row 11). RSPNet's performance also improves when our auxiliary SKD is deployed, by ↑0.8% for UCF101 and ↑1.7% for HMDB51.

When the R3D-18 backbone is used, consistent improvements are again observed (see Rows 23-29) for all these methods when auxSKD is added to them. Our proposed method obtains the best performance using the R(2+1)D backbone on both datasets at 76.3% and 39.6%. When using R3D-18, it achieves the best result on UCF101 at 62.9% and the 2nd best on HMDB51 at 33.0%. Finally, we note that unlike VideoPace [148], RSPNet [16] and ASCNet [65] (for which no code has been released at the time of writing), we do not have an appearance stream in our method.

4.3.4 Ablation Studies

We perform ablations to establish the effectiveness of our auxiliary pretraining process and our VSPP pretext task.

Effectiveness of auxSKD – We verify the impact of our auxiliary pretraining stage by showing its gains in performance. In Table 4.2, we present the results of our proposed method on both R(2+1)D and R3D-18 backbones, with and without auxiliary pretraining for UCF101, using K-100 and K-400 for pretraining. It is clear that in each and every case auxSKD causes an increase in performance. For example, when using the R3D-18 backbone pretrained on K-100, the relative performance increases for the UCF101 and HMDB51 datasets are approximately 2.1% and 6.7%, respectively. This large margin performance gain on HMDB51 further demonstrates the positive impact of auxSKD in mitigating overfitting on this small-sized dataset.

Temperature Parameters – We studied the effect of changing temperatures of auxSKD for both teacher and student models and report the results in Table 4.3. Here we use VSPP as the pretext task for the primary stage. The best result is achieved when both the teacher's temperature ($\gamma^{\mathcal{T}}$) and the student's temperature ($\gamma^{\mathcal{S}}$) are set to 0.02. This suggests that a balanced temperature setting for both teacher and student models is optimal to achieve the best performance.

Method	Pretrain Dataset	UCF101 Top-1	HMDB51 Top-1
Backbone: R(2+1)D			
Ours - auxSKD	UCF101	76.0	37.4
Ours	UCF101	**77.3**	**38.6**
Ours - auxSKD	K-100	74.0	37.3
Ours	K-100	**76.3**	**39.6**
Backbone: R3D-18			
Ours - auxSKD	K-400	65.8	28.8
Ours	K-400	**67.9**	**32.6**
Ours - auxSKD	K-100	60.8	26.3
Ours	K-100	**62.9**	**33.0**

Table 4.2: *Ablation of the auxiliary pretraining stage auxSKD with our proposed approach (auxSKD + VSPP). The results highlight the impact of auxSKD on the learning efficacy across two distinct datasets and two backbone architectures. Notably, when the auxSKD stage is excluded from the pretraining process, there is a decrease in top-1 accuracy, illustrating its vital role in our method's performance. This is consistent across both UCF101 and HMDB51 datasets, as well as R(2+1)D and R3D-18 backbones.*

Ablation on VSPP – Our VSPP pretext task determines both the segment within a clip where there is a speed alteration compared to the natural speed of the rest of the clip and what the speed rate is, effectively parameters λ and ζ. Based on ablation studies in [148], for all the experiments here we consider 4 different speed rates i.e. $Q = 4$, hence $\lambda = \{1, 2, 3, 4\}$.

Table 4.4 outlines the effect of each sub-task in VSPP when our model pretrains on them separately and jointly (on K-100). Our auxSKD pretraining is not engaged for this ablation. The best result is obtained at 60.8% on the UCF101 dataset when pretraining jointly on both tasks and having the maximum number of segments $Z = 4$.

When the number of segments Z during sampling is fewer (i.e. as ζ ranges from 1 to Z) or a subtask is missed out, the performance drops. Note, the first line of the table when there is only one segment, i.e. $Z = 1$ is the equivalent to VideoPace. We believe that increasing the number of segments in the clip pushes the model to temporally explore the video more to find that specific segment with different speed, resulting in better temporal representation.

$\gamma^{\mathcal{T}}$	0.01	0.02	0.05	0.07	0.1	0.01	0.02
$\gamma^{\mathcal{S}}$	0.01	0.02	0.05	0.07	0.1	0.1	0.1
UCF101	75.0	**76.3**	75.3	74.9	74.9	75.5	75.0

Table 4.3: *Effect of changing the temperatures for our method for UCF101 with R(2+1)D backbone. $\gamma^{\mathcal{T}}$ and $\gamma^{\mathcal{S}}$ indicate teacher and student temperatures respectively.*

Speed Prediction	**Segment Prediction**	**#Classes #speed, #segment**	**UCF101**
✓	-	$[Q = 4\ ,\ Z = 1]$	57.5
✓	✓	$[Q = 4\ ,\ Z = 2]$	57.5
✓	✓	$[Q = 4\ ,\ Z = 3]$	59.5
✓	✓	$[Q = 4\ ,\ Z = 4]$	**60.8**
✓	✗	$[Q = 4\ ,\ Z = 4]$	58.3
✗	✓	$[Q = 4\ ,\ Z = 4]$	59.9

Table 4.4: *Ablation of our VSPP pretext task pretrained on K-100 with R3D-18 (no auxSKD stage). We examine the importance of each subtask within VSPP while the number of segments within the clip changes.*

Pretraining Epochs – In Figure 4.5 (left), we evaluate the performance of our method on UCF101 when pretrained on K-100, with and without auxSKD, using different checkpoints. It can be seen that when auxiliary learning is switched on, the performance of our VSPP pretext task is increased at all checkpoints. We also notice that the performance starts to saturate after 20 epochs for both VSPP and VSPP+auxSKD. In Figure 4.5 (right), we can see that after around 20 epochs, the changes in the VSPP losses are not significant. This demonstrates that our model converges quickly and we can ensure its convergence during pretraining with only 20 epochs.

4.4 Experiments on PD Tasks

In this section, we turn our focus toward a specialised and clinically significant domain by evaluating our SSL framework on the PD4T dataset. Unlike action recognition datasets such as UCF101 and HMBD51, PD4T requires a quality assessment of action. This form of evaluation is more complex and challenging than simply identifying actions, as it pays close attention to subtle changes in movement, which are important for correctly

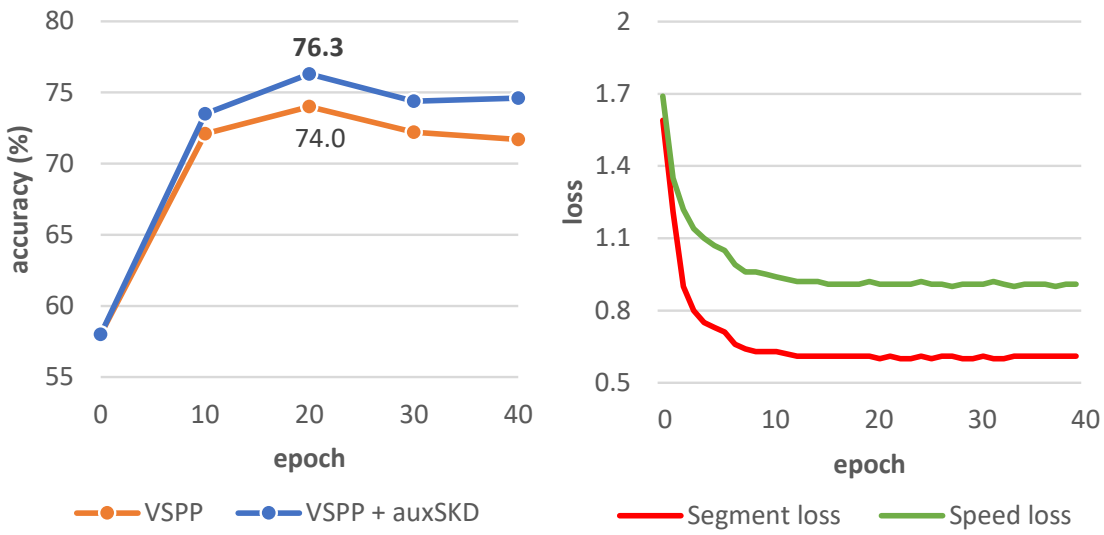


Figure 4.5: *(Left) VSPP pretext task performance with auxSKD (VSPP+auxSKD) and without (VSPP) based on the number of epochs. We pretrained the R(2+1)D model on K-100 for 40 epochs and report the results every 10 epochs on UCF101. (Right) Pre-training losses of our VSSP subtasks on K-100, i.e. speed prediction and segment prediction losses, further illustrate that our model actually converges after around 20 epochs.*

predicting how severe medical conditions such as PD are.

4.4.1 PD4T Dataset

This new fully annotated dataset offers 2931 videos from 30 PD patients tested longitudinally at 8 week intervals. The patients (41 to 72 years old) performed varius PD tasks in clinical settings and their UPDRS [44] quality scores were assigned by trained clinicians ranging from 0 (normal) to 4 (severe). The videos were recorded at 25 fps at a resolution of 1920×1080, using a single RGB camera. PD4T contains four different UPDRS tasks including gait, finger tapping, hand movement, and leg agility. In gait analysis, patients were asked to walk 10 metres at a comfortable pace and then returned to their starting point. In hand movement, patients opened and closed their hand (each hand separately) 10 times, as fully and as quickly as possible. For finger tapping, the patient had to tap the index finger on the thumb 10 times as quickly and as big as possible. In leg agility, the patients placed the foot on the ground and then raised and stomped the foot on the ground 10 times as high and as quick as possible. The number of videos (#video) for each score, as well as the minimum/maximum number of frames (#min/#max) for each task can be seen in Table 4.5. Sample frames are shown in Figure 4.6. It is worth noting that, similar to the PD2T dataset outlined in Chapter 3, the PD4T dataset adheres to the same stringent standards in terms of data protection,

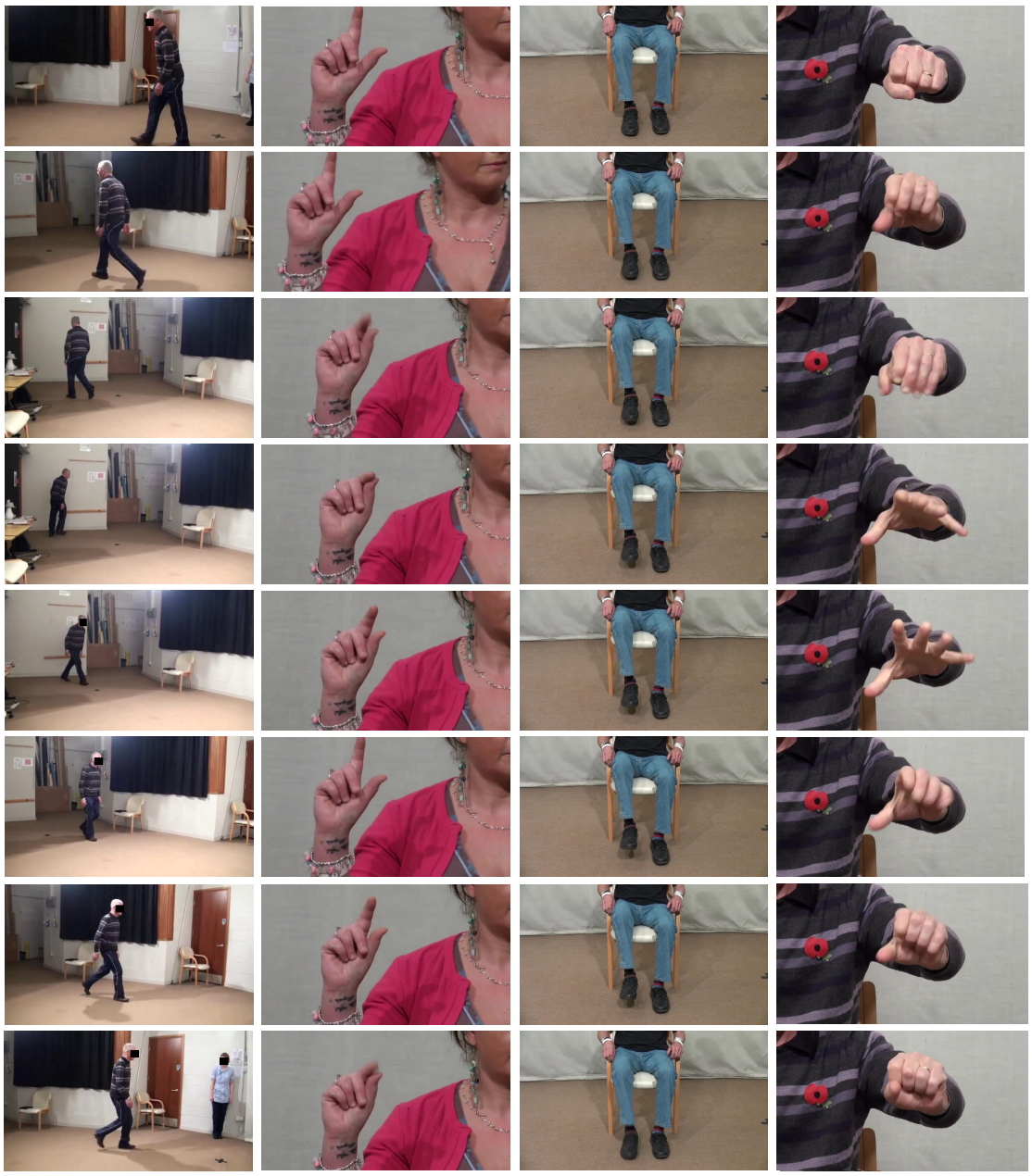


Figure 4.6: *Sample frames from the PD4T dataset: (a) gait, (b) finger tapping, (c) leg agility, and (d) hand movement. All videos are from actual PD patients and were captured at Southmead Hospital, Bristol, UK, as part of a clinical experiment across several months. For hand movement, finger tapping, and leg agility, data were collected from both the left and right sides for each subject.*

ethical approval, and participant privacy.

4.4.2 Details on Fine-tuning

For the fine-tuning stage, we choose CoRe [166], a recent state-of-the-art action quality assessment model, as an example for improvement via our self-supervised pretraining

Score		Normal (0)	Slight (1)	Mild (2)	Moderate (3)	Severe (4)
Gait	#video	196	158	64	8	0
	#min	325	580	421	664	-
	#max	980	1866	13428	10688	-
Finger tapping	#video	167	492	169	24	2
	#min	101	91	117	110	159
	#max	450	724	853	398	460
Hand movement	#video	235	411	179	23	5
	#min	62	59	150	197	220
	#max	334	571	717	1210	648
Leg agility	#video	407	377	54	11	3
	#min	129	126	155	273	345
	#max	513	427	686	504	435

Table 4.5: *The PD4T dataset summary, categorised by severity scores. For each of the four motor tasks — Gait, Finger Tapping, Hand Movement, and Leg Agility — the table lists the total number of videos (**#video**), the minimum (**#min**) and maximum (**#max**) number of frames for the respective task. The severity scores are classified into five categories: Normal (0), Slight (1), Mild (2), Moderate (3), and Severe (4).*

technique. Following this, we fine-tune the model using the PD4T dataset for direct performance evaluation; we employ the I3D model [10] as the backbone to ensure a fair comparison. The baseline model, CoRe, also uses I3D but is pretrained on the K400 dataset using supervised learning. After subjecting our I3D model to a two-stage pretraining process on a PD4T task — initially through auxiliary pretraining using auxSKD and then through primary pretraining using VSPP — we integrate it into the CoRe pipeline to function as its backbone. In CoRe setup, each input video is paired with an exemplar video. This video pair is then fed through the shared pretrained I3D backbone to extract spatio-temporal features, which are then combined with the reference score of the exemplar video. This combined feature set is sent through a GART to obtain the relative quality score between the input and the exemplar. In the inference phase, this process is repeated with multiple exemplars for a more robust final quality score for the input video, achieved by averaging the relative scores. We evaluate CoRe performance with the Spearman Rank Correlation metric ($\mathcal{S}$). A larger value of $\mathcal{S}$ indicates superior performance. The training parameters for CoRe are detailed in Table 4.6. It is worth noting that we use the same pretraining parameters for the pretraining

stage as those used for the action recognition downstream task, as described in Section 4.2.2.

Parameter/Setting	Value/Description
Depth of GART	$d = 5$
Node Feature Dimension	256
Initial Learning Rate (Regression Tree)	1×10^{-3}
Initial Learning Rate (I3D Backbone)	1×10^{-4}
Optimizer	Adam
Weight Decay	0

Table 4.6: *Summary of training parameters and settings for CoRe.*

4.4.3 Results

In Table 4.7, we analyse the efficacy of various pretraining approaches on the PD4T dataset. CoRe serves as the AQA method for fine-tuning the models. The table starts with the baseline approach, which employs an I3D model pretrained on the K400 dataset in a supervised fashion. This baseline shows an average performance of 60.31% across all tasks. Moving on to the second row, where the I3D model is pretrained on K400 using our self-supervised pretext task, VSPP, there is only a marginal decrement (↓ 0.24) in average performance compared to the baseline. The third row brings into light the impact of pretraining the I3D model on the target dataset, PD4T, using VSPP. There is a noticeable improvement in average performance to 62.00% which is 2.8% improvement over the baseline. The improvement suggests that the domain gap between K400 and PD4T has been effectively mitigated by pretraining on target dataset. It is worth highlighting that the performance in the "Leg Agility" task actually improves when using VSPP pretrained on K400 compared to the baseline and also VSPP pretrained on PD4T. This suggests that there is some level of domain overlap for this specific task between the K400 and PD4T datasets, making the more generalised K400 dataset beneficial for this particular task. Lastly, incorporating the auxiliary learning stage (auxSKD) into VSPP and then pretraining on target data (PD4T) leads to a further improvement of 0.76% in the average performance compared to pretraining the model on PD4T using VSPP alone. This validates the importance of the auxiliary learning stage for our PD severity assessment tasks.

Row	Approach	Pretraining dataset	Gait	Finger tapping	Hand movem.	Leg agility	Avg
1	Supervised	K400	78.87	45.93	54.10	62.34	60.31
2	VSPP	K400	77.37	44.48	54.45	**63.98**	60.07
3	VSPP	PD4T	80.15	48.29	58.56	61.00	62.00
4	VSPP + auxSKD	PD4T	**81.19**	**48.66**	**59.38**	61.82	**62.76**

Table 4.7: *Various pretraining approaches on the PD4T dataset. Pretraining on PD task with the VSPP method alone outperforms the same method using the generic K400 dataset. Further incorporation of auxSKD into the PD4T pretraining enhances average accuracy, underscoring the significance of task-specific pretraining in assessing PD tasks.*

4.5 Discussion

4.5.1 Importance of Our SSL Framework for PD

Our experimental results demonstrate the significance of SSL in the assessment of Parkinson's disease severity, where SSL pretraining techniques outperform traditional supervised methods, as shown in Table 4.7. This improvement is attributed to SSL's ability to capture complex patterns and rich representations from data without explicit labels. Unlike supervised learning methods, SSL can uncover subtle and intricate features that might be overlooked in labeled datasets, which is particularly beneficial in the context of PD, where the nature of symptoms is diverse and not always quantifiable with simple labels. In real-world applications, SSL can harness the wealth of unlabelled clinical data to enhance the detection and monitoring of PD, facilitating early intervention and personalised care, which are crucial for improving patient outcomes and quality of life.

4.5.2 Limitations

We identify three main limitations of our work:

(i) a fundamental aspect of our VSPP pretext task is that it thrives on the altered natural pace of motion in a segment of a video while the rest of the clip retains its natural motion. However, any sudden and very fast motion in a clip may violate this assumption as the fast motion within the selected segment of a clip may be missed when it is sampled. This is a similar limitation for other current speed based pretext tasks such as VideoPace and RSPNet.

(ii) other speed-related pretext tasks, such as ASCNet, RSPNet, and VideoPace include an appearance stream in their methodology, however in this work, while the absence of an appearance stream may seem to be a limitation, it was avoided to focus on the power of VSPP as an independent pretext task and promote the auxiliary pretraining stage as two contributions that may be used in a modular fashion by the community. We expect that adding an appearance stream to our model may improve our results.

(iii) the two-stage pretraining approach in our self-supervised learning framework introduces parameter inefficiency by significantly expanding the number of parameters needed. This expansion not only strains computational resources but also escalates storage demands, with each task requiring its unique set of parameters. Such a setup proves challenging in resource-constrained environments, such as edge devices in clinical settings, where minimal computational load and efficient use of storage are imperative.

4.5.3 Future Work

Immediate future work is poised to include experiments with an additional appearance stream and to investigate a more integrated analysis of speed relativity within video clips, with an emphasis on maintaining insensitivity to extreme natural motions, whether slow or fast. There is also an intention to explore methods that would enable the transfer of richer temporal knowledge through the auxiliary pretraining stage to the primary pretraining model, enhancing the depth and robustness of the learned representations. Building upon these enhancements, subsequent studies could also extend the scope of validation to another reduced-size versions of existing large datasets, such as FineAction [94]. By doing so, the research could seek to further substantiate the efficacy of the auxiliary pretraining stage and, importantly, offer valuable insights into diminishing the reliance of self-supervised learning approaches on voluminous pretraining datasets — addressing a salient limitation in the realm of deep learning.

4.6 Conclusions

In this chapter, we introduced an auxiliary-learning phase for self-supervised video representation learning that allows a significant reduction in the amount of unlabelled data required for the pretraining task. The approach exploits similarity-based knowledge distillation to better prepare a (student) network to perform its primary pretraining task. Our experiments show that this new auxiliary phase auxSKD improves the performance of other existing SSL approaches, such as VCOP [157], VideoPace[148], and RSPNet [16]. We also introduced a new video speed analysis task, VSPP, that predicts the index

and altered speed of a segment within a clip which is sampled at a different frame rate to the rest of the clip. Solving this task can strength the network's awareness of the video's natural speed rate and alleviate the imprecise video speed labeling problem [16]. Our experiments illustrate that the features learnt achieve competitive or superior results compared to the state of the art, while training on a much smaller dataset, e.g. K-100 rather than K-400, and at a lower computational cost.

We also expanded our experiments to include PD tasks, specifically by presenting results on a new dataset of functional mobility actions performed by actual Parkinson's patients, named PD4T, for performance quality assessment with potential for longitudinal evaluation. We demonstrated that our SSL framework outperforms supervised method on four different PD tasks of PD4T dataset. This breakthrough highlights the unique advantages of SSL methods for initialising AQA models with domain-specific knowledge and paves the way for future research.

Chapter 5

PECoP: Parameter Efficient Continual Pretraining

5.1 Introduction

Despite recent advances in action quality assessment [22, 48, 79, 115, 138, 156, 159, 166], these methods are affected by insufficient quantities of annotated data for training deep networks [113]. This becomes even more challenging when extra effort is needed to produce very precise labels, e.g. for health-related applications, such as PD severity assessment [22, 33, 97, 102]. A common solution to address such problems is to start with a model that is originally pretrained on a large domain-general dataset, commonly Kinetics-400 [71] (K400), and finetune it on one's target AQA dataset [138, 159, 166] (as we explored in Chapter 3, see Fig 5.1(a)). However, in Chapter 4 we showed that this could be less effective due to the significant domain gap between the general and target AQA datasets.

A promising route to address the shortcomings of this direct jump from classical pretraining to finetuning can be to further pretrain using domain-specific unlabeled data, i.e., Continual Pretraining – a strategy that has had a remarkable impact in NLP [49, 50, 154] and recently in image/object classification [7, 122]. When it comes to video-domain tasks (e.g. as in AQA), this additional pretraining stage on in-domain data may be computationally prohibitive or impractical, due to the requirement for updating all parameters and storing pretrained parameter sets for each separate task.

Another possible approach could be BatchNorm tuning [41, 122] to equip the pretrained model with domain-specific information by only updating the affine parameters of Batch-

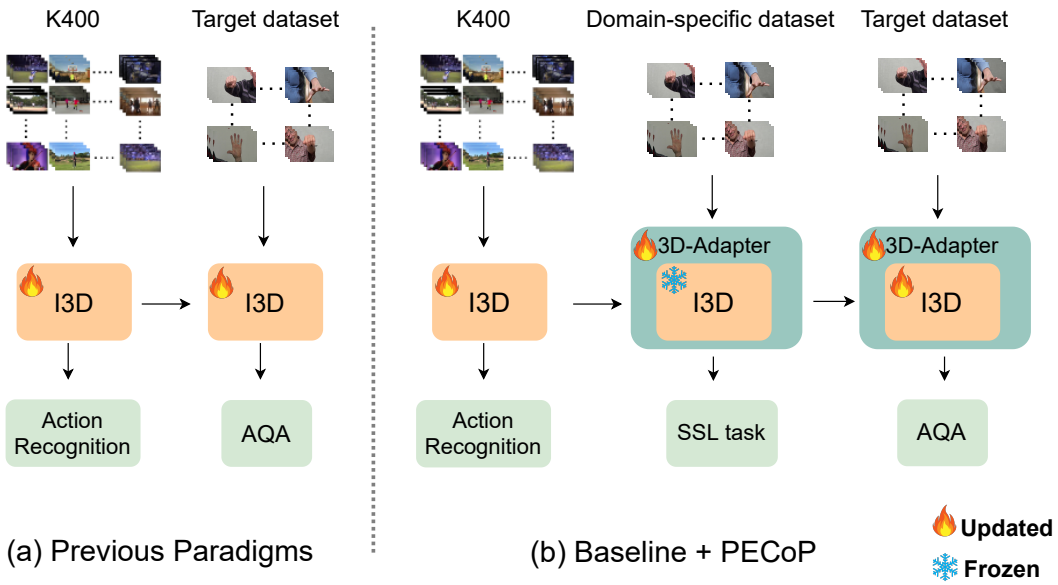


Figure 5.1: *(a) Previous works directly transfer the model pretrained on domain-general data to AQA downstream tasks with target data fine-tuning, (b) in our proposed PECoP framework, the pretrained model continues to learn towards a specific AQA task through an additional pretraining stage, where only a small set of 3D-Adapter parameters are updated on unlabeled domain-specific data in a SSL approach, while the baseline model's weights remain frozen.*

Norm layers, while other pretrained parameters are frozen. Although this technique can greatly reduce the number of trainable parameters, we show that in a continual learning framework it can fail on those AQA tasks that are more domain-specific, e.g. in PD tasks and JIGSAWS [42].

In this chapter, we propose adding a Parameter-Efficient Continual Pretraining (PECoP) adaptation stage to the traditional AQA transfer learning workflow that can efficiently *adapt* the domain-general pretrained model for the downstream AQA task. Inspired by adapter-based methods which have recently achieved strong results with transformer architectures on NLP benchmarks [53, 56, 62, 118], we present 3D-Adapter, a lightweight convolutional bottleneck block which is inserted into a pretrained 3D CNN (e.g. I3D inception modules [10]) and learns domain-specific spatiotemporal knowledge via a self-supervised learning (SSL) approach. During domain-specific pretraining, only the adapter parameters are updated while the original weights of the pretrained model are frozen to allow a high degree of parameter-sharing (see Fig. 5.1(b)). This greatly reduces the computational and storage costs of conventional continual pretraining, and also prevents overfitting by alleviating catastrophic forgetting [56]. The work in this chapter was published in [24].

In Section 5.2, we formalise each of the PECoP components. Section 5.3 presents compar-

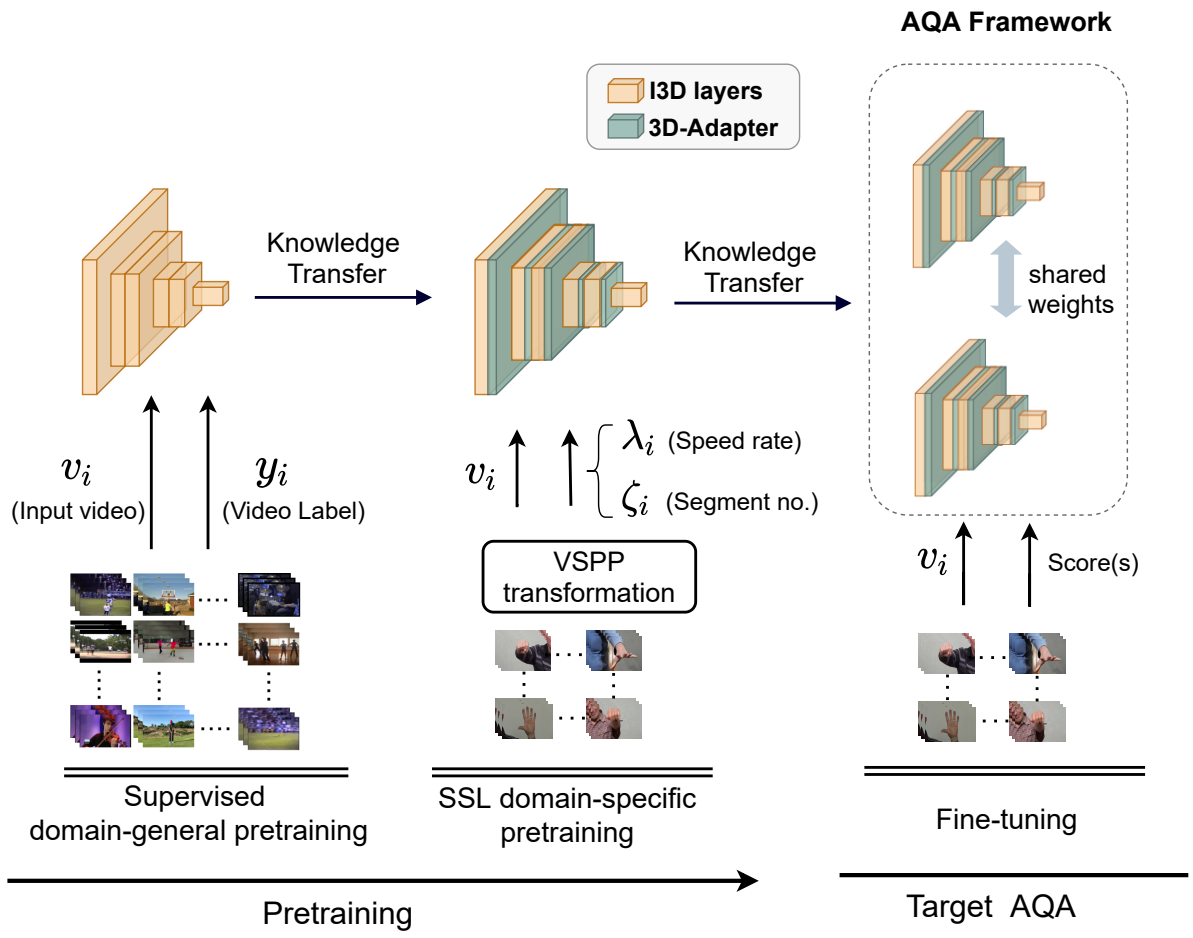


Figure 5.2: *An overview of PECoP – First, a 3D encoder is pretrained on a domain-general dataset (i.e. K400). Then, we equip the pretrained model with 3D-Adapters and update their parameters using VSPP [23], a SSL pretext task, on unlabeled domain-specific data. Finally, we fine-tune the pretrained model on the AQA target task.*

ative experiments using our PD4T dataset and three public AQA benchmark datasets, namely, MTL-AQA [114], JIGSAWS [42], and FineDiving [159]. Section 5.5 delves into the learning efficiency of PECoP, its importance in the AQA context, as well as its limitations and avenues for future work. Conclusions are drawn in Section 5.6.

5.2 Proposed Approach

Next, we outline our proposed continual pretraining approach implemented via self-supervised training of 3D-Adapter modules. The pipeline of our framework is shown in Fig. 5.2.

Let D_g be a large-scale, annotated, domain-general video dataset used for a learning task

T_g, and D_t be a target video dataset in the AQA domain for a learning task T_t, with a significant domain discrepancy between T_g and T_t. Then, given an unlabelled video dataset D_q, where $D_q \subseteq D_t$, our aim is to leverage the representations in D_g and D_q to learn a transferable spatiotemporal feature extractor that is able to perform as well as possible on D_t for task T_t.

5.2.1 Domain-general Pretraining

The first stage of pretraining focuses on learning robust and general spatial-temporal representations. This is achieved by training the backbone encoder from scratch on a large and domain-general dataset, such as Kinetics-400 (see 1st column of Fig. 5.2). Kinetics-400 is a significant dataset for this process, featuring a diverse collection of videos that offer a wide spectrum of spatial-temporal patterns for our encoder to learn from. This diversity forms the basis for rich and adaptable representations that can be effectively leveraged for more specific tasks in the later stage. Given the availability of pretrained weights, we merely initialise the backbone using the existing supervised pretrained weights from Kinetics-400.

5.2.2 In-domain SSL Continual Pretraining

We then equip our K400 pretrained model with randomly initialised 3D-Adapter modules. Our proposed 3D-Adapter has a similar bottleneck architecture as used in Transformers [62, 64] and recently in 2D CNNs [13]. However, what differentiates it from these previous iterations is the necessity for 3D layers, allowing the Adapter to be applied effectively to 3D CNNs and, subsequently, trained on video data. The architecture of our 3D-Adapter and its integrated design with the inception module of the I3D model is shown in Figure 5.6. A performance boost can be obtained if a single 3D-Adapter is inserted after the concatenation layer of each inception module.

A 3D-Adapter consists of a downsampling, depth-wise, 3D convolution with learnable weights $\theta_{down} \in \mathbb{R}^{\frac{C_{in}}{\lambda} \times \lambda \times K \times K \times K}$, a non-linear function $f(.)$, e.g. ReLU, followed by an upsampling, point-wise, 3D convolution with learnable weights $\theta_{up} \in \mathbb{R}^{C_{out} \times \frac{C_{in}}{\lambda} \times 1 \times 1 \times 1}$. Here, C_{in} and C_{out} are the channel dimensions of the input and output feature maps, respectively, $K = 3$, and the compression factor λ denotes the bottleneck's dimension. Hence, given an input feature vector $h_{in} \in \mathbb{R}^{C_{in} \times D \times H \times W}$, then the output feature vector $h_{out} \in \mathbb{R}^{C_{out} \times D \times H \times W}$ of our 3D-Adapter is

$$h_{out} = \alpha.(\theta_{up} \otimes f(\theta_{down} \bar{\otimes} h_{in})) + h_{in} \ , \tag{5.1}$$

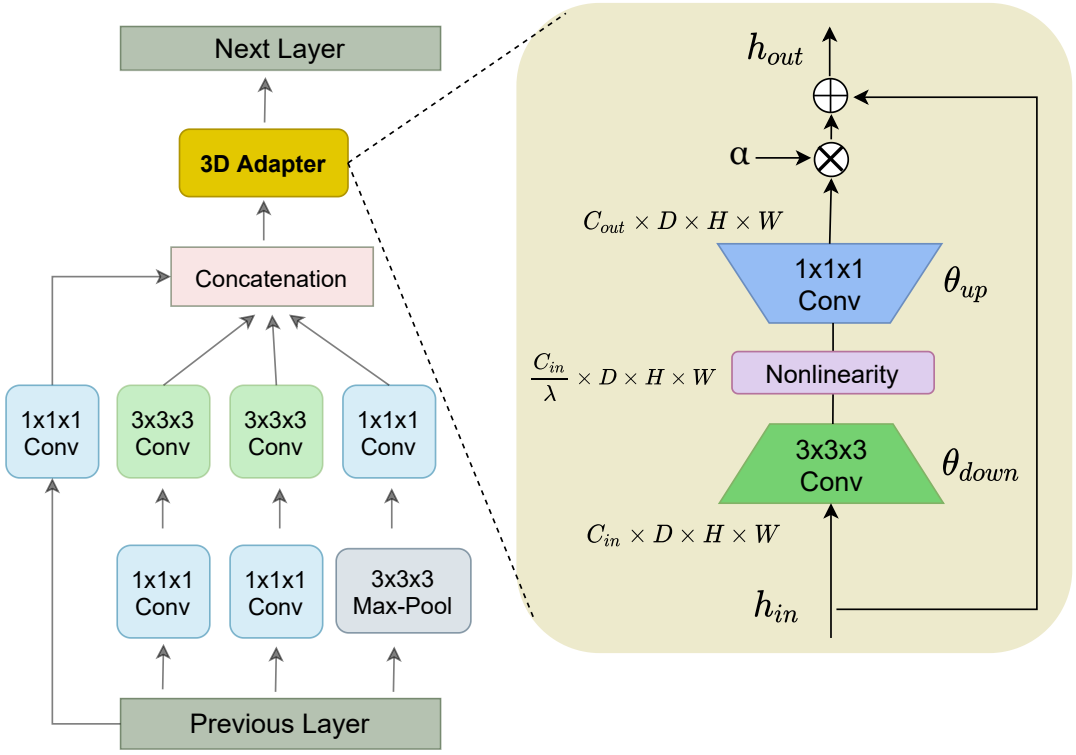


Figure 5.3: *Inception module with adapter used in I3D model. During the pretraining phase, only the adapter module parameters are optimised, while the parameters of other layers within the Inception module remain frozen. The bottleneck structure of the adapter is employed for its efficiency in dimensionality reduction and computational manageability. It compresses high-dimensional input data into a lower-dimensional space, allowing for focused processing with reduced computational overhead.*

where $\otimes$ and $\bar{\otimes}$ are point-wise and depth-wise 3D convolution respectively, and α is a tunable scalar hyperparameter in $\mathbb{R}^{C_{out}}$ which is initialised as ones, following [13, 64].

During the proposed continual pretraining stage, we only allow the 3D-Adapter parameters to be optimised on D_q, while the original model layers' weights stay frozen (middle column in Fig. 5.2). We carry out the training process through the lens of self-supervised learning, a mechanism by which labels are automatically generated from the unlabeled videos present within the D_q dataset.

Since understanding the quality of action is heavily dependent on movement patterns, we focus VSPP, proposed in Chapter 4, as an SSL pretext task for this stage. We shall explore the performance of other recent SSL methods (e.g. VideoPace [148] and RSPNet [16]) during our ablations.

5.2.3 Supervised Fine-tuning

We select state-of-the-art action quality assessment models CoRe [166], USDL/MUSDL [138], TSA [159], and TPT [8] as example models that can be enhanced with PECoP and then fine-tuned on AQA datasets for direct evaluation. In essence, each layer of the models, encompassing both the original layers and the newly introduced adapter layers, are subjected to fine-tuning. This fine-tuning is carried out on the target dataset, D_t (see rightmost column in Fig. 5.2).

USDL [138] – The features obtained from segments of a video clip pass through our continually pretrained I3D backbone and fused through temporal pooling, and then sent through softmax to generate the predicted quality assessment distribution. The KL loss between the predicted distribution and a Gaussian distribution generated from the ground-truth score is applied for optimisation.

MUSDL is a multi-path version of USDL which predicts the final score if multiple-judge scores are available, as is the case in the MTL-AQA and JIGSAWS datasets.

TSA [159] – Upon receiving a pair of query and exemplar instances, we use our continual pretrained I3D to extract spatial-temporal features (similarly to CoRe). Then a Temporal Segmentation Attention (TSA) module is used to evaluate action quality through a series of steps, which include: successively accomplishing procedure segmentation, carrying out procedure-aware cross-attention learning, and executing fine-grained contrastive regression. The supervision of the TSA is based on step transition labels and action score labels. These labels aid the model to focus on exemplar regions that align with the query step and quantify their differences to predict accurate action scores.

TPT [8] – We first split the input video into 5 overlapping clips, and feed each clip into our continually pretrained I3D backbone to get clip level feature representations. Then, TPT is used to convert these representations into temporal part-level representations. Finally, a part-aware contrastive regressor (following [166]) computes part-wise relative representations and fuses them to perform the final relative score regression.

The training and inference pipeline of **CoRe** is explained in Section 4.4.

5.3 Experiments

5.3.1 Dataset

We evaluate our self-supervised continual pretraining adaptation module on our PD4T dataset (Section 4.4.1) as well as three benchmark AQA datasets including MTL-AQA

Figure 5.4: *Sample frames of the three surgical tasks in the JIGSAWS dataset, from left to right: suturing, knot-tying, and needle-passing.*

[114], JIGSAWS [42], and FineDiving [159]. The details of benchmark datasets are detailed below:

(i) **MTL-AQA** [114] contains 1412 video clips collected from 16 different world events and includes a variety of diving actions, covering both individual and synchronous divers, with videos from different angles. The dataset is equipped with various annotations to support research in areas such as AQA, action recognition, and commentary generation. The annotations comprise scores from 7 judges, final scores, difficulty degree and type of diver's action. We followed the evaluation settings suggested in [114] to divide the dataset into a training set of 1,059 videos and a test set consisting of 353 videos.

(ii) **JIGSAWS** [42] is a collection of 103 surgical activity videos, distributed among three different tasks: 39 videos for Suturing (S), 28 videos for Needle-Passing (NP), and 36 videos for Knot-Tying (KT). Further details on each task are elaborated as follows:

- **S** – The individual takes the needle and moves toward the incision (indicated as a vertical line on the tabletop model), threading the needle through the simulated "tissue" from one marked point on one side to a corresponding point on the opposite side of the incision. Following the initial needle pass, the subject removes the needle from the tissue, transfers it to the right hand, and carries out three more similar needle passes.
- **NP** – The individual takes the needle and threads it through four small metal loops from right to left. These loops are elevated slightly above the surface of the tabletop model.
- **KT** – The individual picks up one end of a suture that is connected to a flexible tube, which is in turn anchored at both ends to the tabletop model, and proceeds to tie a single loop knot.

Figure 5.4 shows snapshots of the these tasks. Each task is annotated by multiple sub-

scores (that represent e.g., flow of operation, quality of final outcome, and so on), with the final score defined as the sum of these sub-scores. Following [138, 166], we adopt 4-fold cross validation for evaluation for this dataset.

(iii) **FineDiving** [159] is a fine-grained sports video dataset designed for AQA tasks. It offers 3,000 videos from various diving events and competitions, including the Olympic Games, World Cup, and World Championships. FineDiving has several characteristics: (1) Two-level semantic structure. Each video is categorised using two layers of semantic tags: one for the main action type and another for the sub-action type. The main action type is derived from a sequence of these sub-actions. (2) Two-level temporal structure. Actions within each video are time-stamped to mark their start and end, and every action is further segmented into a series of steps based on a clearly established vocabulary. (3) Certified scoring metrics, including evaluative scores from judges and the level of difficulty, are obtained from FINA. The dataset comes with 52 actions, 29 sub-actions, and 23 types of difficulty degrees. Following the training and evaluation settings outlined in [159], we select 75 percent of the samples of this dataset for training and the remaining 25 percent for testing.

5.3.2 Experiment Setup

The experiments were performed on an Nvidia RTX 3090TI GPU under Cuda 11.6 with cuDNN 8.2. We first initialise our I3D model with K400 pretrained weights which then remain frozen throughout the pretraining stage. After adding 3D-Adapters, the classification head is replaced with two randomly initialised FC layers f_λ and f_ζ corresponding to the segment speed and index outcomes of the VSPP [23] pretext task. In this stage, we generate 32-frame long video clips, and empirically set the two parameters needed for VSSP to $[\lambda = 4, \zeta = 4]$ or $[\lambda = 4, \zeta = 3]$ which is either at, or close to, those recommended in [23].

We perform SSL pretraining on domain-specific datasets by only updating the 3D-Adapter layers over 8 epochs, with batch size of 16 and SGD with a 1×10^{-3} learning rate. Note that the training set videos of the target data is our domain-specific dataset for SSL pretraining.

For data augmentation, we randomly crop the video clips to 224×224 followed by horizontal flip and color jittering of each frame. Following [23], we apply 10x more iterations per epoch for temporal jittering. In all experiments, the input clip length is 32 during pretraining.

5.3.3 Fine-tuning

The pretrained I3D model is then the backbone network of our baselines USDL, MUSDL, CoRe, TSA, and TPT. and we evaluate their performance with the Spearman Rank Correlation metric ($\mathcal{S}$), expressed as percentages.

For the JIGSAWS dataset, in keeping with other methods [138, 166], we provide values after four-fold cross-validation.

Please note that, at this stage, we adopt similar hyperparameter settings and training/evaluation strategy for each baseline as reported in [138], [166], [159], and [8]. respectively.

5.3.4 Comparative Evaluation

We present results on the MTL-AQA [114] and JIGSAWS [42] datasets against state-of-the-art AQA methods MUSDL [138] and CoRe [166] when we enhance them with PECoP, as well as when we enhance them with another recent continual pretraining workflow, HPT [122] (see Table 5.1). In HPT, which has only been applied to image domain tasks till now, simply additional pretraining steps are introduced on domain-specific datasets with all model parameters updated at every stage. We use the same hyperparameters for training both PECoP and HPT.

While with PECoP improved results are obtained across the board, the improvements on JIGSAWS are very significant, e.g. after adding PECoP, MUSDL's average performance on the three tasks in the JIGSAWS dataset improves to 76% (↑ 6%). Similarly, CoRe's average performance on the same tasks increases to 89% (↑ 4%). This clearly shows PECoP's effectiveness in narrowing the substantial domain gap between the JIGSAWS dataset and K400.

Further, we note that adding HPT to the baselines results in a performance drop on JIGSAWS. Specifically, CoRe's performance decreases to 80% (↓ 5%). This decline can be attributed to overfitting, as HPT requires all model parameters to be pretrained on a relatively small dataset (i.e. ∼13M parameters vs. PECoP's 3D-Adapters' ∼1M).

In Table 5.2, we show that not only PECoP dramatically reduces the model capacity, it also requires drastically fewer epochs to converge compared with HPT.

Table 5.3 presents the results on the FineDiving dataset introduced in [159], comparing CoRe [166] and TSA [159], with and without PECoP. Since TSA requires step transition labels for training, we cannot evaluate it on other datasets. As shown, TSA+PECoP

Table 5.1: *Spearman Rank Correlation results on MTL-AQA and JIGSAWS, with and without continual pretraining methods including our proposed PECoP and HPT [121]. PECoP's enhancement of MUSDL and CoRe models demonstrates superior performance, achieving the highest scores and reflecting its efficacy over HPT in these evaluations. Please note that ⋆ViSA [84] and MultiPath-VTPE [89] are customised towards surgical skill assessment and not general AQA tasks.*

Method	Year	MTL-AQA	JIGSAWS			
		Diving	S	NP	KT	Avg $\mathcal{S}$
C3D-SVR [115]	2017	77.16	-	-	-	-
C3D-LSTM [115]	2017	84.89	-	-	-	-
MSCADC-STL [114]	2019	84.72	-	-	-	-
MSCADC-MTL [114]	2019	86.12	-	-	-	-
C3D-AVG-STL [114]	2019	89.60	-	-	-	-
C3D-AVG-MTL [114]	2019	90.44	-	-	-	-
JRG [109]	2019	-	36	54	75	57
USDL [138]	2020	90.66	64	63	61	63
MultiPath-VTPE [89]⋆	2021	-	82	76	83	80
TSA-Net [150]	2021	94.22	-	-	-	-
I3D + MLP [166]	2021	89.21	61	68	66	65
I3D-TA [172]	2022	92.79	-	-	-	-
ViSA[84]⋆	2022	-	84	86	79	83
PCLN [81]	2022	92.30	-	-	-	-
ResNet34-(2+1)D-WD [37]	2022	93.15	-	-	-	-
MUSDL [138]	2020	92.73	71	69	71	70
MUSDL + HPT [122]	2023	93.49	69	75	72	72
MUSDL + PECoP	2023	**93.72**	**77**	**76**	**76**	**76**
CoRe [166]	2021	95.12	84	86	86	85
CoRe + HPT [122]	2023	94.26	80	81	80	80
CoRe + PECoP	2023	**95.20**	**88**	**90**	**88**	**89**

Table 5.2: *Comparison of PECoP and HPT [122] in terms of storage size and pretraining cost. For PECoP, the count of trainable parameters is related to the adapter modules inserted into the I3D model backbone, which contributes to a smaller overall footprint and indicates more efficient utilisation of resources. Note that the timing is per minibatch.*

Continual Pretraining	#trainble parameters	#epochs	Size	Time/minibatch
HPT [122]	∼13M	16	∼54MB	101ms
PECoP	∼1M	8	∼4MB	71ms

improves on TSA by 1.10%. Although TSA outperforms CoRe alone, CoRe+PECoP surpasses TSA and TSA+PECoP to achieve the state-of-the-art performance on FineDiving dataset.

Table 5.4 presents the results for the PD4T dataset. Given only a single action performance score based on the UPDRS scale [44] is available per clip, we compare PECoP and HPT for USDL instead of MUSDL. We observe the Spearman's rank correlation improves when averaged across the four PD4T tasks for both HPT and PECoP when added to both USDL and CoRe (↑ 2.03% and ↑ 3.56% respectively for PECoP), although HPT performs marginally better (↑ 2.22%) when added to USDL. We assume this slight advantage for HPT is likely due to its greater model capacity for handling the complex patterns in the PD4T dataset; however, PECoP achieves nearly equivalent performance gains while significantly reducing continual pretraining and storage costs.

Table 5.4 also shows that the performance on the finger tapping task is significantly lower compared to other tasks. This is likely due to the inherent complexity of accurately capturing and assessing the subtle and rapid movements involved in finger tapping, which poses a greater challenge for the model's analysis capabilities.

5.3.5 Temporal Parsing Transformer

The Temporal Parsing Transformer [8] (TPT) is a recent state-of-the-art AQA method based on transformers [8]. Unlike existing AQA methods that focus on holistic video representations for score regression, TPT decomposes the video into temporal segments (part-level representations) to extract features. Such a decomposition is critical to TPT's learning process to capture the possible phases of a typical AQA action, e.g. a diving

action which contains several key parts, such as approach, take off, flight, etc. We evaluate the performance of TPT[1] on our PD4T dataset with and without PECoP.

As shown in Table 5.5, PECoP significantly boosts the performance of TPT across the various actions in PD4T. We note that, the performance of TPT is significantly lower than CoRe and USDL on PD4T tasks (See Table 5.4). We believe this may be attributed to the substantial degree of action repetition (e.g. in finger tapping or leg agility). In such cases, TPT's part-level representations, as opposed to a more holistic representation, cannot provide enough discriminative information for its learning process and hence TPT's part-level representations do not necessarily align well with some AQA tasks, such as those in PD4T.

Table 5.3: *Spearman Rank Correlation results on FineDiving dataset with CoRe and TSA as the baselines. While TSA alone achieves strong results, the combination of CoRe and PECoP reaches state-of-the-art performance.*

Method	$\mathcal{S}$
CoRe [166]	90.61
CoRe + PECoP	**93.15**
TSA [159]	92.03
TSA + PECoP	93.13

5.4 Ablations

In this section, we perform ablations on our PECoP framework for AQA tasks, focusing on the influence of different SSL methods employed for domain-specific pretraining, the role of BatchNorm tuning in domain adaptation, and the impact of integrating 3D-Adapters into another 3D CNN, e.g. R3D-18 [54].

5.4.1 Different SSL Methods

We investigate the performance of PECoP when using different SSL methods for domain-specific pretraining, i.e. RSPNet [16], a contrastive learning-based SSL approach (based on MoCo [55]), and VideoPace [148], a transformation-based SSL pretex task (similar to VSPP). In this ablation, CoRe has been used as the AQA baseline and JIGSAWS

[1]To train and evaluate TPT we used the code provided in https://github.com/baiyang4/aqa_tpt.

Table 5.4: *Spearman Rank Correlation results on the PD4T dataset for baseline methods USDL and CoRe, with further enhancements from continual pretraining methods PECoP and HPT. Despite HPT's greater model capacity benefitting USDL, PECoP's integration demonstrates nearly equivalent performance improvements with the added advantage of reduced pretraining and storage costs.*

Method	Gait	Finger tapping	Hand movem.	Leg agility	Avg. $\mathcal{S}$
USDL [138]	79.14	42.58	53.93	56.47	58.03
USDL + HPT [122]	**81.93**	46.38	54.15	**58.54**	**60.25**
USDL + PECoP	80.68	**47.44**	**56.19**	58.09	60.06
CoRe [166]	78.87	45.93	54.10	62.34	60.31
CoRe + HPT [122]	81.42	**49.73**	57.06	63.98	63.05
CoRe + PECoP	**82.33**	49.40	**59.46**	**64.27**	**63.87**

Table 5.5: *Spearman Rank Correlation results on the PD4T dataset with TPT as the baseline. Due to the lengthy training duration of TPT (approximately 5 days with an Nvidia RTX 3090TI GPU for each task) evaluations were limited to PECoP without comparison to other continual pretraining methods such as HPT.*

Method	Gait	Finger tapping	Hand movem.	Leg agility	Avg. $\mathcal{S}$
TPT [8]	77.80	36.05	47.80	46.27	51.98
TPT + PECoP	**79.90**	**40.73**	**51.07**	**50.38**	**55.52**

as the target AQA task. As shown in Table 5.6, VSSP achieves the best result for domain-specific pretraining.

Table 5.6: *Determining which SSL pretext task would be better to use - comparing contrastive learning approach (RSPNet [16]) to transformation-based ones (VideoPace [148] and VSPP [23]). The experiment was performed on the JIGSAWS dataset as an example.*

Method	S	NP	KT	Avg. $\mathcal{S}$
RSPNet [16]	83	86	84	84
VideoPace[148]	86	87	87	87
VSPP [23]	**88**	**90**	**88**	**89**

5.4.2 BatchNorm (BN) Tuning

As mentioned earlier, BN tuning can be used to equip a pretrained model with domain-specific knowledge by only updating the affine parameters of BatchNorm layers. Figure 5.5 illustrates the comparative performance of BatchNorm tuning (HPT+BN) and other pretraining strategies, such as domain-general pretraining (Dom-G), domain-specific SSL pretraining (Dom-S) from scratch, and PECoP. Again, CoRe is used as the AQA baseline.

Each plot corresponds to an AQA target task taken from our various datasets. On all these tasks, PECoP outperforms HPT+BN, particularly by a large margin on the tasks within PD4T and JIGSAWS datasets. Further, we observe that, HPT+BN performs significantly worse than Dom-G alone on the all three tasks within JIGSAWS dataset. This suggests that the BN affine parameters, β and γ, generally have a negative impact on downstream AQA tasks when facing a significant domain shift. This happens because β and γ are tuned to the feature distribution of the source domain. When these parameters are applied to a significantly different target domain, they fail to correctly adjust feature statistics, resulting in a misalignment between the source and target domains. This issue becomes worse in smaller target datasets, as the limited number of examples makes it harder for the model to learn the true feature distribution, increasing the risk of overfitting.

In addition, the figure also shows that Dom-S performs variably. It fared worse than Dom-G on MTL-AQA, but exceeds Dom-G in nearly all PD4T tasks. This discrepancy suggests that direct transfer from the K400 dataset may not be ideal for PD4T tasks due to a significant domain gap. On the other hand, Dom-S also performs poorly on all JIGSAWS tasks, implying that self-supervised pretraining from scratch on smaller datasets is suboptimal. These observations highlight the need for adaptable pretraining strategies, an aim achieved by our PECoP approach, which consistently delivers superior performance in each of the evaluated tasks.

5.4.3 3D-Adapters for ResNet

We evaluate the effectiveness of our 3D-Adapter with another 3D CNN, i.e. R3D-18 [51] which is a common backbone network for action recognition tasks. We first insert a 3D-Adapter into each 3D residual blocks of R3D-18 backbone (see Fig. 5.6) and train this model through our continual pretraining framework. This model is then used as the backbone network for CoRe to fine-tune on the target AQA task. We conduct this experiment on our PD4T dataset and the results are reported in Table 5.7. As shown,

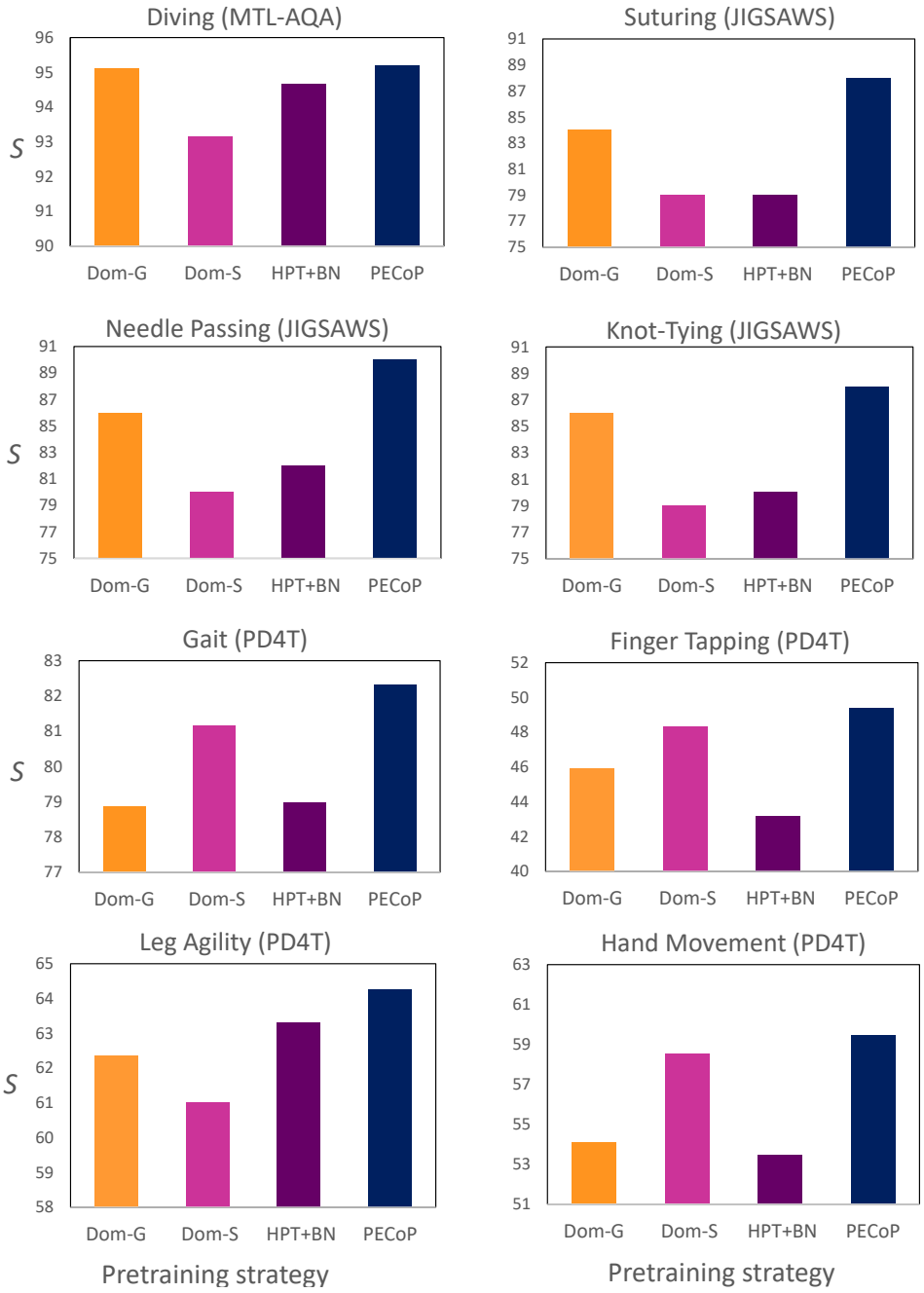


Figure 5.5: *Comparison of PECoP with Domain-Specific SSL Pretraining (Dom-S), Domain-General Pretraining (Dom-G), and BatchNorm Tuning (HPT+BN) across eight different AQA tasks from MTL-AQA, PD4T, and JIGSAWS datasets. Each plot represents a unique AQA task and shows the performance of the four approaches. Dom-G employs pretraining on a domain-general dataset like K400, while Dom-S focuses on domain-specific self-supervised pretraining on target data. HPT+BN fine-tunes only the BatchNorm layers of a pretrained model. PECoP consistently outperforms the other approaches across all tasks, indicating its robustness and adaptability for AQA tasks with different domains and complexities.*

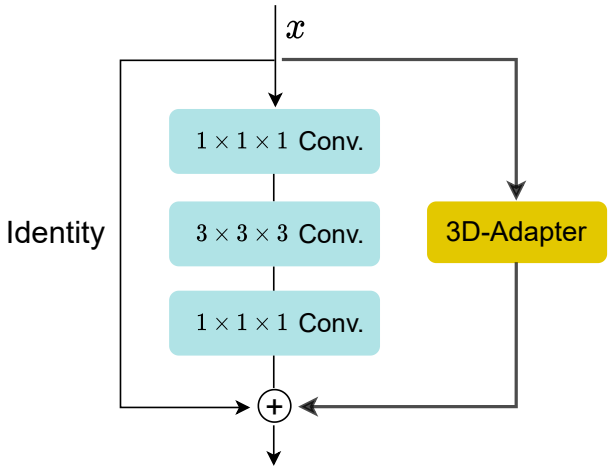


Figure 5.6: *3D residual block equipped with 3D-Adapter used in the R3D-18 model. We empirically find that such a configuration leads to a better performance.*

across all PD4T tasks, CoRe+PECoP outperforms CoRe alone (i.e. CoRe with R3D-18 backbone in both cases).

Table 5.7: *Spearman Rank Correlation results on the PD4T dataset with R3D-18 backbone used in CoRe. The results clearly demonstrate that the inclusion of PECoP enhances the assessment accuracy across all evaluated Parkinson's disease-related tasks.*

Method	Gait	Finger tapping	Hand movem.	Leg agility	Avg. $\mathcal{S}$
CoRe	76.16	35.35	50.53	49.96	53.00
CoRe + PECoP	**79.11**	**39.71**	**55.37**	**52.56**	**56.69**

5.5 Discussion

5.5.1 Learning Efficiency of PECoP

PECoP allows the model to leverage knowledge gained from a previous stage and requires only a small subset of parameters to be specifically learnt for the new task. This approach leads to fewer epochs and less training data required for convergence on a new task, resulting in a reduction of both computational resources and pretraining time (refer to Table 5.2). With fewer trainable parameters, PECoP has a limited capacity to memorise training data, and as a result, it is forced to learn more generalised patterns, making it less prone to overfitting.

Another benefit to PECoP's efficiency is its ability to avoid the issue of forgetting [56]. Unlike traditional continual pretraining methods [7, 122] that require updating all model parameters, thereby risking forgetting previously learned patterns, PECoP leverages pre-existing domain-general knowledge without modifying this foundational knowledge. Instead, it simply augments it with domain-specific knowledge, thus effectively alleviating the issue of forgetting.

5.5.2 Importance of PECoP for AQA

PECoP is crucial for AQA tasks, offering unique benefits that are valuable in diverse applications, ranging from healthcare settings like PD severity assessment to sports evaluations such as diving. Its scalable design eliminates the need for multiple pretrained models for different assessments, a key advantage where quick and precise evaluations across a range of tasks are essential. Furthermore, PECoP's data efficiency makes it well-suited for healthcare settings where annotated data is often scarce. Its computational efficiency also facilitates faster decisions and saves resources, which is vital in environments with limited computational capabilities.

5.5.3 Limitations

One of the key weaknesses in a parameter-efficient continual pretraining approach like PECoP is the potential trade-off between efficiency and complexity in the model. Although using a limited set of adaptable parameters minimises the risk of overfitting, it may unintentionally lead to underfitting, making the model less effective at capturing intricate spatial and temporal features. This limitation becomes especially noticeable in AQA tasks, where accurately capturing subtle details may require a model with greater capacity.

5.5.4 Future Work

PECoP has primarily been evaluated using 3D CNN backbones for AQA tasks. However, its potential effectiveness with other architectures, such as transformers, presents an exciting avenue for future research. Given the backbone-agnostic nature of PECoP, it can be adapted to multiple models, including transformers, by integrating appropriate adapters [17], utilising a similar parameter-efficient continual pretraining approach. Future studies could explore this strategy further, optimising the pretraining process across different architectures and assessing its applicability to other vision tasks, including action recognition and few-shot learning.

5.6 Conclusions

We proposed PECoP, a parameter efficient continual pretraining workflow to better transfer the knowledge learned from existing large-scale video datasets (e.g. K400) to AQA target tasks by only updating a small number of additional bottleneck layers (called 3D-Adapters) through self-supervised learning. Alongside the evaluation on benchmark datasets, we also presented results on PD4T dataset, including four different Parkinson's disease tasks: gait, finger tapping, leg agility, and hand movement. Experiments on four AQA datasets (8 different tasks) with four AQA baselines (CoRe, USDL/MUSDL, TSA, and TPT) demonstrated the significant advantages of PECoP over the conventional continual pretraining approach with respect to both generalisation ability, storage needs, and training cost.

Chapter 6

Conclusions

In this final chapter, the key advances achieved in this thesis are summarised, while also critically examining their limitations and proposing potential areas for future study. The chapter is organised in the following manner: It begins with a summary to revisit the primary objectives and contributions of this thesis (Section 6.1). This is followed by a review of the major findings and their limitations, offering a balanced perspective on the work conducted (Section 6.2). The chapter concludes by suggesting several promising avenues for future research, indicating how the present study could be further developed or refined (Section 6.3).

6.1 Summary

This thesis explored innovative deep learning frameworks for video-based assessment of Parkinson's disease severity. Methods were evaluated in alignment with UPDRS on a variety of motor tasks, including gait, finger tapping, hand movement, and leg agility, to provide a subtle understanding of functional mobility in individuals with PD. Beyond this primary focus, the ideas developed in this thesis are also useful for other areas in computer vision. To demonstrate their versatility, additional experiments were carried out, applying the methods to different computer vision tasks such as action recognition and other action quality assessment tasks, such as diving and surgical skill assessment.

Recent efforts to automate PD symptoms assessment often rely on wearable sensors, which can be costly and limited in scope. Video technology offers a less intrusive and more scalable alternative. Despite advances in deep learning, most video-based research uses skeleton data, which has limitations in accurately capturing subtle movements cru-

cial for PD assessment. This thesis focuses on using RGB video data to overcome these limitations through deep learning strategies. While RGB video data offers certain advantages for PD severity assessment, it is not without its own set of challenges. These can be summarised as follows.

- **Complexity of Movement** – Human movements are complex and can vary significantly among individuals, which becomes even more complicated when considering the symptoms of PD. This complexity requires the development of deep learning models that are sensitive enough to capture subtle spatial and temporal features for an accurate assessment of varying PD severity levels.
- **Camera Motion** – Camera motions introduce variability into video data for PD assessment, adding 'motion noise' that can interfere with the model's learning process, leading to less accurate assessments of PD severity.
- **Availability of Annotated Videos** – The preparation of large-scale, high-quality annotated data for PD assessment poses multiple challenges. Recording motor tasks often requires specialized equipment and environments. Annotation is both time-consuming and costly, as it requires the expertise of trained clinicians. Ethical and legal issues around patient privacy and data security add more complications. These limitations can hinder the generalisation capabilities of deep learning models, affecting their performance on new patient data.
- **Domain Discrepancy** – To address data scarcity in AQA, including those specific to PD, models often initialise with weights pretrained on generic, large-scale datasets. While this is better than starting from scratch, it can result in misalignment between the broader features captured in generic datasets and the more subtle patterns essential for accurate AQA. Thus, there is a need to better adapt these pretrained models for AQA tasks, e.g. PD.

To address these challenges, this thesis presented three different deep learning strategies.

- **End-to-end Supervised Learning** – This approach emphasised in-depth motion analysis by capturing both spatial and long-range temporal features, while also focussing on the most critical parts of the video to enable a more comprehensive understanding of disease severity.
- **Self-supervised Representation Learning** – This strategy focused on extracting robust, high-level visual representations from unlabelled PD video data, using pretext tasks to guide the learning process. The learned features were subsequently

employed as initial weights for downstream tasks aimed at PD severity assessment, thereby enhancing overall model performance.

- **Parameter-Efficient Continual Pretraining** – This strategy aimed to enhance the knowledge transfer from large-scale video datasets to specialised tasks like AQA in a computationally efficient way. By fine-tuning only a select set of parameters through SSL, the approach minimised computational demands while maintaining robust performance in PD severity assessment tasks.

6.2 Findings and Limitations

Chapter 3 employed a multi-stream deep learning architecture with a 3D CNN, serving as the backbone, to accurately capture spatial and temporal features of complex movements in PD assessment. The architecture incorporated a sparse temporal sampling strategy to capture long-range temporal structures in patient movements. Attention units were added to the model to emphasise critical video segments that are crucial for an accurate assessment. The model also reduced the impact of camera motion by incorporating motion boundary features. The effectiveness of this approach was validated on a dataset from 25 clinically diagnosed PD patients, obtaining 72.3% and 77.1% top-1 accuracy on hand movement and gait tasks, respectively.

The limitations of the method presented in Chapter 3 are multiple and noteworthy. First, the evaluation was limited to only two specific PD tasks, which restricts the applicability and robustness of the model across a more comprehensive range of PD symptoms and motor functions. Assessing the model on a larger set of tasks would allow for a more thorough evaluation of its versatility and effectiveness. Second, the UPDRS scores were categorised into three broad classes: 0, (1, 2), and (3, 4). While this approach made the classification task more manageable, it risks losing nuanced information about PD severity, diminishing the model's ability to capture the full range of disease progression. Lastly, the method relies on supervised learning, which, although effective under certain conditions, has limitations, especially in scenarios with limited annotated data, where such approaches often fail to generalise effectively.

In Chapter 4, we addressed the issue of data scarcity in Parkinson's disease assessment by focusing on self-supervised learning methods. Recognising the limitations of traditional self-supervised learning methods, particularly their scale-dependency and computational cost, an innovative pretraining process was introduced that mitigates the need for large-scale datasets. This approach used an auxiliary stage based on Similarity-based Knowledge Distillation (auxSKD) to enhance the adaptability and generalisation of SSL

models, particularly when pretrained on small size datasets like our PD dataset. In this chapter, we also proposed a new pretext task, VSPP, designed to help the model better understand the natural pace of video clips. This feature is especially valuable in handling the complex, variable-speed actions often observed in PD patients.

Furthermore, we introduced a new annotated AQA dataset, PD4T, for the vision community to evaluate various actions performed by actual PD patients. This dataset contains four motor tasks including gait, finger tapping, and leg agility, offering a valuable resource for future PD research. The SSL framework we introduced outperformed conventional supervised pretraining on average across four tasks of PD4T. Additionally, this framework demonstrated its robustness in action recognition, outperforming state-of-the-art SSL methods like VCOP [157], VideoPace [148], and RSPNet [16] in benchmarks such as UCF101 [134] and HMDB51 [75], even when pretrained on a reduced-size dataset, e.g. Kinetics-100 instead of Kinetics-400.

In terms of limitations, the two-stage pretraining mechanism in our SSL framework introduces the issue of parameter inefficiency. Specifically, both the auxiliary learning stage (auxSKD) and the primary pretraining stage (VSPP) must be pretrained for each task independently. This not only increases the computational cost but also poses challenges for real-world scenario, especially in the context of PD assessment. In a healthcare setting, where rapid and efficient analysis is often critical, the need to pretrain multiple stages for each specific PD motor task can be impractical. Also, PD symptoms can change a lot, both between different people and in the same person over time, so the model needs to be updated frequently. The separate pretraining processes for each task thus present a scalability issue, hindering the framework's ability to adapt quickly to the continually evolving nature of PD symptoms.

Beyond the above issue, the VSPP pretext task itself also has limitations worth noting. A core feature of the VSPP is its reliance on the altered pace of movement in a specific video segment while maintaining the rest of the clip at its natural pace. However, this assumption may be compromised if the clip contains sudden or extremely quick motions that might be missed during sampling. This is a drawback that is also observed in other speed-based pretext tasks, such as VideoPace [148] and RSPNet [16]. Moreover, our approach does not incorporate an appearance stream, unlike some other speed-related tasks such as ASCNet [65], RSPNet, and VideoPace. Although this may seem like a limitation, it was an intentional choice to focus on the efficacy of VSPP as an independent pretext task and to emphasise the auxiliary pretraining stage.

In Chapter 5, we introduced a parameter-efficient continual pretraining framework, called

PECoP, as an innovative advance in the AQA transfer learning pipeline. Through the introduction of 3D-Adapter, a compact bottleneck layer, we enabled fine-tuning of pre-trained 3D CNNs for domain-specific spatiotemporal features without altering the original model parameters. This method not only reduced computational and storage cost, but also mitigated issues like overfitting and catastrophic forgetting, often seen in continual pretraining paradigms. We demonstrated PECoP's ability to enhance the performance of recent state-of-the-art AQA methods (MUSDL [138], CoRe [166], TSA [159], and TPT [8]), leading to considerable improvements on benchmark AQA datasets, JIGSAWS (↑ 6.0%), MTL-AQA (↑ 0.99%), and FineDiving (↑ 2.54%). Furthermore, when applied to the PD4T dataset, PECoP surpassed the state-of-the-art, showing a performance improvement of ↑ 3.56% in comparison.

However, one of the main limitations inherent to a parameter-efficient approach like PECoP is the potential trade-off between the model's efficiency and its ability to handle complexity. Utilising a limited set of adaptable parameters effectively minimises the chances of overfitting the model to the training data. However, this strategy could unintentionally cause the model to underfit, limiting its capacity to capture complex spatiotemporal characteristics. This limitation becomes especially significant in the context of AQA tasks, where the precise capture of subtle features is often essential for accurate assessments.

6.3 Directions for Future Work

Five potential avenues for future studies are outlined.

6.3.1 Advanced Methods for Class Imbalance

Given the imbalanced nature of the PD dataset in Chapter 3, UPDRS scores were categorised into three broad classes. Although this strategy simplified the classification task, it could have led to a loss of detailed information about the progression of PD severity. Future research might explore advanced machine learning techniques for better class balance. For example, a promising option could be meta-learning [77, 130], exemplified by tools like Meta-Weight-Net [130], which adaptively alters loss weights during training, thus tackling class imbalances more efficiently than traditional methods such as focal loss.

6.3.2 Appearance Stream for VSPP

The VSPP pretext task proposed in Chapter 4 primarily targets temporal features. While effective, the framework could be enriched by incorporating spatial features. A promising avenue for future research is to integrate an appearance stream into VSPP to enhance its ability to effectively capture a broader range of features and increase the model's versatility and robustness. This enhancement is expected to make the model more practical and effective across a range of applications, like more accurately identifying specific actions, better detecting unusual activities in security videos, and offering improved interaction with users by combining the visual aspects with the dynamics of movement.

6.3.3 Assessing PECoP in Different Architectures and Tasks

While PECoP has been primarily evaluated using 3D CNN backbones for AQA tasks in Chapter 5, its broader applicability remains an open question. The effectiveness of the framework with alternative architectures, such as transformers, has not yet been explored. Moreover, the versatility of PECoP could extend its utility to various other vision tasks like action recognition and few-shot learning. Validating PECoP's performance in these areas could open up new avenues for its application, potentially making it a more universally useful tool in the field of computer vision.

6.3.4 Utilising Enhanced Data Augmentation Techniques

In this thesis, we have utilised weak data augmentation techniques - jittering, cropping, and flipping — for spatial and temporal transformations to improve the robustness of our deep learning models for PD severity assessment. While these methods introduce data variability, they may not capture the full spectrum of PD symptoms and their variations. Therefore, the model's ability to generalise to new, unseen data could be compromised.

Moreover, there is a risk associated with the application of strong data augmentation methods, such as those employing generative models [58]. These sophisticated techniques could potentially lead to the generation of non-representative samples that deviate significantly from the underlying data distribution. In the context of Parkinson's disease, this deviation might lead to the creation of videos that inaccurately reflect the disease's severity, potentially causing the model to misjudge the stage of Parkinson's.

To overcome these limitations, exploring advanced data augmentation strategies, such as VITA [15], is crucial. Such strategies are capable of producing realistic and varied

on-manifold samples, which equip models with an extensive, accurate representation of PD examples. This enhancement could significantly boost the models' generalisation capabilities, leading to more precise and reliable severity predictions.

6.3.5 Advancing Generalisation in Parkinson's Disease Severity Assessment

As we look towards the future in PD severity assessment, the need for models with strong generalisation capabilities is becoming more apparent. The groundwork laid by this thesis through various learning strategies is just the beginning. The long-term research direction necessitates enhancing models to intuitively grasp and interpret the range of PD symptoms and their variations. The aim is to create models that can seamlessly transfer learning from one patient context to another, reducing the need for continuous retraining or tuning. Future efforts will likely involve gathering extensive and diverse datasets that encompass the full spectrum of PD symptoms. Training models on such comprehensive data will enable them to adapt more effectively to new scenarios without constant updates. Moreover, ongoing refinement of our algorithms is essential to ensure their effectiveness in various settings, from hospitals to patients' homes, while maintaining a balance between accuracy and practical usability.

References

[1] European parkinson's disease association (epda), 2020. Available at: https://www.epda.eu.com/about-parkinson-s/what-is-parkinson-s/. 1

[2] S. Abbasi Koohpayegani, A. Tejankar, and H. Pirsiavash. Compress: Self-supervised learning by compressing representations. *Advances in Neural Information Processing Systems*, 33:12980–12992, 2020. vi, 33, 34, 35, 56, 57

[3] E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, and V. Venkatraman. Gait and tremor investigation using machine learning techniques for the diagnosis of Parkinson disease. *Future Generation Computer Systems*, 83:366–373, 2018. 9

[4] U. Ahsan, R. Madhok, and I. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019. v, 21, 23

[5] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33, 2020. 64

[6] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3478–3488, 2021. 36

[7] S. Azizi, L. Culp, J. Freyberg, B. Mustafa, S. Baur, S. Kornblith, T. Chen, P. MacWilliams, S. S. Mahdavi, E. Wulczyn, et al. Robust and efficient medical imaging with self-supervision. *arXiv preprint arXiv:2205.09723*, 2022. 35, 36, 77, 93

[8] Y. Bai, D. Zhou, S. Zhang, J. Wang, E. Ding, Y. Guan, Y. Long, and J. Wang. Action quality assessment with temporal parsing transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 422–438. Springer, 2022. v, 9, 10, 11, 14, 15, 16, 17, 82, 85, 87, 89, 99

[9] S. Benaim, A. Ephrat, O. Lang, I. Mosseri, W. T. Freeman, M. Rubinstein, M. Irani, and T. Dekel. SpeedNet: learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020. 21, 24, 57, 61, 66

[10] J. Carreira and A. Zisserman. 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, 2017. iv, vii, 12, 39, 44, 45, 48, 49, 51, 52, 72, 78

[11] C.-M. Chang, Y.-L. Huang, J.-C. Chen, and C.-C. Lee. Improving Automatic Tremor and Movement Motor Disorder Severity Assessment for Parkinson's Disease with Deep Joint Training. In *EMBC*, pages 3408–3411. IEEE, 2019. 9

[12] M.-N. Chapel and T. Bouwmans. Moving Objects Detection with a Moving Camera: A Comprehensive Review. *arXiv preprint arXiv:2001.05238*, 2020. 45

[13] H. Chen, R. Tao, H. Zhang, Y. Wang, W. Ye, J. Wang, G. Hu, and M. Savvides. Conv-Adapter: exploring parameter efficient transfer learning for convnets. *arXiv preprint arXiv:2208.07463*, 2022. 38, 80, 81

[14] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22): 4712, 2021. 20

[15] M. Chen, C. Wen, F. Zheng, F. He, and L. Shao. Vita: A multi-source vicinal transfer augmentation method for out-of-distribution generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 321–329, 2022. 100

[16] P. Chen, D. Huang, D. He, X. Long, R. Zeng, S. Wen, M. Tan, and C. Gan. RSPNet: relative speed perception for unsupervised video representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1045–1053, 2021. vi, xi, 24, 27, 28, 55, 57, 61, 62, 63, 66, 67, 75, 76, 81, 88, 89, 98

[17] S. Chen, G. Chongjian, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems*, 2022. vii, 37, 38, 93

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 36

[19] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 26

[20] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 36

[21] J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019. 33

[22] A. Dadashzadeh, A. Whone, M. Rolinski, and M. Mirmehdi. Exploring motion boundaries in an end-to-end network for vision-based parkinson's severity assessment. *arXiv preprint arXiv:2012.09890*, 2020. 39, 77

[23] A. Dadashzadeh, A. Whone, and M. Mirmehdi. Auxiliary learning for self-supervised video representation via similarity-based knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4231–4240, 2022. viii, xi, 57, 79, 84, 89

[24] A. Dadashzadeh, S. Duan, A. Whone, and M. Mirmehdi. Pecop: Parameter efficient continual pretraining for action quality assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 42–52, 2024. 78

[25] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441. Springer, 2006. 39, 45, 46

[26] A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE international conference on computer vision*, pages 567–575, 2015. 21

[27] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, pages 1–16, 2023. 37

[28] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 21

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. vii, 37, 38

[30] H. Doughty, W. Mayol-Cuevas, and D. Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7862–7871, 2019. 9

[31] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1801–1810, 2019. 29

[32] A. Elkholy, M. E. Hussein, W. Gomaa, D. Damen, and E. Saba. Efficient and robust skeleton-based quality assessment and abnormality detection in human action

performance. *IEEE journal of biomedical and health informatics*, 24(1):280–291, 2019. 18

[33] M. Endo, K. L. Poston, E. V. Sullivan, L. Fei-Fei, K. M. Pohl, and E. Adeli. Gaitforemer: Self-supervised pre-training of transformers via human motion forecasting for few-shot gait impairment severity estimation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*, pages 130–139. Springer, 2022. 9, 77

[34] D. Epstein, B. Chen, and C. Vondrick. Oops! predicting unintentional action in video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 919–929, 2020. 24

[35] Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu. Seed: Self-supervised distillation for visual representation. *arXiv preprint arXiv:2101.04731*, 2021. 34

[36] Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu. Seed: Self-supervised distillation for visual representation. In *International Conference on Learning Representations*, 2021. 33, 56, 57

[37] S. Farabi, H. Himel, F. Gazzali, M. B. Hasan, M. H. Kabir, and M. Farazi. Improving action quality assessment using weighted aggregation. In *Pattern Recognition and Image Analysis: 10th Iberian Conference, IbPRIA 2022, Aveiro, Portugal, May 4–6, 2022, Proceedings*, pages 576–587. Springer, 2022. 17, 86

[38] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 49, 51

[39] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. 21, 55

[40] P. C. Fino and M. Mancini. Phase-dependent effects of closed-loop tactile feedback on gait stability in parkinson's disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(7):1636–1641, 2020. 2

[41] J. Frankle, D. J. Schwab, and A. S. Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. In *International Conference on Learning Representations*. 77

[42] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, 2014. 7, 17, 78, 79, 83, 85

[43] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 21

[44] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. T. Stebbins, S. Fahn, P. Martinez-Martin, W. Poewe, C. Sampaio, M. B. Stern, R. Dodel, et al. Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): scale presentation and clinimetric testing results. *Movement disorders: official journal of the Movement Disorder Society*, 23(15):2129–2170, 2008. 70, 87

[45] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. T. Stebbins, S. Fahn, P. Martinez-Martin, W. Poewe, C. Sampaio, M. B. Stern, R. Dodel, et al. Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results. *Movement Disorders: official Journal of the Movement Disorder Society*, 23(15):2129–2170, 2008. 2, 19

[46] A. S. Gordon. Automated video assessment of human performance. In *Proceedings of AI-ED*, volume 2, 1995. 10

[47] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020. 25, 34, 55, 58

[48] R. Guo, H. Li, C. Zhang, and X. Qian. A tree-structure-guided graph convolutional network with contrastive learning for the assessment of parkinsonian hand movements. *Medical Image Analysis*, 81:102560, 2022. 2, 20, 77

[49] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020. 35, 77

[50] X. Han and J. Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. *arXiv preprint arXiv:1904.02817*, 2019. 77

[51] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 3154–3160, 2017. 90

[52] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 63

[53] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021. 37, 78

[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 26, 63, 88

[55] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference*

on computer vision and pattern recognition, pages 9729–9738, 2020. 20, 25, 26, 28, 34, 55, 59, 88

[56] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, and L. Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *ACL/IJCNLP (1)*, 2021. 78, 93

[57] O. J. Hénaff, S. Koppula, J.-B. Alayrac, A. v. d. Oord, O. Vinyals, and J. Carreira. Efficient visual pretraining with contrastive detection. *arXiv preprint arXiv:2103.10957*, 2021. 55

[58] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 100

[59] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019. 33

[60] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 32, 33

[61] M. A. Hobert, S. Nussbaum, T. Heger, D. Berg, W. Maetzler, and S. Heinzel. Progressive gait deficits in Parkinson's disease: A wearable-based biannual 5-year prospective study. *Frontiers in Aging Neuroscience*, 11:22, 2019. 2

[62] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 37, 78, 80

[63] M. D. Hssayeni, J. Jimenez-Shahed, M. A. Burack, and B. Ghoraani. Wearable sensors for estimation of Parkinsonian tremor severity during free body movements. *Sensors*, 19(19):4215, 2019. 2, 16

[64] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 37, 80, 81

[65] D. Huang, W. Wu, W. Hu, X. Liu, D. He, Z. Wu, X. Wu, M. Tan, and E. Ding. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8096–8105, October 2021. vi, 24, 27, 28, 29, 63, 65, 66, 67, 98

[66] Y. Huo, M. Ding, H. Lu, Z. Huang, M. Tang, Z. Lu, and T. Xiang. Self-supervised video representation learning with constrained spatiotemporal jigsaw. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 751–757. International Joint Conferences on

Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/104. Main Track. 21, 23

[67] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016. 21

[68] S. Jenni, G. Meishvili, and P. Favaro. Video representation learning by recognizing temporal transformations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 425–442. Springer, 2020. 24, 55

[69] H. Jeon, W. Lee, H. Park, H. J. Lee, S. K. Kim, H. B. Kim, B. Jeon, and K. S. Park. Automatic classification of tremor severity in Parkinson's disease using a wearable device. *Sensors*, 17(9):2067, 2017. 16

[70] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 36

[71] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4, 15, 77

[72] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 49, 62

[73] N. Komodakis and S. Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 55

[74] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 36

[75] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 7, 57, 62, 98

[76] P. H. Le-Khac, G. Healy, and A. F. Smeaton. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020. 25

[77] H. B. Lee, H. Lee, D. Na, S. Kim, M. Park, E. Yang, and S. J. Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. *arXiv preprint arXiv:1905.12917*, 2019. 99

[78] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 21, 66

[79] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. 77

[80] Q. Lei, J.-X. Du, H.-B. Zhang, S. Ye, and D.-S. Chen. A survey of vision-based human action evaluation methods. *Sensors*, 19(19):4129, 2019. 2

[81] M. Li, H.-B. Zhang, Q. Lei, Z. Fan, J. Liu, and J.-X. Du. Pairwise contrastive learning network for action quality assessment. In *Computer Vision–ECCV 2022: 17th European Conference, 2022, Proceedings, Part IV*, pages 457–473. Springer, 2022. 86

[82] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 36

[83] Y. Li, X. Chai, and X. Chen. End-to-end learning for action quality assessment. In *Pacific Rim Conference on Multimedia*, pages 125–134. Springer, 2018. 17

[84] Z. Li, L. Gu, W. Wang, R. Nakamura, and Y. Sato. Surgical skill assessment via video semantic aggregation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VII*, pages 410–420. Springer, 2022. xi, 86

[85] Y. Liao, A. Vakanski, and M. Xian. A deep learning framework for assessing physical rehabilitation exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(2):468–477, 2020. 19

[86] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 19, 46

[87] X. Lin, H. S. Baweja, G. Kantor, and D. Held. Adaptive auxiliary task weighting for reinforcement learning. *Advances in neural information processing systems*, 32, 2019. 31

[88] Y. Lin, X. Guo, and Y. Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8239–8249, 2021. 55

[89] D. Liu, Q. Li, T. Jiang, Y. Wang, R. Miao, F. Shan, and Z. Li. Towards unified surgical skill assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2021. xi, 9, 29, 30, 86

[90] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma. T-C3D: Temporal convolutional 3D network for real-time action recognition. In *Thirty-second AAAI Conference on Artificial Intelligence*, 2018. 22, 44

[91] S. Liu, A. J. Davison, and E. Johns. Self-supervised generalisation with meta auxiliary learning. *arXiv preprint arXiv:1901.08933*, 2019. 31, 55

[92] W. Liu, X. Lin, X. Chen, Q. Wang, X. Wang, B. Yang, N. Cai, R. Chen, G. Chen, and Y. Lin. Vision-based estimation of mds-updrs scores for quantifying parkinson's disease tremor severity. *Medical Image Analysis*, 85:102754, 2023. 2, 20

[93] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 36

[94] Y. Liu, L. Wang, X. Ma, Y. Wang, and Y. Qiao. Fineaction: A fined video dataset for temporal action localization. *arXiv preprint arXiv: 2105.11107*, 2021. 75

[95] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, pages 7834–7843, 2018. 40

[96] M. Lu, K. Poston, A. Pfefferbaum, E. V. Sullivan, L. Fei-Fei, K. M. Pohl, J. C. Niebles, and E. Adeli. Vision-based estimation of mds-updrs gait scores for assessing parkinson's disease motor severity. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pages 637–647. Springer, 2020. v, 2, 9, 19, 20

[97] M. Lu, Q. Zhao, K. L. Poston, E. V. Sullivan, A. Pfefferbaum, M. Shahid, M. Katz, L. M. Kouhsari, K. Schulman, A. Milstein, et al. Quantifying parkinson's disease motor severity under uncertainty using mds-updrs videos. *Medical Image Analysis*, 73:102179, 2021. 2, 19, 77

[98] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11701–11708, 2020. 55

[99] A. Marcante, R. Di Marco, G. Gentile, C. Pellicano, F. Assogna, F. E. Pontieri, G. Spalletta, L. Macchiusi, D. Gatsios, A. Giannakis, et al. Foot pressure wearable sensors for freezing of gait detection in parkinson's disease. *Sensors*, 21(1):128, 2020. 2, 16

[100] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 21, 66

[101] M. H. Monje, G. Foffani, J. Obeso, and Á. Sánchez-Ferro. New sensor and wearable technologies to aid in the diagnosis and treatment monitoring of parkinson's disease. *Annual review of biomedical engineering*, 21:111–143, 2019. 2, 16

[102] C. Morgan, A. Masullo, H. Isotalus, E. Tonkin, M. Mirmehdi, F. Jovan, T. Whone, G. Oikonomou, R. McConville, G. Tourte, et al. Real-world sit-to-stand evaluation.

In *MOVEMENT DISORDERS*, volume 37, pages S195–S196. WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2022. 77

[103] T. N. Mundhenk, D. Ho, and B. Y. Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9339–9348, 2018. 21

[104] A. Navon, I. Achituve, H. Maron, G. Chechik, and E. Fetaya. Auxiliary learning by implicit differentiation. *arXiv preprint arXiv:2007.02693*, 2020. 31

[105] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pages 6752–6761, 2018. 45

[106] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 21, 22, 23

[107] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 28

[108] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi. Online quality assessment of human movement from skeleton data. In *British Machine Vision Conference*, pages 153–166. BMVA press, 2014. 18

[109] J.-H. Pan, J. Gao, and W.-S. Zheng. Action assessment by joint relation graphs. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6331–6340, 2019. 11, 17, 86

[110] J.-H. Pan, J. Gao, and W.-S. Zheng. Adaptive action assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8779–8795, 2021. 17

[111] T. Pan, Y. Song, T. Yang, W. Jiang, and W. Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11205–11214, 2021. v, 27, 55, 63, 64, 66

[112] W. Park, D. Kim, Y. Lu, and M. Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 33

[113] P. Parmar and B. T. Morris. Action quality assessment across multiple actions. *arXiv preprint arXiv:1812.06367*, 2018. 10, 11, 17, 77

[114] P. Parmar and B. T. Morris. What and how well you performed? a multitask learning approach to action quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 304–313, 2019. 2, 7, 10, 11, 17, 79, 83, 85, 86

[115] P. Parmar and B. Tran Morris. Learning to score olympic events. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 20–28, 2017. 9, 11, 17, 77, 86

[116] W. Pei, T. Baltrusaitis, D. M. Tax, and L.-P. Morency. Temporal attention-gated model for robust sequence classification. In *CVPR*, pages 6730–6739, 2017. 40

[117] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019. 33

[118] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. AdapterHub: a framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, 2020. 35, 78

[119] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In *European conference on computer vision*, pages 556–571. Springer, 2014. iv, 9, 10, 17

[120] W. Poewe, K. Seppi, C. M. Tanner, G. M. Halliday, P. Brundin, J. Volkmann, A.-E. Schrag, and A. E. Lang. Parkinson disease. *Nature reviews Disease primers*, 3(1):1–21, 2017. 1

[121] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, et al. Self-supervised pretraining improves self-supervised pretraining. *arXiv preprint arXiv:2103.12718*, 2021. xi, 55, 86

[122] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, et al. Self-supervised pretraining improves self-supervised pretraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2584–2594, 2022. xi, 35, 77, 85, 86, 87, 89, 93

[123] J. D. Rennie and N. Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, volume 1. AAAI Press, Menlo Park, CA, 2005. 19

[124] K. Roditakis, A. Makris, and A. Argyros. Towards improved and interpretable action quality assessment with self-supervised alignment. In *The 14th PErvasive Technologies Related to Assistive Environments Conference*, pages 507–513, 2021. vi, 10, 11, 17, 29, 30

[125] S. Rupprechter, G. Morinan, Y. Peng, T. Foltynie, K. Sibley, R. S. Weil, L.-A. Leyland, F. Baig, F. Morgante, R. Gilron, et al. A clinically interpretable computer-vision based method for quantifying gait in parkinson's disease. *Sensors*, 21(16):5437, 2021. 2

[126] A. Samà, C. Pérez-López, J. Romagosa, D. Rodriguez-Martin, A. Català, J. Cabestany, D. Perez-Martinez, and A. Rodríguez-Molinero. Dyskinesia and motor state detection in parkinson's disease patients with a single movement sensor. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1194–1197. IEEE, 2012. 1, 16

[127] M. C. Schiappa, Y. S. Rawat, and M. Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37, 2023. 21

[128] A.-K. Seifert, M. G. Amin, and A. M. Zoubir. Toward unobtrusive in-home gait analysis based on radar micro-doppler signatures. *IEEE Transactions on Biomedical Engineering*, 66(9):2629–2640, 2019. 16

[129] B. Shi, J. Hoffman, K. Saenko, T. Darrell, and H. Xu. Auxiliary task reweighting for minimum-data learning. In *NeurIPS*, 2020. 31

[130] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019. 99

[131] L. Sigcha, N. Costa, I. Pavón, S. Costa, P. Arezes, J. M. López, and G. De Arcas. Deep Learning Approaches for Detecting Freezing of Gait in Parkinson's Disease Patients through On-Body Acceleration Sensors. *Sensors*, 20(7):1895, 2020. 16

[132] A. L. Silva de Lima, L. J. Evers, T. Hahn, L. Bataille, J. L. Hamilton, M. A. Little, Y. Okuma, B. R. Bloem, and M. J. Faber. Freezing of gait and fall detection in parkinson's disease using wearable sensors: a systematic review. *Journal of neurology*, 264:1642–1654, 2017. 2

[133] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 45, 49, 51

[134] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 7, 57, 62, 98

[135] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13693–13696, 2020. 55

[136] Y. Su, X. Wang, Y. Qin, C.-M. Chan, Y. Lin, H. Wang, K. Wen, Z. Liu, P. Li, J. Li, et al. On transferability of prompt tuning for natural language processing. *arXiv preprint arXiv:2111.06719*, 2021. 36

[137] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 44

[138] Y. Tang, Z. Ni, J. Zhou, D. Zhang, J. Lu, Y. Wu, and J. Zhou. Uncertainty-aware score distribution learning for action quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9839–9848, 2020. iv, 2, 9, 10, 11, 12, 13, 15, 16, 17, 30, 77, 82, 84, 85, 86, 89, 99

[139] A. Tejankar, S. A. Koohpayegani, V. Pillai, P. Favaro, and H. Pirsiavash. Isd: Self-supervised learning by iterative similarity distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9609–9618, 2021. 33, 34, 56, 57, 59

[140] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019. 32, 33, 56, 57

[141] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 55

[142] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 11

[143] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. viii, 63, 64

[144] F. Tung and G. Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019. 32, 33, 34

[145] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015. 31

[146] G. Wang, Y. Zhou, C. Luo, W. Xie, W. Zeng, and Z. Xiong. Unsupervised visual representation learning by tracking patches in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2563–2572, 2021. 55

[147] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 45

[148] J. Wang, J. Jiao, and Y.-H. Liu. Self-supervised video representation learning by pace prediction. In *European conference on computer vision*, pages 504–521. Springer, 2020. v, xi, 21, 24, 55, 57, 60, 61, 63, 64, 65, 66, 67, 68, 75, 81, 88, 89, 98

[149] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 40, 44, 45, 48, 51

[150] S. Wang, D. Yang, P. Zhai, C. Chen, and L. Zhang. TSA-Net: tube self-attention network for action quality assessment. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4902–4910, 2021. 10, 86

[151] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 40

[152] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1910–1919, 2019. 21

[153] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 40

[154] H. Wu, K. Xu, L. Song, L. Jin, H. Zhang, and L. Song. Domain-adaptive pretraining methods for dialogue understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 665–669, 2021. 35, 77

[155] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 25

[156] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y.-G. Jiang, and X. Xue. Learning to score figure skating sport videos. *IEEE transactions on circuits and systems for video technology*, 30(12):4578–4590, 2019. 17, 77

[157] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. v, 21, 22, 65, 66, 75, 98

[158] G. Xu, Z. Liu, X. Li, and C. C. Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020. vi, 31, 32

[159] J. Xu, Y. Rao, X. Yu, G. Chen, J. Zhou, and J. Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2949–2958, 2022. v, 7, 13, 15, 16, 17, 77, 79, 82, 83, 84, 85, 88, 99

[160] E. Yang, S. Nair, R. Chandradevan, R. Iglesias-Flores, and D. W. Oard. C3: Continued pretraining with contrastive weak supervision for cross language ad-hoc retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2507–2512, 2022. 35

[161] F. Yang, Y. Wu, S. Sakti, and S. Nakamura. Make skeleton-based action recognition model smaller, faster and better. In *Proceedings of the ACM multimedia asia*, pages 1–6. 2019. 19

[162] X. Yang, M. Mirmehdi, and T. Burghardt. Back to the future: Cycle encoding prediction for self-supervised contrastive video representation learning. *arXiv preprint arXiv:2010.07217*, 2020. 27, 66

[163] G. Yao, T. Lei, and J. Zhong. A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118:14–22, 2019. 20

[164] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6548–6557, 2020. 24

[165] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017. 33

[166] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou. Group-aware contrastive regression for action quality assessment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7919–7928, 2021. iv, v, 2, 9, 10, 11, 12, 14, 15, 16, 17, 71, 77, 82, 84, 85, 86, 88, 89, 99

[167] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007. 48

[168] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 33

[169] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 31

[170] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020. 33

[171] S.-J. Zhang, J.-H. Pan, J. Gao, and W.-S. Zheng. Semi-supervised action quality assessment with self-supervised segment feature recovery. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):6017–6028, 2022. 29, 30

[172] Y. Zhang, W. Xiong, and S. Mi. Learning time-aware features for action quality assessment. *Pattern Recognition Letters*, 158:104–110, 2022. 17, 86

[173] Y. Zhao, L. Tan, P. Lau, W. Au, S. Li, and N. Luo. Factors affecting health-related quality of life amongst Asian patients with Parkinson's disease. *European Journal of Neurology*, 15(7):737–742, 2008. 1