Skill Determination from Long Videos

Hazel Doughty

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

December 9, 2020

62132 words

Abstract

Skill determination in computer vision is the problem of evaluating how well a person performs a particular task, by analysing a recorded video of that performance. Typically, skill determination from video has focussed on short tasks, where the performance of a single action is evaluated in accordance with predefined scoring metrics. This thesis is the first work to explore 'general' skill determination and demonstrates that skill can be automatically determined from video for a variety of different tasks, ranging from surgery to drawing and rolling pizza dough, using the same method. To do this, the problem is formulated as a pairwise ranking of video collections, thus skill can be determined relative to other videos in a task and does not require task-specific knowledge or scoring metrics.

In long videos, parts of the video are often irrelevant for assessing skill and there may be variability in the skill exhibited throughout a video. Therefore, it is necessary to determine skill in long videos by attending to the skill-relevant parts. This thesis thus proposes an approach to train temporal attention modules, learned with only video-level supervision, which separately attends to video parts indicative of higher and lower skill.

Learning to determining skill in each task individually limits the ability to scale to a large number of tasks, due to the training and annotation cost. This thesis explores whether there are common features for determining skill shared across different tasks. It finds that there is potential for sharing information even between seemingly unrelated tasks, however it is difficult to predict what aspects tasks will share without external knowledge.

This thesis also presents the first method to learn adverbs from instructional videos. It identifies that adverbs in the narrations of instructional videos are often skill relevant as they describe how particular actions should be performed. Using weak-supervision from adverbs in the narrations of instructional videos the method is able to learn representations shared across different actions and tasks which describe the manner in which individual actions have been performed.

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

Firstly, I would like to thank my supervisors Walterio Mayol-Cuevas and Dima Damen for their help, guidance and support throughout my PhD. I have learnt so much about research from working with both of them. I would additionally like to thank Dima for the further opportunities and advice she has offered me over the years. Her mentoring has helped me get to where I am now.

Thank you to everyone in the VILab who have all made the lab a fun place to work: Davide, Mike, Will, Laurie, Faegheh, Sam, Jonny, Vangelis, Jian, Ramon, Abel, Perla, Eduardo, Sasha, Young, Toby, Xingrui, Yanan, Janis, Erik and Miguel. I would particularly like to thank Davide, Mike and Will for their moral support. Thanks also go to everyone who attended the virtual board game nights, which made writing up in lockdown much more enjoyable.

Finally, I would like to thank my mum, dad and sister for their encouragement and George for his support and understanding over the past four years.

Publications

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

1. Hazel Doughty, Dima Damen, Walterio Mayol-Cuevas. Who's Better? Who's Best? Pairwise Deep Ranking for Skill Determination. *Conference on Computer Vision and Pattern Recognition*, 2018.
2. Hazel Doughty, Walterio Mayol-Cuevas, Dima Damen. The Pros and Cons: Rank-aware Temporal Attention for Skill Determination in Long Videos. *Conference on Computer Vision and Pattern Recognition*, 2019.
3. Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, Dima Damen. Action Modifiers: Learning from Adverbs in Instructional Videos. *Conference on Computer Vision and Pattern Recognition*, 2020.

Additionally, during my PhD, I have contributed to the following papers:

1. Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price and Michael Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. *European Conference on Computer Vision*, 2018.
2. Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price and Michael Wray. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Contents

List of Figures

List of Tables

Acronyms

AMT	Amazon Mechanical Turk
AQA	Action Quality Assessment
ASR	Automatic Speech Recognition
BEST	Bristol Everyday Skill Tasks (Dataset)
BN	Batch Normalisation
BoW	Bag of Words
C3D	Convolutional 3D (Type of CNN)
CCA	Canonical Correlation Analysis
CMCS	Completeness Modelling and Context Separation
CMU-MMAC	Carnegie Mellon University Multi-Modal ACtivity (Dataset)
CNN	Convolutional Neural Network
fps	frames per second
GloVe	Global Vectors (Vectorised Word Representation)
GRU	Gated Recurrent Unit
HIT	Human Intelligence Task
HMM	Hidden Markov Model
HoF	Histogram of Oriented Flow
HoG	Histogram of Gradients
I3D	Inflated 3D (Type of CNN)
IDT	Improved Dense Trajectories
JIGSAWS	JHU-ISI Gesture and Skill Assessment Working Set
LSTM	Long-Short Term Memory

MAML	Model-Agnostic Meta-Learning
mAP	mean Average Precision
MIL	Multiple Instance Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
MSR-VTT	Microsoft Research Visual-to-Text (Dataset)
NCE	Noise Contrastive Estimation
NDCG	Normalised Discounted Cumulative Gain
OSATS	Objective Structured Assessment of Technical Skills (Scoring system for surgical skill)
P@1	Top 1 Precision
PoS	Part-of-Speech
ReLU	Rectified Linear Unit
RGB	Red Green Blue (colour model)
RNN	Recurrent Neural Network
TSN	Temporal Segment Network
t-SNE	t-distributed Stochastic Neighbourhood Embeddings
TTD	Trajectory-Pooled Deep-Convolutional Descriptor
TV-L^1	Total Variation and L^1 norm (dense optical flow method)
SIFT	Scale-Invariant Feature Transform
STIP	Space-Time Interest Points
STPN	Sparse Temporal Pooling Network
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
SVR	Support Vector Regression
VLAD	Vectors of Locally Aggregated Features
W-TALC	Weakly-Supervised Temporal Activity Localisation and Classification

Chapter 1

Introduction

Skill determination is the problem of assessing how well a subject performs a given task. Skill can be assessed by analysing a variety of different types of data, such as the time of completion, an image of the end result of the task or accelerometer data which represents the movement of the subject's hands or tools. This thesis focuses on determining skill from video, as videos capture information both about the motions and methods used to complete task and the intermediate and final results of the task. Thus, it suitable for capturing skill in a wide variety of different tasks.

How-to videos on sites such as YouTube and Vimeo, have enabled millions to learn new skills by observing others more skilled at the task. From drawing to cooking and repairing household items, learning from videos is nowadays a commonplace activity. However, these loosely organised collections normally contain a mixture of contributors with different levels of expertise. The querying person needs to decide who is better and who to learn from. Furthermore, the number of how-to videos is only increasing, fuelled by more cameras recording our daily lives. Being able to assess the skill of the subject, or rank the videos based on the skill displayed, would enable delving into the wealth of this online resource both for training humans and intelligent agents. This could allow for both automated feedback and better selection of demonstrations *e.g. which video should a robot imitate to prepare you scrambled eggs for breakfast?*.

Previous efforts in skill determination from video have focussed on specific tasks within the domains of surgery [10, 221] or sports [9, 134, 135, 147]. Methods were thus designed with prior knowledge about skill in the particular task and are therefore not widely applicable to other tasks. This thesis instead looks at determining skill for a wide variety of different tasks, with a focus on daily-living tasks, such as rolling pizza dough, drawing

or tying a tie, which do not have predefined scoring metrics.

Videos of daily-living tasks are typically long, *i.e.* minutes in length. They consist of many different actions which a subject performs to complete the goal of the task. There may be different sequences of actions which can be used to complete a task well and subjects performing the same sequences of actions may differ in their success. Thus determining skill is not a question of identifying whether a set sequence of actions has taken place. Instead, the aim is to determine how-well the overall task has been performed, while simultaneously identifying the key properties which indicate an expert performance of a particular task.

This thesis explores skill determination in long videos of daily-living tasks. It begins with a global view of the task and aims to learn to determine skill for a variety of tasks from only labelled data, without prior knowledge pertaining to individual tasks. The thesis then moves on to explore some of the issues unique to skill determination in long videos. It first identifies that many parts of a video are irrelevant for assessing skill and proposes a method to assess the overall skill in a long video by attending to its skill-relevant parts. It then looks at important actions within a task and learns to describe the manner in which these actions are performed using adverbs from instructional videos.

1.1 Challenges and Contributions

As eluded to above, there are two main challenges to tackle when learning to determine skill in daily-living tasks.

The first challenge is to be able to determine skill for a variety of tasks using the same method. When aiming to determine skill for specific tasks, prior knowledge about skillful performances in those tasks can be built into a model. For instance, measuring the distance the needle moves in surgery would not be applicable to rolling pizza dough or braiding hair. Instead, a method to determine skill in daily-living tasks should be widely applicable to many different tasks.

The second challenge is identifying the parts of a video most useful for determining skill. In long videos of daily-living tasks, there may be many parts of a task which are irrelevant to determining skill. For instance, gathering the ingredients and turning on the hob are not informative to how well a person makes scrambles eggs and will vary little in terms of skill between videos. It is therefore important to be able to learn which parts of a video are most useful to determining skill in addition to the features which indicate a skillful performance.

A third challenge is to overcome the above two challenges without requiring full-supervision of every component. It would be possible to learn the important parts of a task by labelling the usefulness of each video frame to determining skill. However, collecting these labels would be time-consuming and prevent a method from being applicable to many tasks. Therefore, to learn to determine skill in daily-living tasks it is necessary to be able to cope with a limited amount of labelled data.

The contributions of this thesis are:

- Chapter 3 formulates the problem of skill determination for daily-living tasks and presents the first method to determine skill in videos for a wide variety of tasks.
- Two datasets for skill determination in daily-living tasks are presented in this thesis, one in Chapter 3 and the other in Chapter 4.
- The challenges of fine-grained ranking of long videos are addressed in Chapter 4 by demonstrating the need for rank-aware temporal attention and proposing a model to learn this effectively.
- The transferability of features for skill determination are investigated in Chapter 5, with experiments on the end-to-end (Chapter 3) and rank-aware attention (Chapter 4) methods for multi-task learning and zero-shot transfer between tasks.
- Chapter 6 presents the first method for weakly-supervised learning from adverbs, in which relevant video segments are embedded in a latent space and adverbs are learnt as transformations in this space.

1.2 Thesis Overview

This thesis is organised as follows. Chapter 2 presents relevant work concerning understanding of long videos, ranking and skill determination.

Chapter 3 explores how to learn to determine skill from videos of daily-living tasks. It presents a method which ranks skill in a variety of tasks without requiring prior knowledge about each task. This chapter also discusses the collection and annotation of the first skill determination dataset for daily-living tasks, EPIC-Skills.

Chapter 4 focuses on the issues with determining skill in long videos, as only certain video parts will be relevant to determine skill. To tackle this problem, Chapter 4 presents a method to learn temporal attention for skill determination. It proposes a rank-aware loss which causes the model to focus on parts of the video indicative of high or low skill.

The collection of a second skill determination dataset, BEST, consisting of longer and more complex daily-living tasks is also covered in this chapter.

The methods presented in these two chapters are applicable to a wide variety of skill tasks, however the model weights for different skill tasks have to be learnt separately with individual ranking and attention. Chapter 5 examines this issue by focusing on multi-task and transfer learning within the context of skill determination.

Chapter 6 takes a closer look at skillful performances of individual actions within a task. Adverbs in the narrations of instructional videos are identified as a way to find which actions are relevant to skill as they indicate how these actions should be performed to complete the task well. The chapter proposes to learn a representation for these adverbs from the weak-supervision of the narrations, where these representations are learnt transformations in a video-text embedding space and are shared across actions and tasks. Thus, given instructions for a particular task, the adverb representations could be transferred to the new task to evaluate whether particular steps are performed well.

Finally, Chapter 7 concludes this thesis with a summary of the findings and directions for future work.

Chapter 2

Background

This chapter provides a background to the topic of skill determination and explains the necessary related work which this thesis builds upon. First, an overview of the relevant work in video understanding is given in Section 2.1, with specific focus on understanding of long videos and instructional videos. Section 2.2 then covers different techniques for the problems of ranking and retrieval which are relevant to this thesis. Finally, prior works on skill determination, action quality assessment and other related problems are discussed in Section 2.3.

2.1 Understanding of Long Videos

Video understanding covers a broad range of topics with the overall aim to be able to understand the contents of a video. The most well-studied problem within video understanding is the task of action recognition which typically uses temporally segmented clips of a few seconds in length.

However, there exist other problems within video understanding which require longer term temporal understanding as the activity or task may occur over several minutes. Furthermore, in many applications it is unreasonable to expect that videos will be well segmented therefore it may be necessary to deal with untrimmed and noisy videos which contain parts irrelevant to the target problem. There has been recent work which explores this problem, particularly in the context of action localisation [96, 125, 138, 195], and aims to discover the relevant parts of a video during the learning process without temporal annotations indicating where in a video these parts are.

Another area where long videos are prevalent is instructional videos. These videos are

commonly available on sites such as YouTube and demonstrate how to complete a particular task. Instructional videos can be especially useful for learning as not only do they contain video data which demonstrates how a task should be performed, they also contain instructor's narrations explaining the task and the steps being shown in the video. However, the narrations tend to only be roughly aligned with the video and may also contain parts irrelevant to the task being explained.

This section will begin looking at video understanding in general with Section 2.1.1. Section 2.1.1 will mainly examine methods learnt from temporally segmented video clips because this is where earlier work was focussed, with many techniques later being borrowed for understanding of long videos. Section 2.1.2 will then cover related works which aim to understand longer and untrimmed videos, with Section 2.1.3 focussing particularly on works which learn from instructional videos.

2.1.1 Video Understanding with Clips

A large amount of work in video understanding has been focussed on the task of action (or activity) recognition. In this task, the aim is to predict the action class present in a given short video clip. While this thesis focuses on skill determination in *long videos*, it is necessary to first see how video understanding is performed on short clips, as elements of such methods are applicable to longer videos.

Handcrafted Features. Before deep learning, handcrafted features were typically used for action recognition and other related video understanding tasks. For instance, Laptev *et al.* [88] used Scale-Invariant Feature Transform (SIFT) [102] to find and track interest points throughout a video. Local features such as histogram of oriented gradients (HoG) [27] and histogram of optical flow (HoF) [28] are then extracted around these tracked points. These features are pooled with Bag of Words (BoW) [26] to provide a more compact representation of the video features and a Support Vector Machine (SVM) is used to classify the action in a clip. This was the standard pipeline for many years, with new methods proposing to improve certain elements of the pipeline, such as the feature sampling method [187, 188, 189] or the encoding of the visual features [142].

One prominent example of the improvement of this pipeline is Improved Dense Trajectories (IDT) [187]. In this paper, Wang *et al.* build on the dense trajectory feature tracker from [189], which was shown to find higher overall quality trajectories than prior approaches like SIFT. The authors propose to improve sampling of features by first removing camera motion. This camera motion is estimated by matching keypoints across frames with a combination of SURF features [8] and dense optical flow. Removing the

camera motion improves the reliability of motion-based feature descriptors, such as HoF, and therefore the performance of the method on action recognition. The results are further improved by excluding any keypoints within a detected human body region from the camera motion estimation.

Advent of CNNs. Convolutional Neural Networks [89] (CNNs) have the capability to replace all the stages of the common pipeline as a single neural network can be trained end-to-end to learn a hierarchy of features from raw pixel values to an output classifier. CNNs are especially suited to images as multiple layers of convolutions and pooling can effectively extract information from adjacent pixels and down-sample the image to represent more and more abstract features as further layers are added. CNNs saw an explosion of usage for image classification after they demonstrated superior performance to handcrafted features [82] on large-scale image datasets such as ImageNet [32]. Initial attempts to apply CNNs to video understanding [126] simply treated video frames as individual images and applied a CNN to frames independently. However, this ignores additional temporal aspect of video understanding problems as no motion is encoded.

Encoding Motion with CNNs. To encode this motion, Ji *et al.* [70] propose a 3D convolutional neural network. As well as operating over the spatial aspect of the video frames, the 3D convolutions in this network operate over the temporal dimension of a stack of multiple contiguous frames. A downside of these 3D convolutions is that they operate over a relatively small portion of the video temporally. Therefore, Ji *et al.* combine the 3D CNN with auxiliary handcrafted motion features and regularise the 3D-CNN to produce feature vectors close to these auxiliary features. While this paper shows some promising results for CNNs in video understanding, even with this regularisation the 3D CNN still struggles to outperform non deep learning methods on the KTH action recognition dataset [158].

Karpathy *et al.* [78] attribute the under-performance of CNNs in video understanding to the lack of large scale video classification datasets. Therefore, they propose the Sports-1M dataset, a dataset of 1 million YouTube videos over 487 different sports classes. The authors also propose a multi-stream CNN architecture which processes input frames at multiple spatial resolutions: one low-resolution stream of the full image, and another high-resolution stream of a centre crop of the image. To improve robustness when training the model, 20 clips are sampled from each video and after first cropping the centre region, additional random crops are taken from this region with a 50% probability of the crop being flipped. These pre-processing techniques were adopted by later methods and remain common in CNNs developed for video understanding problems.

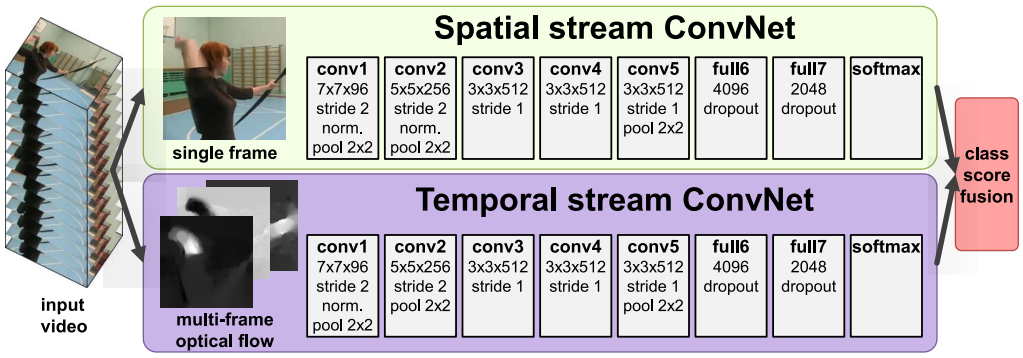

Figure 2.1: *Two stream architecture for video classification. In addition to the spatial CNN which takes a single video frame as input, there is a temporal CNN which takes multiple frames of dense optical flow extracted from the video. The predictions from each CNN are then fused to obtain a final score. Figure from [166].*

With this architecture, Kaparthy *et al.* investigate different strategies of fusing information from consecutive video frames to gain some temporal understanding. They find slow fusion of temporal information throughout multiple layers of the network to be the most successful approach, although the increase over fusing either at the very beginning or very end of the network is marginal, and using only a single frame displays relatively strong performance. This suggests that these methods of fusing features over multiple video frames do not actually encode much motion information.

To better encode motion information, Simonyan and Zisserman [166] propose to include a second CNN which operates on stacks of optical flows images, in addition to a CNN on RGB frames. These optical flow images can be produced with any dense optical flow method, Simonyan and Zisserman use Brox [12], while more recent methods use TV-L^1 [211]. The goal is for the CNN which takes RGB video frames as input to encode relevant spatial information, while the CNN using optical flow frames will predominantly learn temporal information. The two streams are trained independently and output scores from each stream are then combined to produce the final predictions (see Figure 2.1). Simonyan and Zisserman found late fusion via averaging to be most successful, with learning an additional linear SVM to fuse the features resulting in over-fitting.

The inclusion of the temporal stream with images of dense optical flow gave a vast improvement over a single spatial CNN and was one of the first deep learning works to obtain results comparable with IDT in action recognition. This work was highly influential in the field of video understanding and a two-stream approach with RGB frames and stacks of precomputed dense optical flow frames became the de-facto approach in action recognition and other video understanding tasks.

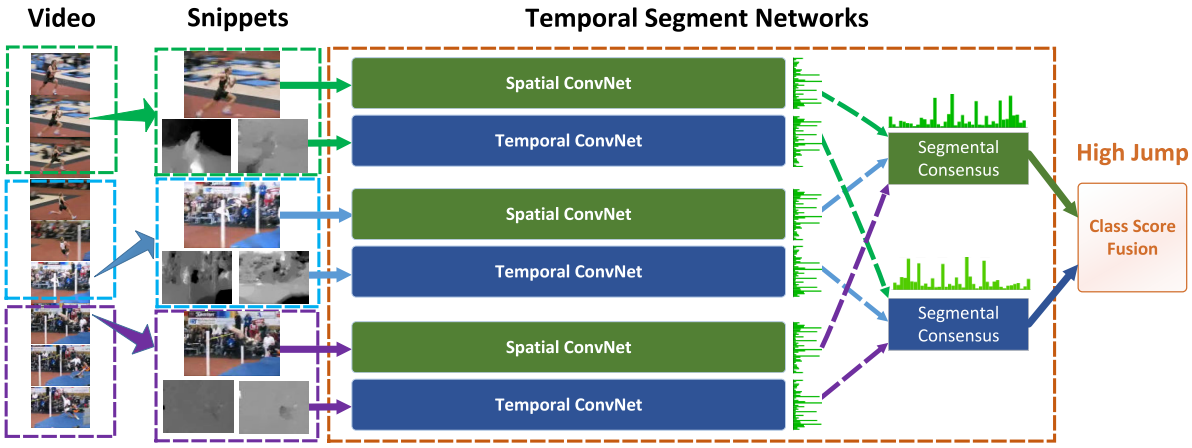

Figure 2.2: *Temporal Segment Network architecture. An input video is divided in K segments with a short snippet sampled from each video segment. A two-stream CNN is used to obtain predictions for each snippet. The predictions from snippets are first fused for each modality, before the final prediction is obtained by fusing the spatial and temporal predictions. Figure taken from [194].*

Towards Longer-Term Temporal Structure. While the work of Simonyan and Zisserman improved the encoding of local motion information, the temporal stream is not capable of capturing long range temporal structure. Wang *et al.* [194] build on the two-stream approach and propose Temporal Segment Networks (TSN) to tackle this problem. The authors note that "consecutive frames are highly redundant" [194] and therefore opt for a sparse temporal sampling strategy. Sparse sampling also alleviates issues with memory and computation limits.

The proposed architecture is depicted in Figure 2.2. Videos are first uniformly divided into K segments and a short snippet is sampled from each segment. Each snippet is passed through a two-stream CNN, with CNNs on the different snippets sharing parameters. A consensus is formed for each stream over the output scores for different snippets. As in the original two-stream CNN, the outputs for each modality are fused by averaging scores.

Wang *et al.* experiment with different segmental consensus functions, such as maximum and weighted average as well as top K pooling and attention weighting in a later extension [196]. However, the authors conclude averaging to be the most effective. The paper also proposes alternate modalities to encode the motion, namely RGB difference and warped optical flow. RGB difference is the pixel wise change between two consecutive frames. Warped optical flow is an adapted version of the dense optical flow previously used in two-stream networks which additionally aims to remove camera motion. While RGB difference and warped flow both have better classification accuracy than the stan-

dard RGB input, they offer little additional benefit when the optical flow modality is included. Using a BN-Inception [65] backbone, TSN obtained a new state-of-the-art for action recognition on both UCF-101 [172] and HMDB-51 [84], outperforming the two-stream approach on the former by 10%.

Although TSN has good performance, it still fails to take into account the evolution of an action. It makes its predictions from additional video snippets over the original two-stream architecture [166], however the network is invariant to the ordering of these snippets as the segmental consensus is formed through averaging. Carreira and Zisserman [18] revisit the idea of 3D CNNs [70, 182] in order to better model temporal structure in action recognition. They propose two-stream inflated 3D ConvNets (I3D) which uses start-of-the-art image classifiers as a starting point by inflating the 2D convolutions and pooling operations into 3D. Carreira and Zisserman argue that this approach is more effective than going through the manual trial and error process of searching for a good 3D architecture. The I3D network consists of an appearance and motion stream, similar to the two-stream architecture [166] and TSN [194]. Each stream takes a stack of 64 frames as input, these are standard RGB frames for the appearance stream and dense optical flow frames for the motion stream. Carreira and Zisserman also propose bootstrapping existing weights from pre-trained ImageNet models in their 3D architecture by making use of 'boring' videos, *i.e.* videos created from a repeated image. Weights of the original 3D filters are repeated across the temporal dimension and rescaled.

The proposed I3D network is compared to several other approaches to temporal modelling, namely previous 3D CNN architectures, the two-stream architecture and adding a Long Short Term Memory network (LSTM) on top the CNN output from multiple consecutive frames. The I3D network obtains a $> 9\%$ improvement over these previous methods on the Kinetics dataset which is also proposed by Carreira and Zisserman in this work. Smaller, but significant, improvement is also obtained on UCF-101 and HMDB-51. The majority of the increase in accuracy comes from improvement on the optical flow stream which the authors attribute to the larger temporal receptive field of 64 frames versus 10 in previous methods.

A large part of the subsequent success of the I3D architecture has been due to the pre-training on the Kinetics dataset [79] which made learning the increased number of weights in a 3D architecture more feasible. The Kinetics dataset consists of 10 second clips of 400 action classes taken from YouTube videos. Each class contains at least 400 clips and the total number of clips is 306,245. The dataset was later extended to contain 600 [19] and more recently 700 [20] action classes. By first pre-training on Kinetics, I3D obtains new state-of-the-art results for action recognition on UCF-101 and HMDB-51,

outperforming TSN. There is a large increase over previous 3D CNN architectures such as C3D [182] despite C3D being trained on the larger Sports1M dataset [70]. This is likely due to a combination of factors. I3D makes use of more recent advances in image recognition CNN architectures, thus it is a much deeper architecture than C3D but has fewer parameters. The Kinetics dataset also contains a larger array of different actions therefore learnt weights are likely more transferable to UCF-101 and HMDB-51.

The standard pipeline for video understanding on short clips has undergone a large number of changes in recent years. CNNs have now replaced the previous separate processes for identification of regions of interest, feature extraction, feature aggregation and classification. The success of CNNs was largely due to the introduction of new larger scale datasets for action recognition such as Sports1M, UCF-101, HMDB-51 and Kinetics. These datasets have allowed exploration of how to best encode temporal information to gain understanding of videos beyond the individual frames they contain. However, the methods examined in this section are limited in that they only aim to encode short-range temporal information as this is what the problem of action recognition requires. Recent works utilise temporal modelling [94, 216], multi-stream 3D convolutions [43], joint encoding of spatiotemporal and motion features [71] and more efficient 3D convolutions [183], however these works also only consider short-range temporal information.

2.1.2 Long and Untrimmed Videos

As Kaparthy *et al.* note "unlike images which can be cropped and re-scaled to a fixed size, videos vary widely in temporal extent and cannot be easily processed with a fixed-sized architecture" [78]. The works described in the previous section all focus on video understanding in trimmed temporal clips of several seconds, thus they avoid this issue. In many works, videos are treated as fixed sized clips and are short enough that the proposed temporal architectures can operate over the entire video. Alternatively, many methods rely on all the contents of the video being relevant to the ground-truth action and thus form the final prediction from average pooling of sparsely sampled frames or snippets.

Average Pooling. Ng *et al.* [210] study how existing temporal architectures can be extended to classification in long videos. They test single frame methods against a consensus formed from average pooling or an LSTM. These methods are tested on the Sports1M [78] and UCF-101 [172] datasets. While UCF-101 is a trimmed action recognition dataset containing clips of 10-15 seconds, the videos in Sports1M are much longer and more varied. The first 2 minutes of videos in the Sports1M are taken for use in the

models, with 120 frames sampled at 1 frame per second. Ng *et al.* find that while both the LSTM and average pooling are better than using a single frame, the average pooling significantly outperforms the LSTM by 4% on Sports1M. They also find that using the full 120 frames sampled per video gives a further increase over performance in contrast to sampling only 30 frames. The success of average pooling is likely due to two factors. First, while the videos in Sports1M are long and may contain some irrelevant frames, only 5% of videos are labelled with another confounding action. Therefore, sampling of more frames increases the likelihood of obtaining more useful information about the action of interest. Second, LSTMs have been shown to struggle with longer sequences [165, 185] and are somewhat difficult to train [83, 136, 165]. While LSTMs are able to remember information over more steps than RNNs and thus solve some of the vanishing gradient problems of RNNs, they still struggle with longer term information. Therefore, LSTMs show inferior performance compared to non-recurrent networks in many sequence based tasks [18, 50, 185].

Finding relevant events in video. In both short and long videos, it may be the case that parts of the video are more relevant to the ground-truth label or temporally trimmed videos may not be available in training or testing. Therefore, it can be desirable to determine the most relevant temporal parts of the video for the task and limit the prediction to only use these parts. While the relevance to the task can also be an issue spatially in videos [34, 92, 97, 104, 171], the rest of this section focuses on temporal relevance and attention as these works are more pertinent to this thesis.

One way of selecting relevant parts of a video is by learning a sampling distribution. Piergiovanni *et al.* [146] do this for action recognition. They pose that activities are often composed of multiple temporal parts or sub-events and propose learning gaussian filters to identify these sub-events. Each gaussian filter is represented by a centre, duration and resolution parameter. The centre and duration represent the shape of a single gaussian distribution, while the duration indicates how often this distribution is repeated through the video. The gaussian filters are learnt using pre-extracted CNN features from frames of the input videos. Pre-extracted features are often used in these scenarios as it makes the training process quicker and prevents over-fitting on a small dataset.

The features are pooled using the value of the gaussian distribution as the weighting and then concatenated with the output of other filters. In training, these gaussian filters are static and shared across all videos. In testing, the parameters of these filters can be fine-tuned with LSTMs for individual test videos. Piergiovanni *et al.* first try sharing the temporal filters across all activities, however this approach gave little benefit over using predetermined filters in a pyramid structure [155]. Instead, temporal filters are

learnt per-activity. This gives a significant increase over max-pooling the pre-extracted CNN features, around 10% on HMDB-51. Piergiovanni *et al.* also test the impact of extracting features from different pre-trained models: VGG [167], IDT [189], C3D [182] and TDD [192]. The features used have a large impact on the final result, with TDD performing 11% better than C3D, although the introduction of the learnt temporal features show a similar increase on both feature types.

Moltisanti *et al.* [119] also use distributions to determine the most relevant frames for action recognition. Instead of using temporally trimmed clips, as is common in action recognition, Moltisanti *et al.* learn from untrimmed videos where actions are weakly labelled with single temporal points and may overlap. To learn a representation for these actions, each action is represented with a plateau function around the annotated temporal point. Frames are then sampled from within these plateaus. The parameters of the plateau functions are periodically updated based on the softmax scores of the relevant action classes. With this method Moltisanti *et al.* are able to close the gap between this type of weak-supervision with temporal points and full temporal boundary supervision with a performance difference of only 1% on THUMOS. The gap is however much larger on EPIC-KITCHENS [30] where there are many more action classes per video: 26% classification accuracy with temporal points versus 36% with full supervision.

Temporal Attention. Using gaussian or plateau functions as a way of sampling frames assumes that consecutive frames within an action or sub-event are all relevant. Instead, there may be specific parts of an action, which are the most informative. An alternative approach is to learn a function which directly predicts the usefulness of individual video frames or snippets to the target problem. This is known as *attention*. While these attention values could be supervised directly, this requires further training annotations which are difficult to gather. Instead, attention values can be learnt with video-level supervision rather than frame-level supervision.

Attention can be either hard or soft. In hard attention [6, 118], regions are selected with hard binary choices — video frames can either be relevant to the task or can be irrelevant and ignored. Hard attention methods face difficulty in training as they are not differentiable. In soft attention, weighted averages are used instead of hard selections. An overall video-level feature can be obtained by using the predicted attention values to weight video segments when pooling.

Long *et al.* [101] use a soft attention to classify actions. Unlike the method of Pirgiovanni *et al.* [146], attention values are computed directly from input features instead of using a semi-static distribution. Assuming a video of length T is represented by succes-

sive frame or clip-wise features $X = (x_1, x_2, ..., x_T)$. This attention value is computed with a non-linear function represented by two fully-connected layers:

$$a = \text{softmax}(w_1 \tanh(W_2 X + b_1) + b_2) \tag{2.1}$$

This attention is then used to combine features into a single overall video feature with weighted average pooling:

$$x = aX \tag{2.2}$$

Long *et al.* note that "normally, a single attention unit can only be expected to reflect one aspect of the video. However, there can be multiple pertinent parts in a video that together describe the overall event" [101]. Therefore, they introduce multiple attention filters, the outputs of which are concatenated to construct an 'attention cluster'. An attention cluster is used for each modality of the video: spatial, temporal and audio. One difficulty with using multiple attention filters is that they may not naturally focus on diverse aspects of the video and instead may attend to the same most informative video parts, ignoring other complementary parts. To combat this, Long *et al.* include a shifting operation with additional learnt linear parameters for each attention filter which causes each filter to give a different distribution. This method is tested on the UCF-101, HMDB-51 and Kinetics datasets. While the method did achieve state-of-the-art results, its improvements over action recognition methods without attention, such as TSN [194], are marginal.

Attention mechanisms become much more useful in longer videos, where frames are more likely to be irrelevant to the ground-truth label. Pei *et al.* [139] estimate attention values with a bidirectional RNN which takes into account context from surrounding frames. This attention value is then incorporated into a recurrent network as the gating mechanism. Using this attention mechanism as a scalar gate makes the model easier to train than Gated Recurrent Units (GRUs) or LSTMs which typically have 2 and 3 vectorial gates respectively. Pei *et al.* demonstrate this by comparing to other recurrent methods on three types of sequential data: audio, text and video. In video they test on the CCV dataset [72], which contains YouTube videos of an average duration of 80 seconds. While this method does outperform a plain RNN, GRU and LSTM, it is not compared to other attention methods and likely suffers from some of the same issues with recurrent networks described previously.

Weakly-Supervised Action Localisation. Wang *et al.* [195] proposed the problem of weakly-supervised action localisation. In contrast with action recognition, the goal of action localisation is to detect an action by identifying its start and end time in addition to its class. This is usually trained with ground-truth temporal annotations indicating the start and end time of actions. In weakly-supervised action localisation these temporal bounds are not available, instead the goal is to learn from only video-level labels indicating the action(s) present. Two datasets are commonly used for this problem: THUMOS [73] and ActivityNet [15]. The THUMOS training set is based on UCF-101 with additional untrimmed videos from 20 classes for validation and testing. ActivityNet consists over 20,000 videos from 200 classes. Videos are typically between 5-10 minutes long with an average of 1.5 action instances per video and therefore much longer and sparser than datasets typically used for action recognition.

To tackle the problem of weakly-supervised action localisation, Wang *et al.* propose UntrimmedNet [195]. Clip proposals are first generated from the untrimmed videos, either by uniformly splitting the videos or using HoG features to detect visual changes between adjacent frames. These clips are then passed through a classification network. To select the clips most likely to contain an action, Wang *et al.* propose two approaches. The first is a hard selection approach which uses multiple instance learning (MIL) on the classification scores. Instead of having individually labelled instances, MIL assumes a bag of instances with an overall label for the bag which may differ between instances within the bag. In negatively labelled bags all instances are negatively labelled, but in positively labelled bags one or more instances will correspond to the label, with the rest being negatives. In weakly-supervised action localisation, the bags are videos and instances are video clips or frames. With this formulation, the k proposals with the highest classification scores per action are used to train the model. Wang *et al.* also propose a method of soft selection, similar to the softmax attention filter described above. The soft and hard selection approaches have a similar performance to each other, although both outperform previous methods for action recognition and obtain good performance for weakly-supervised action localisation, comparable with some fully-supervised methods.

Nguyen *et al.* [125] propose the Sparse Temporal Pooling Network (STPN) for weakly-supervised action localisation which does not first require generating a set of clip proposals. Instead, they use a soft attention directly on extracted I3D features from which proposals are later selected. This attention is similar to the filter used by Long *et al.* described above, but instead uses a sigmoid activiation function to detect multiple occurrences of the same action. As only parts of the video will contain the action of interest,

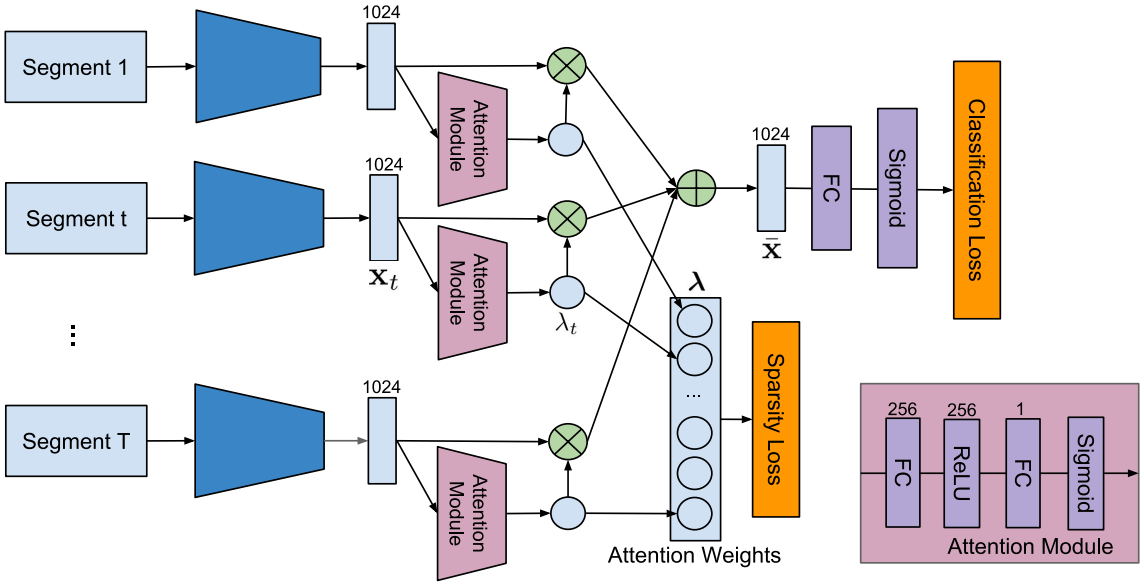

Figure 2.3: *Sparse Temporal Pooling Network. Input videos are first uniformly split into T segments with an I3D feature extracted for each segment. The attention module consists of two fully connected layers and computes an attention weight for each segment. These attention weights are used to weight the corresponding segments in the temporal pooling and obtain an overall video-level feature. The network is trained with a video-level classification loss and an L^1 loss to enforce sparsity on the attention weights. Figure taken from [125].*

Nguyen *et al.* optimise the attention with a L^1 Norm sparsity loss as well as the video-level classification loss. This is a class agnostic attention as the same attention filter is used for videos of all classes, therefore it aims to separate segments of the video containing actions from other background segments. To better localise actions of particular classes, Nguyen *et al.* propose using temporal class activation maps. These are 1D maps in the temporal domain taken from the classification module. By combining these maps with the class agnostic attention, the sparse temporal pooling network is able to improve on the prior results from UntrimmedNet [195] for both ActivityNet and THUMOS.

Paul *et al.* [138] follow the approach of Wang *et al.* and formulate weakly-supervised action localisation as an MIL problem. They propose a MIL loss which uses the average of the max k activations for each class within a softmax cross-entropy classification loss. In their proposed method, W-TALC, the authors also propose a co-activity similarity loss. This loss enforces the idea that in a pair of videos with the same class, temporal regions identified that class should have similar features to each other. In turn, these features should be different to those from other parts of the pair of videos identified as other classes. This is estimated using the class softmax scores and enforced with a ranking hinge loss. The authors find the co-activity loss to give the best improvement:

without this the performance drops 7-8%. Although the addition of the MIL loss does also give a boost to performance.

Liu *et al.* [96][1] note that while one video segment may be needed to recognise an action, action localisation requires identifying all the segments relevant to the class of interest. Therefore, the authors propose to model the completeness of the action while also separating actions from background context which can be visually similar. Like STPN [96], the proposed completeness modeling and context separation (CMCS) method also uses a combination of class agnostic attention and class activation sequences. Class activation sequences are produced with a temporal convolution. CMCS contains multiple branches of class activations in order to identify the full extent of an action. These are optimised with a diversity loss to encourage different branches to focus on separate parts of an action. To better separate actions from their surrounding context, Liu *et al.* identify clips within the training videos which contain little motion. These are used in training a pseudo background class. This method gives a marginal increase in performance over the W-TALC method proposed by Paul *et al.* [138]. Each component gives a small but complementary improvement, with the inclusion of multiple class-specific attention branches and the use of the pseudo background class having the most effect.

In summary, it is often necessary to identify the most useful temporal parts of a video for the target problem, particularly when dealing with long or untrimmed videos. Soft attention is an effective way to learn this relevancy, especially when videos are minutes in length such as in weakly-supervised action localisation. While significant progress has been made in this area, the commonly used datasets THUMOS and ActivityNet have some limitations, namely videos typically do not contain multiple different types of actions. Instead of distinguishing between multiple confounding actions, the main task is to distinguish between the action of interest and background segments where no action is happening. Chapter 4 and Chapter 6 explore these challenges further, with Chapter 4 focussing on temporal attention for skill determination and Chapter 6 aiming to identity segments relevant to a given action within instructional videos.

2.1.3 Instructional Videos

Instructional videos, also known as 'how-to' videos, demonstrate how one should complete a particular task. As well as including a visual demonstration of the task, instructional videos are usually narrated by the instructor to explain what is happening and

[1] Concurrent to the work presented in Chapter 4, also published in CVPR 2019.

to draw attention to critical elements of the task. Millions[2] of people watch narrated instructional videos to learn how to cook new dishes, assemble flatpack furniture or even perform car maintenance tasks. These videos are typically minutes in length, similar to the videos used for action localisation tasks. However, different to those videos, instructional videos contain many different actions occuring in quick succession.

Instructional videos are also different in that the accompanying narrations can act as a form of free supervision. These narrations are transcribed into text for subtitles either manually or automatically. While the narrations, and corresponding subtitles, are only roughly aligned with the video and may contain tangents irrelevant to the task, these narrations contain information which explains what the instructor is doing in the video. Therefore, many works have aimed to learn from instructional videos datasets [2, 110, 160, 217, 218] with the weak supervision provided by the narrations.

Movies are another popular source of video dataset with free weak-supervision [36, 37, 88, 176]. In movies, this supervision comes from either the subtitles of the actors speech or from the script, which contains stage directions in addition to a transcript of the speech. However, movies typically focus on talking heads with few object interactions. Instructional videos focus on object interactions as opposed to person interactions and as their purpose is to explain how to perform a task well, they are inherently more related to determining skill.

Step Localization. One of the most well studied problems in instructional videos is localisation of the key steps of a task [2, 61, 105, 151, 161, 218]. This has similarities to the action localisation problem covered in the previous section as the goal is to learn a representation to find the temporal bounds of particular steps or actions. However, step localisation in instructional videos makes use of the steps being shared between multiple videos of the task and often assumes there is a list of steps consistent throughout all videos of a task with a set order. In reality there is some variance between videos, with some steps being performed in a different order or omitted entirely in some videos.

Alayrac *et al.* [2] were the first to study step localisation in instructional videos while only using supervision from the accompanying narrations. They present an instructional video dataset of five tasks: making a coffee, changing a car tire, jumping a car, performing cardiopulmonary resuscitation and repotting a plant. The dataset contains 30 videos of each task. To discover and localise the steps within a task, Alayrac *et al.* take all the videos of a single task and perform a two-stage clustering process where the narrations

[2]Many instructional videos on YouTube have more than 10 million views, for example www.youtube.com/watch?v=s9r-CxnCXkg

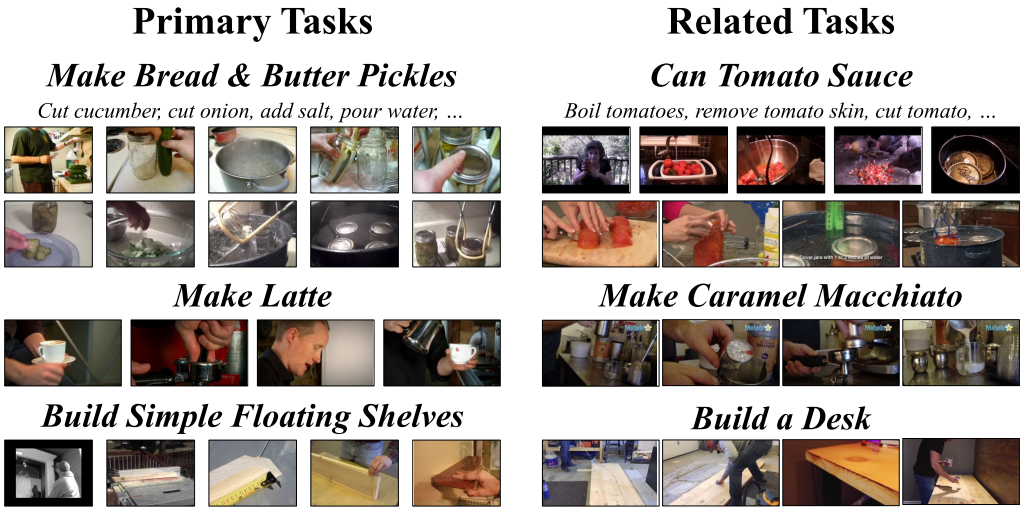

Figure 2.4: *The CrossTask dataset consists of instructional videos of 'primary' and 'related' tasks. Related tasks are distinct from primary tasks, but may share some steps or particular verbs and nouns within those steps. For instance, 'cut' is present in both 'can tomato sauce' and 'make bread & butter pickles'. Figure from [218].*

of each video are first clustered, followed by the videos. The text is clustered first as the authors assume variability in language is easier to capture than visual variability. To cluster the text, direct object relations between verbs and nouns are discovered to better focus on the relevant portions of the narrations. The order of steps discovered through the text clustering is then enforced when clustering video snippets. This method is able to recover similar verb-object descriptions to the ground-truth steps in the five tasks, albeit often with some additional stages. The approach also performs step localisation well, although it is far below a fully-supervised approach. It struggles with the repetitive steps in jump car and the variability of the narrations in make coffee.

Zhukov *et al.* [218] aim to use similarities between related tasks to better localise steps within instructional videos. For this purpose they propose the CrossTask dataset which contains 4.7K instructional videos over 83 different tasks within cooking, car repair and DIY. Of these tasks, 18 are *primary* tasks, each with 80 videos, and the other 65 are *related* tasks with 30 video per task. These *related* tasks come from similar domains to a primary task and will likely share some actions and objects with the primary tasks (see Figure 2.4). For instance, 'make latte' is a primary task, while 'make caramel macchiato' (a different type of coffee) is a related task. The primary tasks are annotated with temporal localisations of the steps within the task. For each task an ordered list of steps is provided, this is obtained from the WikiHow [1] description of the task.

Zhukov *et al.* aim to make use of objects and actions being common across tasks and learn

a component model shared between these tasks. Steps can be classified and localised with a single component model instead of learning the steps for different tasks separately as was done by Alayrac *et al.* [2] and many others [61, 105, 151, 161]. To learn to locate the steps within a video, a classifier is learnt for each action and object, *i.e.* there is a classifier for 'pour' and a classifier for 'milk' which can then be combined to identify the step 'pour milk'. These classifiers are optimised alternately with the localisation module to learn a representation of the step components within a task. The localisation module learns a binary label matrix indicating the assignment of video snippets to the known ordered list of steps, which is optimised in combination with several constraints: the temporal ordering of steps, each step occurring at least once and that the step should appear close to where it occurs in text. The use of the component modules allows the method to be applied to previously unseen tasks, as classifiers for new steps can be built from the learnt components. For instance, the classifier for 'pour water' in 'make a latte' can be built from the components of 'pour milk' in 'make pancakes' and 'boil water' in 'cook Brazilian rice'. Zhukov *et al.* find this sharing of components between the primary and related tasks gives an increase of 4%.

Instructional Videos for Pre-training. Miech *et al.* [110] present the large-scale HowTo100M dataset which contains 136 million video clips from 1.22 million narrated instructional videos. To collect this dataset Miech *et al.* first acquired a list of 'how-to' tasks from WikiHow. After filtering out tasks containing non-physical actions, such as 'feel', and abstract categories, such as relationship advice, 23,611 visual tasks remained. Videos are collected by querying YouTube for these tasks and selecting popular videos with English subtitles. It is worth noting that the tasks and videos in the CrossTask dataset are a subset of the HowTo100M dataset.

Videos are split into clips based on the duration of each subtitle. The labelling is only weakly-supervised as the narrations are only roughly aligned with the contents of the video. In addition, while the narrations will often describe what is happening, they will also contain some irrelevant information such as title credits, anecdotes and information about alternate ways the task could be done. This means the narrations are quite noisy when used as labels. The authors note that only in 51% of cases one or more of the narrated actions or objects are present within the corresponding video clip.

Despite the noise and weak supervision, HowTo100M is a useful dataset due to its size and diversity, particularly for pre-training video. Miech *et al.* learn a video-text embedding space with these noisy video-caption pairs. This model is then applied to several downstream tasks, such as step localisation on the CrossTask dataset [218]. Interestingly, training for video retrieval on HowTo100M without any fine-tuning beats the perfor-

mance of Zhoukov *et al.* [218] on CrossTask. Even with more dissimilar target data, HowTo100M is still beneficial, with pre-training on HowTo100M improving performance for clip retrieval on YouCook2 [217] and MSR-VTT [204].

Other Uses of Instructional Videos. Instructional videos have also been used for variety of other tasks. For instance, Alayrac *et al.* [3] aim to discover the various states an object can have and which actions cause changes between these states. They cluster objects with a similar appearance into multiple different states. This clustering is done under the constraints that only one object can be manipulated at a time and that, within a video, every occurrence of the first state is before any occurrence of the second, transformed state. Alayrac *et al.* then find actions which occur during the transition between discovered object states. While the majority of this method is unsupervised and does not use supervision from the instructional videos' narrations, the authors do extend to retrieving clips via text. In their main evaluation setup, clips of the instructional videos have been selected such that they contain a single object state transformation (as well as other irrelevant segments). The authors extend the discovery of such clips to a text retrieval problem where an SVM classifier is used to locate a given action. The performance of the object state discovery and action localisation is much lower on these automatically collected clips than the curated collection, however the proposed model still outperforms the baselines by a significant margin.

Related to object state discovery is the task of procedure planning. Chang *et al.* [22] predict the actions needed to obtain the result in a goal image from a given starting image. To do this they propose two modules, one which predicts the next action based on the current state and previous action and another which predicts the next state given the current state and next action. These models can be combined and iterated between to predict the path of actions and states from the start to the goal. The two models are trained with information about the intermediate actions and states between start and goal and are applied to new videos of previously seen tasks. The authors demonstrate that the method does condition its output on the goal and starting images as opposed to learning a preset set of steps necessary to complete a task. The method is also applied to the problem of walkthrough planning, where the visual output after each action is shown to demonstrate the path through a task.

Another task made possible by the type of weak supervision available with instructional videos is visual grounding [63] and reference resolution [62]. Narrations in instructional videos often refer to objects as 'it' or with other pronouns or ambiguous descriptions. Humans watching the instructional videos can easily use visual context and the information from previous narrations to determine what 'it' refers to. Reference resolution

aims to determine which object these unspecific words refer to, while visual grounding additionally aims to localise the object within the video. Huang *et al.* [63] create a graph between detected object bounding boxes and text from the video's narrations, with additional edges between different phrases in the text so linguistic reference resolution can be performed. The matching between text and bounding boxes within this graph is formulated as a MIL problem where at least one of the detected boxes should correspond to the text. Since the narrations of instructional videos often correspond to the actions taking place in the video, words are given a higher probability of being grounded to a box temporally close to when it is mentioned. The reference aware MIL loss also takes into account partial correctness by penalising an incorrect object more than the correct object grounded in an earlier part of the video.

The accompanying text of the narrations contained within instructional videos offers a type of weak supervision for many tasks within video understanding. These narrations come for free with instructional videos, however they are weak and noisy as the narrations are only roughly aligned with the videos and will often contain irrelevant discussion. This makes instructional videos a convenient but challenging source of data. Instructional video datasets are also particularly useful as they contain many videos of the same task, therefore they can be used for higher-level understanding of a task's structure as well as lower level understanding of the individual actions and steps. Much of the existing work in instructional videos has focused on *what* is happening in the videos. This thesis instead looks at differences between *how* and *how-well* people perform a task. Chapter 6 will explore how particular key steps are performed and will learn this from instructional videos.

2.2 Ranking and Retrieval

This section will cover work related to this thesis which aims to learn to order items. In some cases this ordering can be performed by determining a item's relevance to a given query, where the relevance of a single item will differ between various queries. For instance, when using a search engine on the web, items are ordered according to their relevance to the search term. In other cases the ranking is independent of a query, but dependent on a given criteria. For instance, when ranking items in terms of quality rather than relevance. In this thesis, the latter is referred to as ranking and the former is referred to as retrieval. Section 2.2.1 will cover the various approaches of learning to rank and the common architectures used to do this. Section 2.2.2 will examine retrieval in computer vision, with a focus on cross-modal retrieval between video and text.

2.2.1 Ranking

When learning to rank, the general aim is to learn a function which orders the items in accordance with their ground-truth ranking. This ground-truth ranking could be a complete ranking, where all items are ordered relative to each other in a single list. Alternatively, it could be a set of partial rankings, where the ordering of an item may only be known in relation to a subset of other items. In computer vision the items ordered are typically images or videos. There are three common approaches taken when learning to rank: pointwise, pairwise and listwise.

Pointwise approaches tend to be used when a ground-truth score is available. Methods can learn to regress to this score from the input image or video, or classify the correct score if it is categorical. Rankings can then be obtained by ordering items in accordance with their predicted score. Therefore the goal is to lean a model $f(\cdot)$ to predict the correct score y from data item x:

$$f(x) = y \tag{2.3}$$

A simple way to learn pointwise ranking would be to evaluate the difference between the predicted score and the ground-truth with Mean Squared Error or some other appropriate distance function. Crammer and Singer [25] observe that it can be unhelpful to update a model on all mistakes, even when the predicted score is close to the ground-truth. Instead they propose to allow intervals in regressing to pointwise scores and only update when the predicted score is outside of an acceptable margin.

While problems with ground-truth categorical scores can be modelled as classification problems, this is somewhat counterintuitive as it ignores the relationship between increasing or decreasing scores. However, Li *et al.* [90] propose to solve this issue by instead learning class probabilities with a soft classification. The final scoring function is obtained by combining class probabilities with a monotonically increasing function.

Pointwise ranking offers the benefit that it can take into account how close items' ground-truth scores should be, however these ground-truth scores are often not available.

Pairwise ranking is useful when ground-truth scores are not available. Instead preferences between pairs of items are used to learn an overall ranking function. The can be done by learning a model $f(\cdot)$ which assigns the higher ranked item x_i in a pair (x_i, x_j) a higher score:

$$f(x_i) > f(x_j) \tag{2.4}$$

Herbrich *et al.* [60] were the first to train a ranking model with pairs. Instead of using the traditional regression approach, they modelled the ranking problem as an ordinal regression. This formulation only assumes that there exists an ordering among the items, meaning each has a rank which can be regressed to instead of a ground-truth score. With this pairwise training, the goal is to classify whether a pair is correctly ordered according to this ranking. Herbrich *et al.* therefore adapt an SVM designed for classification to classify whether a pair of items is correctly ranked. The proposed RankSVM also takes into account the target ranks, so while the inclusion of the pairwise preferences does improve results over classification and metric regression methods it would still be unsuitable if there are only partial ranks as opposed to a fully known rank containing all items.

Joachims [75] extends the RankSVM to remove the need for a full ranked list. This version of the RankSVM became more commonly used for computer vision problems. For instance, Parikh and Grauman [131] use RankSVM with precomputed image features to order the strength of particular attributes, such as smiling. They learn a ranking function per attribute with only a partial ordering of images based on relevant attributes.

Burges *et al.* were the first to bring pairwise ranking into a deep learning framework with RankNet [14]. The goal in RankNet is only to minimise the number of inverted pairs, meaning this approach can cope with partial rankings unlike [60]. To optimise the ordering of pairs, a binary cross entropy loss is adapted to classify whether a pair is correctly ordered. This was originally applied to the problem of ranking internet search queries, but was again adapted to computer vision domain for ranking relative attributes [173]. In this work, Souri *et al.* find RankNet to be much more successful than RankSVM, with around 20% improvement on facial attribute dataset LFW-10 [64]. This demonstrates the importance of backpropagating through the network to learn fine-grained features. It is possible to incorporate similarly ranked pairs when training RankNet, although the authors did not find the inclusion of similar pairs helped. This was confirmed by similar findings from Sculley [159] when using a hinge-based loss.

Pairwise ranking generally obtains competitive results with the pointwise formulation while not requiring ground-truth scores or ranks. More recent pairwise ranking methods also offer the advantage that a complete ranking is not necessary in training and the pairs can instead form several partial rankings. This makes pairwise ranking a valid option for many different types of tasks.

Listwise ranking is another alternative formulation used to learn a ranking function. Instead of optimising the ranking locally with orderings of individual ranked pairs, the

listwise approach uses a global view of the ranking and aims to minimise the loss of the entire ranked list.

Cao *et al.* take this approach with ListNet [16]. By transforming the predicted scores and ground-truth judgements into probability distributions, they are able to measure the difference between the predicted and the ground-truth lists and back-propagate the error through the network. The authors find this listwise approach outperforms pairwise approaches, such as RankSVM and RankNet, and argue that this is due to large numbers of pairs having a costly training procedure and often not being relevant to the loss.

For many ranking applications, the evaluation metric will often evaluate the correctness of the full permutation of the list, for instance Spearman's rank correlation or normalised discounted cumulative gain (NDCG). The listwise approach offers the advantage that these evaluation metrics can be optimised for directly, although some approximation is necessary to make these metrics differentiable. Taylor *et al.* [178] approximate NDCG by computing the expected value from a distribution over ranks. This approach is able to outperform regression methods, such as mean squared error, and pairwise methods, such as RankNet, on several web search corpora.

Although listwise approaches have promising results, the complexity of the objective function can make these methods difficult to train [98]. The annotations required for listwise approaches can also be harder to obtain than pairwise annotations.

Ranking of Images and Videos. The pairwise learning to rank approach is the one usually adopted for use in computer vision due to its flexibility with annotations. To learn from pairwise annotations with deep networks, a Siamese framework is used. An example Siamese network used by Souri *et al.* [173] to rank relative attributes is depicted in Figure 2.5. It consists of two identical sub-networks which are fed into a single loss function. The sub-networks each output a predicted relative score for the corresponding item and the loss function evaluates whether these predictions order the items in a pair correctly. The input is a pair of images or videos with a label to indicate their correct ordering. The two networks share the same network weights and are updated using the sum of the gradients from the two sub-networks. At testing time a single sub-network can be used to predict a relative score for each video as shown in Figure 2.5.

An example of the Siamese architecture being adopted for ranking videos is the work of Yao *et al.* [208], who use ranking to perform highlight detection. Instead of trying to sample highlights from long videos, Yao *et al.* split each video into clips and rank the clips based on how much of a 'highlight' they are. The 'highlightness' ranking is then used to either summarise the videos with a time-lapse, where clips with lower highlight

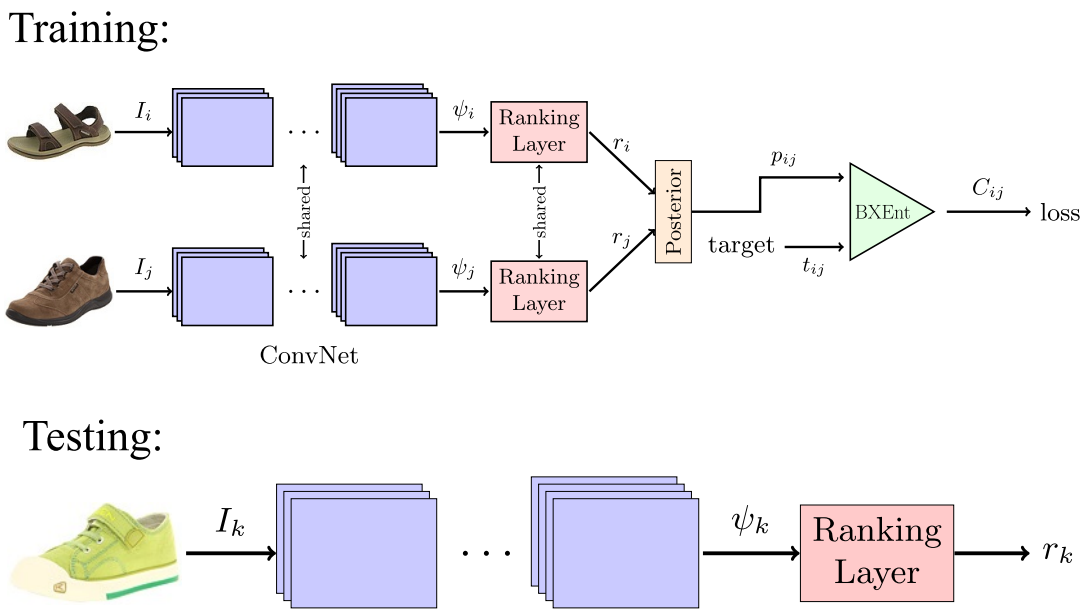

Figure 2.5: *This Siamese ranking network is used by Souri et al. to rank relative attributes in images. Each image in a pair is passed through an identical sub-network, with weights shared across the two networks. Each network outputs a score from the ranking layer. The ordering of the image via this score can be evaluated by the chosen loss. In testing, only a single sub-network is needed to obtain a relative score per image. Figure taken from [173].*

values are played faster, or with skimming, where a set time budget of clips are selected by their highlight score. To perform the ranking, Yao *et al.* use a Siamese architecture as pictured in Figure 2.6. The Siamese architecture consists of both an appearance network which takes RGB frames and a motion network which takes a stack of optical flow frames. Features from these inputs throughout the duration of the clip are then averaged to gain an overall clip level feature. The Siamese network weights are learnt with the following margin ranking loss function:

$$\sum_{(x_i,x_j)\in P} \max(0, 1 - f(x_i) + f(x_j)) \tag{2.5}$$

where P is the set of ranked pairs and $f(\cdot)$ is the neural network.

Ranking has also been used for image quality assessment. Liu *et al.* [99] aim to reduce the need for ground-truth image quality scores in this problem, and thus propose a solution which uses images automatically worsened in quality by adding blur and compression artifacts. While these automatically generated images do not have ground-truth scores, they are generated such that they will be worse in quality than the original image. Liu *et al.* therefore train for image quality assessment with a ranking model. This can then

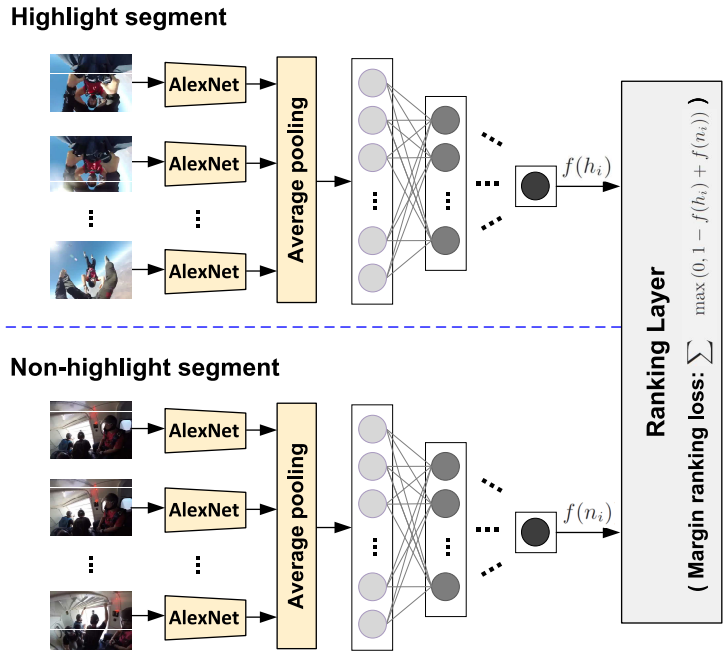

Figure 2.6: *The Siamese architecture used to train a pairwise deep ranking model for highlight detection. Features are first extracted from video segments which are average pooled to obtain an overall video-level feature. This is done for both the highlight clips and the non-highlight clips. Video-level features are passed through a Siamese network with shared weights which outputs a score of 'highlightness' for each video. The weights in the Siamese network are learnt with a margin ranking loss. Figure from [208].*

be fine-tuned with a small number of image quality assessment scores to calibrate the ranking to the desired output scores.

Both highlight detection and image quality assessment are not traditionally formulated as ranking problems, instead these methods investigate ranking as an alternative approach where annotations may be easier to obtain. As mentioned previously, ranking relative attributes is the most common ranking problem in computer vision.

Singh and Lee [168] recognise that not all of the ranked images are relevant to the attribute of interest. Thus they extend the previous work [173] on ranking of relative attributes to incorporate a spatial transformer [67]. With this, they are able to focus on a smaller more relevant region of the image when performing the ranking. This approach sees improvement in the majority of facial attributes tested, particularly attributes such as dark hair and open mouth which will naturally only be recognisable from a certain portion of the face. The method shows less improvement on UT-Zap50K [209] which contains product images of shoes, likely as there is little confounding information in product images so the network is already focussing on features from the relevant regions.

In conclusion, ranking is often adopted as an approach when scores are difficult to obtain. Due to the ease in collecting annotations and the popularity of the Siamese architecture, pairwise ranking is the most commonly used framework when learning to rank. This approach has been used for a variety of different tasks in computer vision, including ranking of video snippets, however far fewer works study ranking problems than classification or regression.

2.2.2 Retrieval

Retrieval is similar to ranking in that the goal is still to order items, however instead of learning a static ranking function, the aim is to order items by their relevance to a given query. Another key difference with retrieval is that often only the ordering of the most relevant items are used to evaluate these methods and lower ranked items are ignored. This is because in retrieval applications the top-ranked items are the ones which will be seen; it is unlikely users would actually look past the top N results.

In computer vision, retrieval began as an image-to-image task where the goal is to find images which depict the same object. A commonly studied image-to-image application is place recognition. Other image retrieval methods have looked at finer-grained similarities where success is not as clear cut as the image containing the same object instance.

With the increased use of sites such as Google Images or YouTube, cross-modal retrieval between vision and language became more common. The goal is to retrieve images or videos relevant to the text query, although methods often also work for the reverse problem of retrieving text relevant to an image or video. Much of the focus of this problem is in finding the most relevant parts of the text query. Some retrieval methods take into account the full query sentence, potentially with an estimation of the relevant words, while others know the most relevant types of words for their application in advance.

Video retrieval introduces an additional problem of which video parts are the ones relevant to the query text. As with other video understanding problems (see Section 2.1) temporally trimmed videos are not always available or applicable to the target problem. The issue of temporal relevancy differs in a retrieval setting as it is no longer possible to learn attention based on a video's class as in [96, 125, 138, 146], instead it is necessary to use the text to inform which parts of a video are relevant.

This section first discusses image-to-image retrieval and its relation to ranking in Section 2.2.2.1, before reviewing cross-modal image-text retrieval in Section 2.2.2.2. Section 2.2.2.3 then reviews video retrieval works relevant to this thesis.

2.2.2.1 Image-to-Image Retrieval

Early works in image retrieval focused on retrieval of the same object instance [127, 141] or, in a larger scale, landmark retrieval [7, 68, 144, 145]. Given a query image depicting a certain building or place, the goal is to determine the location by retrieving other images containing the same building. Methods which tackle this problem have to be robust to changes in viewpoint, illumination and weather. Many works in this area have focused on improving the hand-crafted features used to encode the images as it is important to extract key information which ignores confounding objects, such as people or cars, and is robust to changes in viewpoint, illumination and season.

Instead of performing learning on top of hand-engineered image descriptors, Arandjelovic *et al.* [5] aim to learn a retrieval function in an end-to-end manner. They propose NetVLAD which makes vectors of locally aggregated descriptors (VLAD) [69] trainable and therefore possible to incorporate into a CNN. By learning the important parts of the image for place recognition, their method is able to focus on elements like building facade and skyline which are similar to the query image while ignoring distracting elements such as cars and people.

Works in place recognition typically focus on instance retrieval where items are deemed relevant if they contain the same exact object or landmark. Other works have focussed on category based retrieval, where relevance is defined by images containing the same type of object [56, 177]. Wang *et al.* [190] instead aim to learn fine-grained image similarity where the similarity between items of the same category can differ. They do this in a deep learning framework with an architecture inspired by the pairwise ranking approach. As shown in Figure 2.7 the architecture is similar to a Siamese network, but with three streams instead of two. Each training sample is a triplet, consisting of a query image (x_i), a positive image (x_i^+) and a negative image (x_i^-). The goal is to learn space where the positive image is closer to the query than the negative image, *i.e.*

$$d(f(x_i), f(x_i^+)) < d(f(x_i), f(x_i^-)) \tag{2.6}$$

for all triplets (x_i, x_i^+, x_i^-) where x_i is the query image, x_i^+ and x_i^- are the positive and negative images respectively and d is a distance metric. This goal is very similar to the pairwise ranking objective (Equation 2.4), where the distance between pairs of images is considered instead of the scores of individual images. Therefore, the hinge-based loss used for pairwise ranking can also be used for learning retrieval with triplets:

$$L = \sum_{(x_i, x_i^+, x_i^-)} max(0, d(f(x_i), f(x_i^+)) - d(f(x_i), f(x_i^-)) + m) \tag{2.7}$$

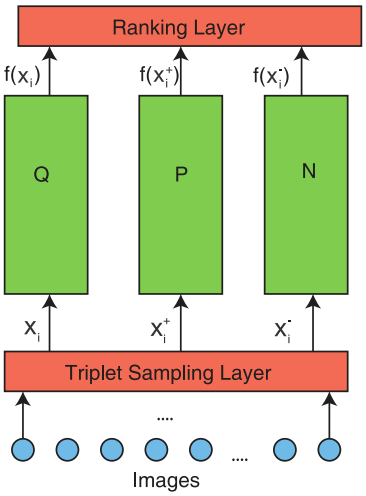

Figure 2.7: *Network architecture of the triplet model for learning fine-grained image similarity. Figure taken from [190]. Triplets consist of a query x_i, positive x_i^+ and negative x_i^-. These are each passed through a branch of the network with weights shared between branches. The output for each branch is an embedded feature where the triplet loss in the ranking layer encourages the positive to be closer to the anchor than the negative is.*

where m is the desired minimum margin between the two image pairs (x_i, x_i^+) and (x_i, x_i^-). To aid learning of fine-grained similarities, each branch of this triplet network (Figure 2.7) is a multi-scale network with three paths, two operating on lower resolution images.

Since triplets increase cubicly with the number of images, it is often not possible, nor desirable, to train with every possible triplet. Instead, Wang *et al.* ignore trivial negatives which already satisfy the loss and focus on sampling more difficult, in-class negatives. More recent work has looked at different ways of sampling triplets such as semi-hard negatives [128, 132, 157] (*i.e.* examples which look somewhat similar but are irrelevant to the query) and easy positives [206] (*i.e.* positive examples which the model is confident are relevant), although these sampling methods are only effective with large datasets.

2.2.2.2 Cross-modal Retrieval with Language

The rise of image sharing sites, such as Flickr, caused many more images to be available on the web. These images would be associated with user created captions or tags which give some, albeit noisy, supervision as to what the image contains. Works began looking at automatic image captioning [40, 86, 91, 204, 207], where the aim is to generate a caption for a given image. Many of image captioning works do this by learning a shared embedding space where related images and text are close together in the space, therefore retrieval became an obvious alternate task.

With increased number of web images, comes an increasing number of classes to divide these images into, particularly when using image tags [32]. This makes it increasingly hard to learn an image classifier as the distinction between the classes blur. Image retrieval alleviates some of these issues, as a visual-text embedding allows existing semantic knowledge about the relationship between classes to be utilised. Frome *et al.* [47] do just this. They pre-train a visual classification model and a skip-gram language model before combining the two. In the combination, a transformation is learnt from the output of the visual model to the semantic embeddings. By utilising semantic information gained with unannotated text from Wikipedia, the embedding model proposed by Frome *et al.* makes more reasonable semantic errors than previous methods and improves retrieval results for image classes not seen in training.

Cross-modal visual-text embeddings can be trained using a similar triplet loss to the one used by Wang *et al.* [190]. Instead, the query consists of text (t_i), while the positives (x_i^+) and negatives (x_i^-) remain as images and separate functions f and g are used to embed the image and text respectively:

$$L = \sum_{(t_i, x_i^+, x_i^-)} max(0, d(g(t_i), f(x_i^+)) - d(g(t_i), f(x_i^-)) + m) \tag{2.8}$$

Gong *et al.* [51] utilise the increasing amount web images to improve image-text retrieval. These images have noisy and weak annotations of title, tags or descriptions instead of the high-quality sentence descriptions normally used. They propose a method based on Canonical Correlation Analysis (CCA) [179] which is able to transfer information from a large number of weakly-annotated images to a smaller fully-annotated set. While they find the title to be the most beneficial type of weak-annotation, the combination of title, tag and description obtains the best performance. The addition of weakly-supervised data gives the largest performance increase when fewer fully-annotated images are used, although there is still a modest improvement with 25,000 fully-annotated images.

Instead of projecting images into a text space as Frome *et al.* [47] do, Wang *et al.* [193] learn a shared image-text space which pre-trained models for both image and text are projected into. To preserve some structure on this space, Wang *et al.* introduce additional triplet losses. In addition to having a text-to-image triplet loss between an image query and sentence positive and negatives, they also include a image-to-text loss:

$$L = \sum_{(x_i, t_i^+, t_i^-)} max(0, d(f(x_i), g(t_i^+)) - d(f(x_i), g(t_i^-)) + m) \tag{2.9}$$

where the triplets (x_i, t_i^+, t_i^-) consist of an image query and positive and negative text examples. Single modality video-to-video and text-to-text losses are also used, these are defined similarly to Equation 2.7. The authors find the addition of the image-to-text loss helpful, even when performing text-to-image retrieval. The structure preserving single modality losses also give an additional but smaller boost, although the largest improvement comes from using a non-linear function to project the image and text features into the shared space.

2.2.2.3 Retrieval in Videos

Video retrieval is most commonly performed in a cross-modal setting between videos and text (or vice versa), however videos introduce the additional issue of the temporal dimension where the relevance to the text may not be clear from every frame in the video. Recent works in video retrieval typically either focus on the most relevant parts of the text for retrieving a video or focus on finding which parts of the video are most relevant to a text query.

Text Relevancy. As with cross modal retrieval in images, additional noisy web data can also improve the text embeddings in cross-modal video retrieval. Otani *et al.* [129] help distinguish between fine-grained concepts in text by augmenting the training video data with image search results corresponding to the query text. They use an RNN to aggregate the word features from the text, however they treat the full video clip as relevant and average the frame features.

Torabi *et al.* [181] investigate the effect of using sequence based models to aggregate the word features. They demonstrate that using an LSTM to aggregate GloVe features of individual words is more effective than average pooling when retrieving movie clips. Torabi *et al.* also test whether using a learnt weighted average of the video frames is more successful than a uniform average. This is done through an attention mechanism which uses the output of the LSTM to weight frames. However, this showed little improvement, likely because the movie clips used [152] are only a few seconds in length and are trimmed to contain the video portion relevant to the text description.

Using Parts-of-Speech. Instead of aiming to discover the words in a query most relevant to retrieving videos, other works focus on specific categories of words *i.e.* parts-of-speech. For instance, Xu *et al.* [205] identify triplets of subjects, verbs and nouns in the text under the assumption that these words capture the essential semantic meaning necessary to match videos to text. They learn a dependency tree to compose the elements within these triplets before projecting them into a joint video-text embedding space.

Xu *et al.* find that when retrieving textual descriptions for videos their method is able to retrieve more specific captions, which are more likely to refer to the correct objects and actions than the CCA [170] baseline.

Mithun *et al.* [116] also focus on verbs and nouns. They learn two embedding spaces, an object-text space and activity-text space which are combined to perform the final retrieval. For both spaces, the sentence is embedded via a GRU. The input modalities from the video are separated, based on their applicability to each space. The video features in the object-text space are learnt with average pooling of RGB features over a video clip. To learn the video features for the activity-text space, motion features from a spatio-temporal 3D CNN are fused with a 1D CNN operating on the audio. Separating the spaces proves much more effective than learning a joint video-text-audio space.

Wray *et al.* [202] also learn separate spaces focussing on actions and objects, however these spaces are informed by separate parts of the text as opposed to separate input modalities. Verbs and nouns are identified from the input caption with part-of-speech parsing. A separate video-text embedding space is learnt for both nouns and verbs where both appearance and motion features are used to embed the video in each space. A final joint video-text embedding space is then learnt from the combination of individual component spaces. Wray *et al.* find this approach of first disentangling the embeddings to be more successful than learning a single video-text embedding either with the entire caption or only the verb and noun, especially for the task of action recognition.

This work also investigates whether including additional embedding spaces other parts-of-speech are useful for retrieval in videos, including determiners (*e.g.* the, every), adjectives (*e.g.* red, wooden) and adpositions (*e.g.* on, up). These additional parts-of-speech had little effect of the results of both video-to-text and text-to-video retrieval, likely as they are much rarer. For instance, in MSR-VTT [204] there is an average of 0.63 adjectives per caption in comparison to an average of 3.33 nouns per caption. Another possibility is that these types of words have less of a consistent physical appearance than actions or objects, thus an embedding space is not an effective way to represent these parts of speech.

Temporal Relevancy. Different to image retrieval, video retrieval poses the additional challenge of the temporal domain. Not all information relevant to the text will present in a single frame, instead certain parts of the video may be more relevant to different parts of the text. Hendricks *et al.* [4] propose the problem of localising moments in video using natural language queries. Queries come from annotator descriptions of distinct video moments. Similarly to action localisation (see Section 2.1.2), localising with

language queries goes beyond clips of single actions and requires finding the temporal bounds of events in untrimmed video. This is a more find-grained problem then action localisation as the text will refer to specific moments, rather than an entire class of actions. Additionally, due to words like 'before' or 'after', global video context is needed to contextualise the event. Hendricks *et al.* combine local features, average pooling of global video features and temporal endpoint features and aim to minimise the squared distance between the combination of these and the sentence feature.

Mithun *et al.* [117] since extended moment retrieval with natural language to a weakly-supervised problem. Given a text query, they aim to learn to retrieve the relevant part of a video with only video-level labels. To this end, they learn a joint video-text embedding where the sentence feature is used to guide the embedding of the video feature. The sentence feature is obtained with a GRU over the features of individual words. Video segments are then weighted according to their cosine similarity to the query sentence and the pooled video feature is embedded in the joint video-text space. This approach obtains comparable results to fully supervised methods for the recall of the top 10 results, although underperforms when fewer results are retrieved.

As mentioned in Section 2.1.3, instructional videos contain text which corresponds to the video, albeit with a large amount of noise. Miech *et al.* [110] introduced the large-scale HowTo100M dataset and demonstrated its use for pre-training before fine-tuning on a target task. In the original paper, the authors focused on the scale of the data as a way to mitigate the effect of the misalignment between the videos and narrations.

In their follow up work, Miech *et al.* [111][3] present a new loss to address the misalignments and the noise in narrated instructional videos. This approach combines multiple instance learning (MIL) [33] with noise contrastive estimation (NCE) [55]. As opposed to the triplet loss, which can have issues with sampling the correct negative, NCE aims to distinguish the positive example from a distribution formed with multiple negative examples. This handles the noisiness of instructional videos and avoids the case where the chosen negative in a triplet loss may actually be a mis-labelled positive. To cope with the misalignment, Miech *et al.* add an MIL component to the NCE loss. In the MIL-NCE loss, the best matching a narration from a set of possible positive narrations is used. The authors find the optimal size of this set of potential positives to be 3-5, with performance dropping when a larger set is used. This means the proposed loss is able to better cope with the slight misalignments often present in instructional videos, but cannot cope with a large misalignment between the action happening and being

[3]Work done concurrently with Chapter 6.

described. Miech *et al.* use their method to train on the HowTo100M dataset and then finetune to several downstream tasks in action recognition, text-to-video retrieval and action localisation. For every task, the MIL-NCE loss is able to outperform pre-training on HowTo100M with a standard cross-modal triplet loss.

In summary, recent works in video retrieval highlight the importance of learning a good video embedding feature whether this be through including a large number of (noisy) examples or by focussing on only the most relevant video parts. Other works have also shown that focussing certain parts-of-speech more relevant to the target task can improve results over using the entire caption.

2.2.3 Ranking and Retrieval Conclusion

Ranking and retrieval are highly related problems which aim for a fine-grained understanding of the data. The majority of works in video understanding have focussed on classification tasks where the predicted label is either correct or incorrect. Ranking and retrieval instead focus on the difference between videos and whether one video is more relevant, or contains more of the desired property, than another.

While ranking and retrieval do share some similarities, the goal of each, and therefore the techniques used, are quite different. A key problem in both video ranking and video retrieval is determining which parts of the video are relevant to the ranking criteria or retrieval query. The majority of works ignore this issue, or deal with short video clips with irrelevant parts already removed. While HowTo100M offers a large weakly-supervised video dataset, due it the noise contained in the dataset, the focus is on pre-training for downstream tasks on smaller, fully labelled datasets. Therefore, the question of how to determine the most relevant parts of a video still remains. Chapter 4 focuses on this problem in the context of ranking for skill determination, while Chapter 6 looks at how to learn weakly-supervised embeddings for adverb retrieval.

2.3 Skill Determination

Skill determination is the problem of evaluating the performance of a participant in a particular task. Depending on the scope of the task it can also be referred to as 'action quality assessment'. This is not to be confused with image [99, 199] or video [200] quality assessment, which assess the quality or aesthetics of the footage itself. Works in action quality assessment have the same goal as skill determination, although the tasks only consist of a single action and are therefore typically only seconds in length.

This section focuses on previous efforts in skill determination from video. While other modalities such as accelerometer data [38, 46, 106, 198, 220] have been used, video is more widely applicable to different tasks and doesn't require use of specific tools.

Skill determination can be formulated as either a classification [93, 150, 154, 163, 164, 219, 221], regression [52, 133, 134, 135, 147, 203] or ranking problem [9, 106, 107]. With classification and regression, the aim is to predict the correct score for the task. Regression can be used when this score is a continuous variable, as with Olympic scores for diving, while classification can be used for categorical scores. This however relies on there being a predefined scoring metric for the task. Without a scoring metric, skill determination can be formulated as a ranking problem where the aim is to correctly order videos in accordance with an expert's ranking.

Previous work in skill determination has focussed on tasks in two domains: sports and surgery. Participants in these tasks have to go through a large amount of training, therefore automatic feedback in these areas would vastly reduce the load on instructors. Also, these tasks tend to have available scoring systems and therefore more easily obtainable ground-truth. Since the majority of methods focus on these specific domains, most skill determination approaches are not applicable outside of either sports or surgery. Accordingly, Section 2.3.1 will focus on prior skill determination works in sports, while Section 2.3.2 will examine previous works in assessing surgical skill. Section 2.3.3 then goes on to examine several problems distinct from, but related to skill determination.

2.3.1 Sports

Sports is a natural domain in which to study skill as they require intensive training, therefore any automated feedback system would be able to alleviate the strain on instructors. For many sports, annotations are also easy to obtain as the competitive nature of sports means teams or individuals are scored and ranked in their performance. The majority of works in sports focus on individual Olympic sports with ground-truth scores. These sports, such as diving or gym vault, tend to only take a matter of seconds as they contain a single, albeit complex, action. Skill assessment on these sports is often referred to as action quality assessment as the complexity and fluidity of the motions are the main indications of skill.

Specific Sports. Gordon [52] was the first to examine automated assessment of skill in videos. In this paper, Gordon aimed to address the question of which type of performances were appropriate for automated assessment from videos. He concluded that only tasks where it was possible to develop a scoring rubric were suitable and that all features

relevant to the scoring should be "directly observable or explicitly derivable" [52] from the source video. This mainly constitutes tasks where a certain set of physical actions need to be executed in a specific manner. As an example of an appropriate task, Gordon studied the gymnastic vault and proposed to predict the score from the captured trajectory by deducting points for violating specific properties. For instance, a deduction is given if the performer does not reach an appropriate height or does not land a sufficient distance away from the vault. These features are specific to the gym vault task and therefore Gordon posits that further tasks require their own set of rules.

Other more recent works have continued this idea of designing features to assess skill in a specific task, such as in basketball [9, 77, 143]. Bertasius *et al.* [9] use footage from head mounted cameras to assess the performance of individual players in a basketball game. To assess skill, Bertasius *et al.* first train a ball detector to locate the position of the basketball in each frame. The region of the frame around the ball is then used to classify the detection into several different basketball events, such as shooting the ball or possession of the ball. Detections of these activities in a 10 second window are then used to predict the performance of a player using a Gaussian Mixture Model. To create an overall video score, the clips are aggregated using relevance, however this relevance is hard-coded as whether any player is shooting the ball. Since there are no meaningful scores for individual players in basketball, Bertasius *et al.* instead formulate the basketball performance assessment problem as a ranking problem with ground-truth scores from a basketball expert.

Multiple Sports. Pirsiavash *et al.* [147] were the first to look more generally at skill assessment in sports and proposed a method applicable to both diving and figure skating. They use a Discrete Cosine Transform to map a time series of automatically predicted body poses into the frequency domain. Since diving and figure skating are both Olympic events, they have readily available scores, thus skill determination in these tasks is posed as a regression problem. Pirsiavash *et al.* also examine the applicability of their action quality assessment method to highlight detection and feedback generation. Highlights are generated by selecting the segments furthest from the average score of the video. The feedback generation is specific to the pose estimation framework proposed, as it estimates the way in which a joint should be moved to maximise the predicted score. While this work achieved success in determining skill for more than one task, the authors note several limitations. First, the method is heavily reliant on the predicted pose, if this prediction is incorrect the predicted score will be unreliable. Second, the authors remark that they "do not model objects during actions (such as sports balls or tools) and do not consider physical outcomes (such as splashes)" [147]. Therefore, this method

is unsuitable for any task where the body pose is not sufficient for assessing skill.

Parmar and Morris [134] aimed to remove the need for body pose in these tasks and instead proposed a more general method to predict scores for diving, gym vault and figure skating. This method uses features from C3D [182], a 3D CNN which had shown success in recognising different sports actions. In this work, Parmar and Morris also note that not all parts of a video are equally important "a diver may be perfect through the air but fail to enter the water vertically and make a large splash which is reflected as a poor overall dive score" [134]. Thus, they propose to use two LSTMs on top of the C3D features, one to predict the execution score and one for the difficulty score. However, this is not appropriate for longer tasks like figure skating as LSTMs struggle with longer sequences. Parmar and Morris also found that using support vector regression (SVR) on average pooled C3D features vastly outperformed the LSTM approach (0.57 versus 0.74 Spearman's rank correlation for diving). The results for C3D+SVR were a large improvement over the prior, posed-based method of Pirsiavash *et al.* [147] (0.41 versus 0.74) and demonstrate the suitability of extracted CNN features for skill determination.

Using Additional Information. Instead of averaging all features uniformly across a video, Xiang *et al.* [203] look at specific parts within a task. They identify four components of diving: jumping, dropping, entering and ending, and automatically segment videos into these parts. Xiang *et al.* train a separate network for each part, as the features which identify skill may be distinct for each. The output of each network is then concatenated before regressing to the overall score. This showed an improvement over the averaged C3D features used by Parmar and Morris [134], however breaking a task into specific components makes the method more task specific and unable to be extended to longer tasks, where videos may vary in the actions they contain.

Recently, Parmar and Morris [135] also incorporate additional information to improve regression to an action quality score. This work uses a multi-task learning approach where two supplementary tasks are included in addition to action quality assessment. The first is predicting attributes of the dive, such as the position and the number of somersaults it contains. The second is generating captions based on the event commentary. These additional tasks are particularly helpful when less training data is available; the proposed method obtained a +4 improvement in Spearman's rank correlation when using 10% of the training data versus a +1 improvement with the full dataset.

Transfer Between Skill Tasks. To better examine the relationship between action quality assessment on different sporting tasks, Parmar and Morris presented the AQA-7 dataset [133] consisting of 7 tasks. The tasks are: diving, synchronised diving from

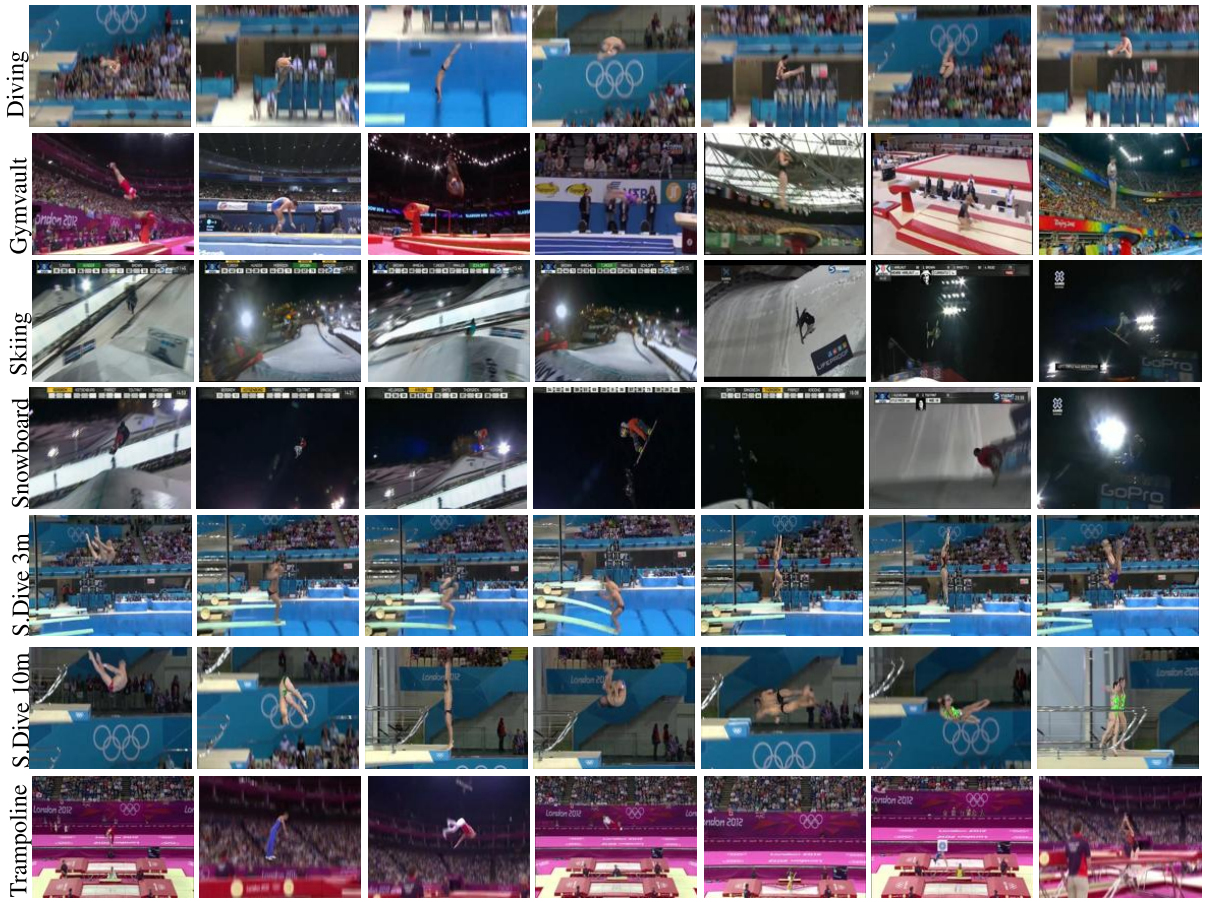

Figure 2.8: *Examples from each of the 7 tasks in the AQA-7 dataset. Figure taken from [133].*

both 3m and 10m platforms, gym vault, ski big air, snowboard big air and trampolining. Examples of these tasks can be seen in Figure 2.8. Footage comes from televised coverage of Olympic events. In total the dataset contains 1,189 videos. All tasks take approximately 5 seconds or less, except trampolining which is around 20 seconds per video. While the tasks each have different setting and scoring metrics, it is expected that they will share some common high-level features since all tasks involve jumping and doing somersaults and twists in the air. An incorrect landing may look very different between diving (a splash) and gym vaulting (falling over), however the landing is a key factor in assessing the performance in all of these tasks. Therefore, this dataset is ideal for examining the relationship between skill in different tasks and how skill models can be transferred between tasks.

Parmar and Morris [133] investigate this with several basic transfer learning methods. They use their C3D-LSTM framework [134] to test zero-shot transfer between tasks, *i.e.* they train a model on one task and directly apply it to another task without fine-tuning. In many cases, transferring from one task to another produces near random

performance. Some related tasks do obtain reasonable performance, for instance when transferring from synchronised diving 10m to synchronised diving 3m the Spearman's rank correlation is 0.44, although the same is not true for the reverse transfer. There are also some unpredictable positive transfers, such as skiing to diving. This highlights that more work is needed to understand the skill-relevant features and their applicability to other tasks. Parmar and Morris, also test multi-task learning, finding that learning the six shorter tasks together (everything except trampolining) is helpful in most cases.

In summary, the majority of work in skill determination for sports has followed Gordon's notion [52] of appropriate tasks being those with scoring metrics, where skill is the quality of movement in a predefined set of motions. Most tasks studied are only seconds in length and each part of the video is assumed to have equal importance, leaving the possibility of assessing skill in longer, more complex tasks an open question.

2.3.2 Surgery

Videos in the surgical domain are typically minutes in length, therefore skill determination work in this area has been more focussed on the temporal aspect of the video. While videos of surgical tasks are generally used to obtain ground-truth, many works instead use accelerometer data to predict skill [38, 46, 106, 107, 150, 154, 198, 220]. This was particularly true for earlier work [106, 150, 154], however more recent methods have begun using video [10, 74, 163, 221] as this data can be easier to capture.

Works which aim to determine skill in surgical tasks can generally be divided into two categories, those which use global motion features [10, 106, 107, 163, 220, 221] and those which first split the task into a sequence of different actions [93, 150, 154].

Sequences of Actions. Rosen *et al.* [154] take the latter approach, building hidden markov models (HMMs) to describe the transition between surgical actions (surgemes) for both experts and novices from accelerometer data. The likelihood of a given sequence of actions belonging to either the expert or novice categories can then be used to identify skill level. Reiley and Hager [150] extend this approach to individual actions. In this work, each trial of a suturing task is first parsed into a sequence of surgemes. For each of the 8 surgemes, the authors train 'beginner', 'intermediate' and 'expert' HMMs. After calculating the most likely model for each surgeme, majority voting is used to classify the sequence into the appropriate skill level.

One disadvantage of this approach is that novices will perform the surgemes in a different way to experts, making it difficult to parse lower skill trials into sequences of surgemes. Lin and Hager [93] aim to improve the segmentation into surgemes using video data,

instead of accelerometer data. They find using other contextual clues from video data to be much more robust to variability in skill level and obtain a large improvement in the result of skill classification when one skill category is absent from training.

Surgery-Specific Metrics. However, these approaches still require knowledge of the type (or at least number) of possible actions within a task. In order to create a more general approach across different types of laparoscopic (keyhole) surgeries, Malpani *et al.* [106] use global task features. These include the time of completion, the total path length traversed by the surgical tools and the area swept by the instrument wrist. This work was later extended by Malpani *et al.* [107] to include further features such as the number of times the gripper was closed and the number of peaks in magnitude of the instrument tip's velocity. In these works Malpani *et al.* also moved away from the coarse grained categorisation used by prior works, instead opting for a ranking approach. By using a RankSVM to compare the global features across different trials, Malpani *et al.* were able to accurately rank skill for both suturing and knot tying tasks.

More recently, this type of approach has been brought into the deep learning era by using region based object detection methods to identify the locations of tools [74]. From tool bounding boxes, the tool usage patterns and economy of motion can be calculated with metrics such as the total time each instrument is used and the distance travelled.

Crowd-sourced Annotations. The works from Malpani *et al.* [106, 107] were also the first to examine crowd-sourcing of skill annotations. As well as obtaining skill ranking annotations from expert surgeons, annotations were also crowd-sourced[4]. Malpani *et al.* [107] found that the inter-reliability between the crowd was not too disimilar to the inter-reliability of experts (0.81 versus 0.88 Fleiss kappa). Furthermore, when taking expert preferences as ground-truth, pooled crowd preferences were at least 83% accurate. This demonstrated that crowd-sourcing is a viable way to collect skill annotations, even for surgery tasks where it seems prior knowledge would be necessary.

Motion Features. More recent works have aimed to obtain motion features directly from video data and make these global features more applicable across different types of surgery. These more recent works also aim to assess skill by predicting OSATS scores [108] instead of categorising experience or ranking. OSATS criteria are measured on a 1-5 point scale and cover various aspects of surgical skill, such as respect for tissue and flow of operation.

Bettadapura *et al.* [10] propose augmented bag of visual words. This operates on pre-

[4] from undergraduate students outside of medicine

extracted Harris3D and HoF features and uses detected temporal events to capture the time and co-occurence of visual words. As well as being able to recognise different activities, the authors show the proposed method can correctly categorise videos into different OSATS scores with $> 60\%$ accuracy across the different criteria.

Since this work, the majority of effort in surgical skill assessment has gone towards improving the features used to model the motion dynamics across a video. Sharma *et al.* [164] proposed predicting skill with motion textures, later extended to sequential motion textures [163]. These are computed from kernel matrices which encode the similarity of clustered STIP, HoG and HoF features between two frames. The authors found motion texture features to be successful for assessing skill as motion existed in nearly every frames of the novice videos, while experts performed far fewer motions and did so in a more deliberate manner.

Zia *et al.* [220] improved upon this method by using the Discrete Cosine Transform and Discrete Fourier Transform on the same STIP, HoG and HoF features. This gave a large increase in classification of OSATS criteria for both suturing and knot tying and is computationally much less expensive than sequential motion textures. However, the authors note that the method is "designed for basic repetitive types of surgical motions" [220] and that other feature types not dependent on the periodicity should be used for non-repetitive tasks.

Building on this work, Zia *et al.* [221] used approximate entropy in the time series to assess skill and proposed cross approximate entropy to measure the asynchrony between two time series, *i.e.* the two hands of a surgeon. While the results of these methods were comparable to prior approaches when using accelerometer data, they vastly improved the performance with extracted video features. Again, since approximate entropy is a measure of regularity, this method relies on the repetitive nature of surgical tasks.

JIGSAWS Dataset. Due to the nature of surgical tasks, many of the above approaches do not use publicly available data, or if they do it is very limited in scale. The JHU-ISI Gesutre and Skill Assessment Working Set (JIGSAWS) dataset works towards solving this issue. This dataset contains three tasks performed using the da Vinci tele-robotic surgical system [54]:

- **Knot-Tying**: a subject has to tie two loop knots with suturing thread around a flexible tube.
- **Needle-Passing**: a subject first picks up the needle and then passes it from right to left through four small metal hoops. The hoops are attached to a flexible material a small height above the surface.

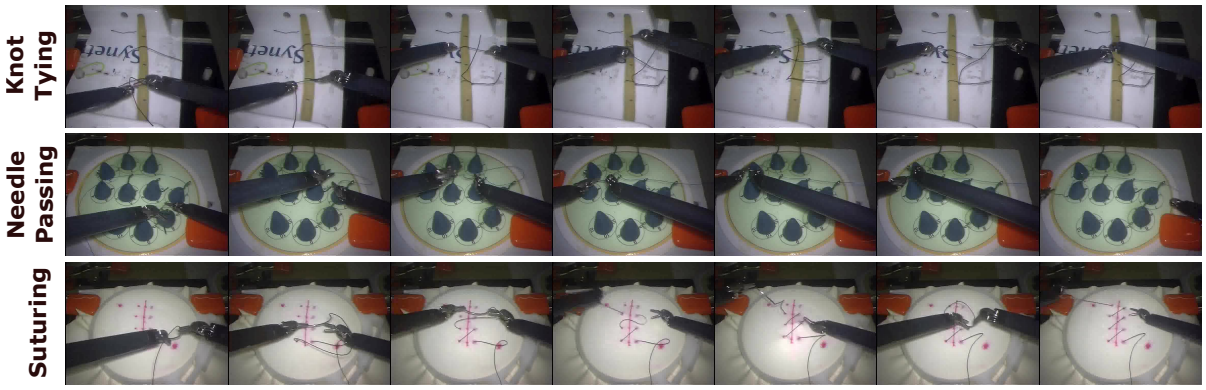

Figure 2.9: *Examples videos from the Knot-Tying, Needle-Passing and Suturing tasks in the JIGSAWS dataset.*

- **Suturing**: after picking up the needle, the subject has to stitch around an 'incision', denoted by a vertical pink line. To do this the needle must be passed through the guide dot on the right hand side and exit through the guide dot on the left. This is then repeated three more times.

Examples of each task can be seen in Figure 2.9. All three tasks are performed by eight participants, with each participant repeating the tasks five times. As some data was corrupted during the collection of the dataset, there are 36 trials available for Knot-Tying, 28 for Needle-Passing and 39 for Suturing. The performances of these tasks range from 30 seconds to 3 minutes. Knot Tying is generally the shortest task, with an average of 57 seconds per trial, while Needling-Passing and Suturing have averages of 108 and 112 seconds respectively. Both accelerometer data and 640×480 stereo video data are available for use.

The dataset is annotated with surgical gestures, the user's level of past experience[5] and OSATS criteria [108] modified to exclude irrelevant factors such as the use of assistants. To obtain the OSATS annotations, a surgeon with extensive experience in laparoscopic surgery watched each video. The JIGSAWS dataset uses six criteria, all marked in the range from 1-5: respect for tissue, suture/needle handling, time and motion, flow of operation, overall performance and quality of the final product. These are then combined to obtain a global score for each video.

In summary, unlike skill assessment in sports tasks, methods dealing with surgical tasks have to cope with long videos. Much of the research in this area has examined how to deal with this temporal aspect, either through modelling the progression of the task through different surgical actions [93, 150, 154] or extracting global motion features to

[5] 'Beginner': < 10 hours, 'Intermediate': > 10 and < 100, 'Expert': > 100

describe the full task [10, 106, 107, 163, 220, 221]. Both of these types of methods rely heavily on the constrained and repetitive nature of surgical tasks. Although recent work in motion features have made them more automated and less task-specific, these methods are still unsuitable for non-surgical tasks. This thesis explores skill-determination in a variety of daily-living tasks where the videos are minutes in length.

2.3.3 Tasks Adjacent to Skill Assessment

While work in skill determination has been focussed in specific domains, there have been more generalisable methods proposed for problems related to skill determination. This section will describe these problems, prominent works which tackle these problems and their relationship to skill determination.

Action Completion. This is the task of identifying whether an action has been successfully achieved. While many works have aimed to identify what an action is [18, 70, 139, 146, 166, 182, 189, 194], Heidarivincheh *et al.* [58] were the first to detect whether an action has been finished successfully. In this work, the authors propose a supervised approach to build a completion model per action using a set of human pose features. This work was later extended in [59], where Heidarivincheh *et al.* use a convolutional RNN with frame-level voting to identify the moment at which the action is completed.

The problem of action completion is similar to skill determination in that it aims to determine whether a person has been successful at the task. Similar to action quality assessment in sports, action completion targets shorter videos, generally seconds in length, as opposed to the longer videos addressed in skill determination from surgical videos. However, action completion only gives a binary assessment of whether the person has succeeded or failed at completing the action. On the other hand, skill determination aims to separate this further and assess how-well has the task be completed.

Adverbs. How an action has been performed can be discovered by recognising the adverbs applicable to the action in a video. While video captioning datasets [81, 213] do contain adverbs, no prior work using these datasets aims to model or recognise these adverbs. The only prior work to utilise adverbs is that of Pan *et al.* [130]. This work presents a multi-stream model which fuses RGB, optical flow, pose and expression information to recognise an action and the relevant adverb. In addition to this, a spatial attention mechanism is trained with human bounding boxes to focus on the image regions where the action takes place. This was tested on a variety of different actions and adverbs, including 'smoke triumphantly', 'run freely' and 'sword exercise clumsily'. The authors found that the problems of action recognition and adverb recognition were not

highly related and did not benefit from sharing the same model.

While skill determination aims to assess 'how-well' a task or action is performed, adverbs more generally describe 'how' it is done. Some adverbs are directly related to skill such as 'clumsily' and 'expertly' and thus recognising these adverbs works towards assessing skill. Others could be related to the skill of a specific task, for instance it is ideal if surgery is done 'carefully' and 'quickly', but less important that it be done 'happily'. Chapter 6 aims to link adverbs and skill through the use of instructional videos.

Object Attributes. Related to adverbs of actions is the analogous task of learning attributes (or adjectives) of objects. Learning adjectives for nouns has been investigated in the context of recognising object-attribute pairs [11, 23, 39, 66, 87, 115, 123, 124, 131, 197] from images.

A standard approach of early work in this area was to train discriminative attribute classifiers from a pool of images across different object categories [11, 39, 87, 131]. Farhadi *et al.* [39] use logistic regression to select a subset of texture descriptors, HoG descriptors, edge orientations and colour descriptors which can generalise well across different objects and contexts. The idea being that if the presence of an attribute is annotated in enough different contexts, the features selected will be able to distinguish from common co-occurring attributes.

However, the appearance of attributes can vastly differ depending on the object the attribute applies to. Take the example 'old', while they could share some properties an 'old bike' will have a very different appearance to an 'old laptop'. Wang *et al.* [197] tackle this problem with a model which captures both the object-dependent attributes and the object-independent attributes. This is done with a Bayesian network which learns connections between pairs of attributes either directly (independent) or through objects (dependent).

Both Chen and Grauman [23] and Misra *et al.* [115] work on the problem of contextuality of attributes by aiming to classify unseen object-attribute pairs. Chen and Grauman [23] formulate this as a transfer learning problem to recognise unseen object-attribute compositions. Their proposed method discovers analogous relationships based on previously seen objects and attributes allowing it to naturally discover where transfer is possible. Misra *et al.* [115] instead learn a classifier for every attribute and object independently. They then learn an additional network which composes separate object and attributes classifiers for novel combinations.

Instead of using classifiers to recognise attributes, Nagarajan and Grauman [123] model attributes as a transformation of an object's embedding (see Figure 2.10). The proposed

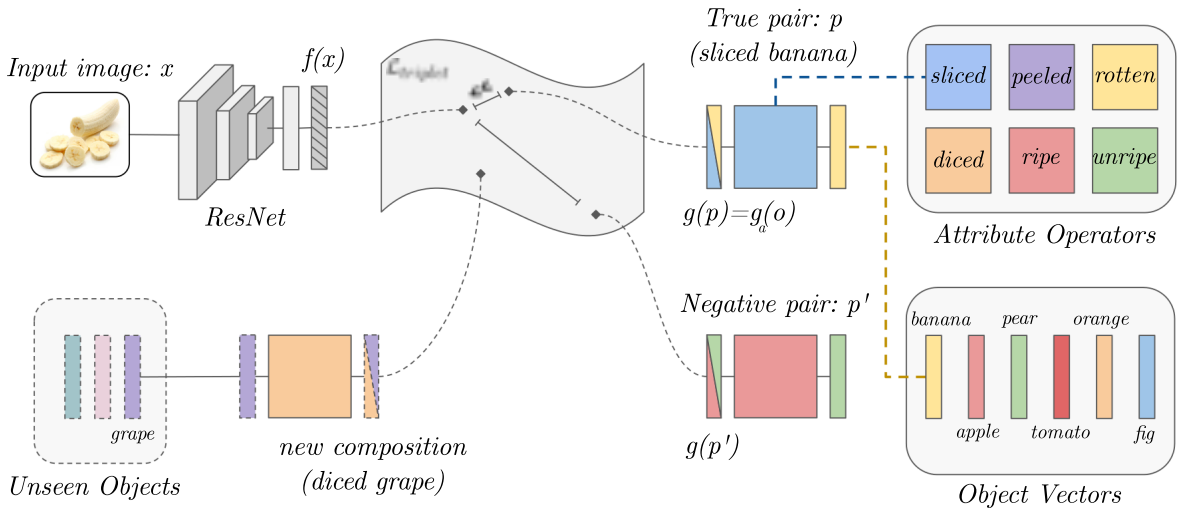

Figure 2.10: *Nagarajan and Grauman [123] propose a factorised model for recognising object-attribute pairs. Attributes are operators which modify the embedding of an object vector. This representation enables disentangling of the object and attribute and allows attributes to be composed with previously unseen objects. Figure from [123].*

method uses GloVe vectors [140] to represent objects and learns to represent attributes via a linear transformation matrix in the embedding space. With this, Nagarajan and Grauman can disentangle objects and attributes. Their method can generalise to unseen pairs by learning to embed an image close to the relevant object modified by any learnt attributes. The method presented in Chapter 6 is inspired by this approach. This paper also proposed several regularisers to help learn good operators, inspired by linguistic properties of adjectives. The regularisers ensure that the operators are invertible, commutative, undo the effect of their antonym and that an embedded image can be correctly classified as containing the relevant attribute and object.

As with adverbs, correctly identifying the attributes of an object could be used for skill determination. In some tasks, certain properties of intermediate or final results are required and their presence can indicate success in the task. For instance, if the egg whites are not whisked until they form 'stiff' peaks it is likely the meringue will not bake correctly. While these methods could be applied to skill determination, this thesis instead focuses specifically on the temporal domain and explores how these methods can be applied to the analogous task of identifying adverbs for actions in Chapter 6.

Chapter 3

Learning to Determine Skill from Video

This chapter explores skill determination for daily-living tasks, such as using chopsticks or rolling pizza dough. The aim is to go beyond the domains of sports and surgery, where automatic skill assessment is typically studied, and be able to determine skill for a variety of different tasks. This would allow automated feedback or could be used identify suitable videos to learn from.

Extending the domain of skill determination to include daily-living tasks is challenging. The nature of sports means there are predefined criteria to measure a person's or team's success. While in surgery, the severity of the problem and need for intensive training has led to scoring metrics being defined across different surgical tasks. However, in daily-living tasks such metrics are not available, nor would they be common across the diverse range of tasks encompassed by 'daily-living'.

Not only does this provide challenge in the annotation and formulation of the skill determination problem for daily-living tasks, but it means a method cannot incorporate prior task knowledge. For instance, in a method specifically designed to score videos of diving, body pose is an important feature and can be used explicitly [147]. In surgery, specific features, such as the trajectory of the needle, can be used to estimate the skill [106]. No such common features are present across the diverse range of daily-living tasks.

As the problem is to determine skill within a breadth of tasks, it is therefore also paramount to determine skill with relatively limited data per task. Gathering a large amount of data for each task would be impractical and would limit the amount of tasks where skill can be automatically assessed.

These challenges are tackled in this chapter. Specifically, Section 3.1 discusses the approach used to determine skill in daily-living tasks. Section 3.2 describes the collection

and annotation of the EPIC-Skills dataset for skill determination in tasks ranging from drawing to rolling pizza dough. In Section 3.3, naive metrics, such as time of completion, are discussed and the need to determine skill from video is explained. The method to determine skill from videos is then presented in Section 3.4 and experiments are conducted in Section 3.5.

3.1 Ranking Skill from Video

Automatic skill assessment from video would allow the wealth of online videos capturing daily tasks, such as crafts and cooking to be explored for training humans and intelligent agents. It could allow automated feedback of a person's performance in the task as well as enabling improved instruction, by helping users select the better videos to learn from.

Previous methods have focussed on determining skill, or assessing the quality of actions, in sports or surgical tasks. As these types of tasks have readily available scoring metrics, existing approaches often regress to this score [147, 163, 219]. Alternatively, other prior works take a coarse approach, instead categorising participants into 'novice' and 'expert' categories [48, 220]. Chapter 2 gives a full overview of these approaches. This thesis aims for a more fine-grained understanding of skill than simply classifying people into expert and novice, however daily-living tasks do not have existing scoring systems. While it would be possible to define scoring metrics for daily-living tasks, this would have to be done per task and designing such metrics would be difficult.

Instead, the assumption is made that if human observers consistently label one video as displaying higher skill than another, there is enough information in the visual signal to automate that decision. Examples of the types of annotations humans could easily provide are shown in Figure 3.1 (left), where videos could either be similar in the skill they display ($\approx$), or one video could be obviously better than another ($>$).

This chapter proposes to determine skill with a pairwise deep ranking model, which characterises the difference in skill displayed between a pair of videos, where one is ranked higher than the other by annotators. The method uses a Siamese architecture where each side of the Siamese network is a two-stream (spatial and temporal) CNN. The Siamese architecture is trained using a novel ranking loss function which considers the extent of the task within the video, and includes pairs of videos indistinguishable in terms of skill. By assigning videos a relative skill score for the given task, it is possible to learn to predict an overall *skill ranking* from pairwise annotations.

Figure 3.1: *Determining skill in videos of daily tasks.* ***Who's Better*** *(Left): pairwise decisions between videos of the same tasks, performed with varying or comparable levels of skill.* ***Who's Best?*** *(Right): ranking learned per task from the sets of pairwise decisions.*

3.2 EPIC Skills Dataset

As outlined in Section 3.1, the aim is to be able to rank skill for a variety of daily-living tasks. The existing skill datasets (outlined in Chapter 2) are either in the domain or sports or surgery and are therefore not suitable for this purpose. Datasets to determine whether people have successfully completed an action do exist [59] (see Chapter 2 for further detail), however skill determination focuses on fine-grainer differences rather than making binary decisions about success. Therefore, a new skill determination dataset is necessary. The collection and annotation of this dataset is described in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Data Collection

A search was first carried out to ascertain whether any existing video datasets can be re-purposed and annotated for skill determination. This would save the time and resources needed for data collection and allow future works to explore the use of existing annotations for determining skill. For example, annotations of temporal action boundaries could be used to exclude actions irrelevant to skill or allow overall skill to be predicted from the assessment of individual actions. The following datasets were explored for potential tasks: 50 Salads [137], Breakfast [85], BEOID [29], CMU-MMAC [31], GTEA Gaze+ [41] and MPII Cooking [153], as they contain multiple participants performing the same task. However, these datasets were mainly created for the purpose of action or activity recognition and therefore the activities are relatively easy to complete. Thus, the majority of videos within a task have little variation in skill. CMU-MMAC did show variation within some tasks.

3.2.1.1 CMU-MMAC

The Carnegie Mellon University Multi-Modal Activity dataset (CMU-MMAC) is an egocentric kitchen-based dataset. In CMU-MMAC, 39 participants complete five different food preparation tasks in the same kitchen. Participants follow the same recipe and have access to the same tools and ingredients, but may differ in the sequence of actions used to complete the tasks. The five tasks are preparing brownies, pizza, a sandwich, a salad or scrambled eggs. Video is captured by five cameras, one head-mounted and the remaining four stationary. Audio, motion capture, accelerometer and gyroscope data are also recorded.

Through observation, it was determined that the pizza making task showed the greatest variety in terms of skill. This activity generally contains the following steps: collect ingredients, open can of pizza dough, take dough out of can, unroll dough, neaten edges of dough, add tomato sauce, spread tomato sauce, grate cheese, cut sausage, add sausage to pizza, put pizza in oven and put away ingredients. Neither collecting the ingredients nor adding toppings show variety in terms of skill, therefore these steps are excluded from the videos. This leaves the opening and rolling of the pizza dough which is subsequently referred to as the Dough Rolling task. The recording from the head-mounted camera is used, as this contains the best view of the task. Once the egocentric videos are trimmed to contain only the steps of interest, there are 33 videos[1] with an average duration of 1 minute 42 seconds and standard deviation of 29 seconds.

3.2.1.2 Data Recording

Since the search in existing datasets only yielded one task, it was necessary to record further tasks. Drawing and Chopstick Using were selected as suitable tasks. These tasks displayed variety in terms of skill within the available participants and are significantly different to each other and the Dough Rolling task.

Drawing. Participants were asked to draw a copy of a reference image. Multiple variations of the Drawing tasks were created to allow future work in sharing skill between similar tasks. In the first, participants were asked to draw a cartoon image of Sonic the Hedgehog. In the other variation, they were asked to copy a grey-scale photograph of a hand. Similarly to the Surgery tasks, each participant repeated the task five times. This gives a large variation in skill as participants generally improved between runs. For each Drawing task, four participants were recorded, resulting in 40 videos (20 per task).

[1]Despite CMU-MMAC containing 39 participants, only 33 videos are available for the pizza making activity.

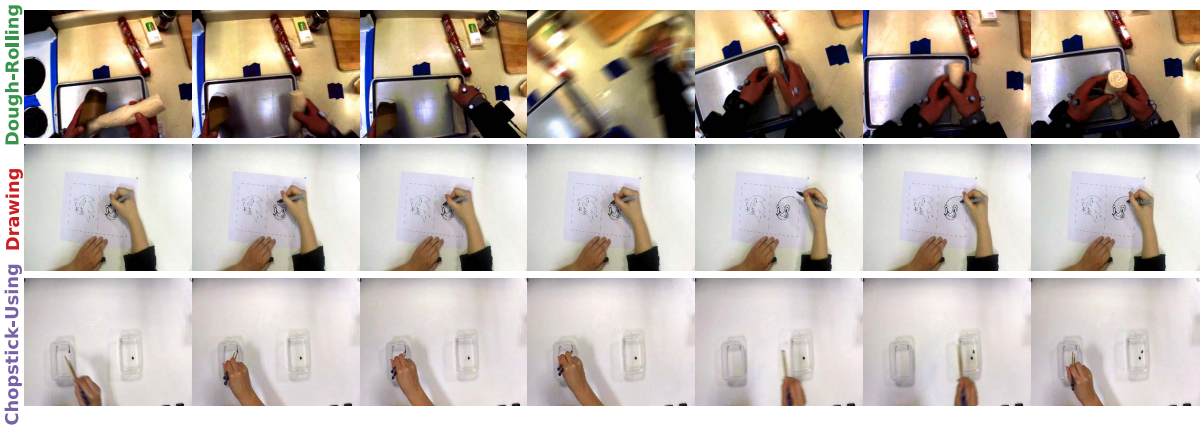

Figure 3.2: *Example Videos from Dough Rolling, Drawing and Chopstick Using*

Chopstick Using. In this task, participants were challenged to pick-up and move coffee beans using chopsticks. The set-up consists of two pots, one containing four coffee beans. The participants' goal was to move those four beans to the other, empty, pot. The weight and rounded nature of the coffee beans meant even highly-skilled participants found this task challenging. Participants were limited to one minute per trial in case they were unable to complete the task. As in Drawing, each participant completed five trials of the task. With eight participants, this resulted in 40 videos of Chopstick Using.

Both tasks were captured using a stationary and head-mounted camera at 60fps with a resolution of 1920×1080 pixels. The rest of this thesis uses the stationary camera recording, which is mounted to have a bird's eye view of the task, just above head height. An egocentric viewpoint can often provide a more insightful view of the task, as it captures the participant's viewpoint, however the participants' attention did not always follow their hands. For example, several participants focused solely on the reference image in Drawing[2]. Examples of the videos obtained for these tasks, and the Dough Rolling task from CMU-MMAC, can be seen in Figure 3.2.

The method proposed in this chapter is also tested on the Surgery tasks from the JIGSAWS dataset [48] due to the similarity in length and complexity and to further test the generality of the proposed approach.

[2]Results comparing the stationary and egocentric viewpoints are published at https://www.eyewear-computing.org/EPIC_ICCV17/short_papers/EPIC17_id17.pdf

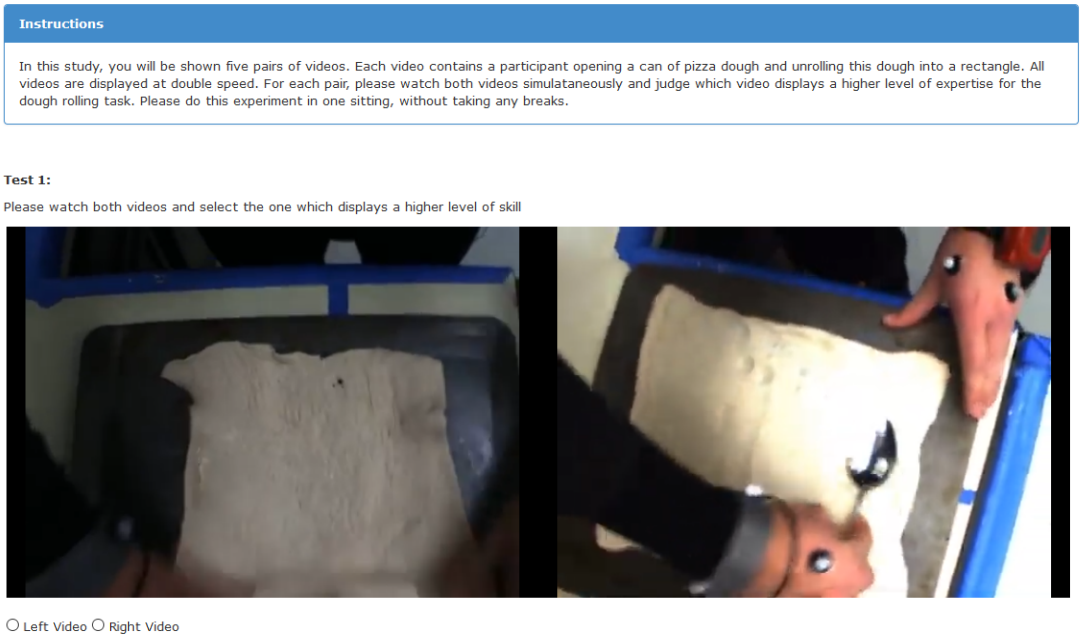

Figure 3.3: *The interface used to collect pairwise rankings.*

3.2.2 Data Annotation

The Surgery tasks from the JIGSAWS dataset have existing skill scores from a surgical expert (as outlined in Chapter 2). However, annotations needed to be collected for the Dough Rolling, Drawing and Chopstick Using tasks.

As explained in Chapter 2, pairwise comparisons have often been used to provide effective assessments when obtaining an absolute score is difficult. Malpani et al. [107] showed that it is possible to crowd-source this type of annotation for skill in surgery where non-experts were able to rank tasks with reasonable accuracy (see Chapter 2). Since it is hard to obtain experts for daily-living tasks, a similar approach was followed to annotate the Dough Rolling, Drawing and Chopstick Using tasks with crowd-sourcing.

Annotations were obtained through Amazon Mechanical Turk (AMT). This is a platform which allows people around the world to complete short tasks, known as Human Intelligence Tasks (HITs). AMT workers were asked to watch pairs of videos from the same task simultaneously and select the video displaying the higher level of skill. Annotators were asked for strict preferences per pair. Each worker was presented with five video pairs per HIT from the same task. This allows workers to become familiar with the task, and the skill variation within the task, before submitting their rankings. Figure 3.3 shows the pairwise ranking interface.

Since the process is crowd-sourced, it is important to ensure the quality of the annotations. Several filtering techniques are used, such as ensuring the workers have prior experience with AMT ($>$ 500 HITs completed), have a reasonable approval rating for other annotating tasks ($>$ 95%) and do not complete the HIT in significantly less time than it would take to watch the pairs of videos. In addition, one video pair in the HIT is a quality control pair with an obvious difference in skill between videos.

To remove subjectivity in preference, and account for cases where a pair of videos may contain a similar level of skill, the consensus of the crowd is used. Each video pair is annotated by four workers and only pairs of videos where *all* annotators agree on the pair's ordering are taken as ground-truth annotations. This set of video pairs are thus referred to as the *consistent pairs*.

It is important to note that pairs in the set of *consistent pairs* will not necessarily create a complete ranking of the videos. Many pairs will not have a consistent ranking agreed on by the annotators, either due to noise from the annotation process or pairs being similar in terms of skill. Furthermore, it may not be possible to infer the ranking of these pairs from the set of *consistent* pairs. An example of this partial ranking is shown in Figure 3.4 for the Chopstick Using task. The method proposed in Section 3.4 will take this into account and will only require ranked pairs of videos.

The *consistent pairs* are further checked for any discrepancies, by testing for *triangular inconsistencies*. Assume there are three videos x_i, x_j and x_k with annotations indicating x_i shows higher skill than x_j ($x_i > x_j$) and x_j shows higher skill than x_k ($x_j > x_k$). This gives a ranking $x_i > x_j > x_k$. From this ranking it can be inferred that $x_i > x_k$. However, if the set of pairs was $\{x_i > x_j, x_j > x_k, x_k > x_i\}$ there would be no overall ranking. These triangular inconsistencies are discovered by creating a directed graph where nodes are the set of videos in the task and edges are the set of ranked pairs, *i.e.* the edge $x_i \rightarrow x_j$ is created if video x_i had been annotated as having higher skill than video x_j. Such a graph is shown in Figure 3.4 for the Chopstick Using task. Cycles in this graph would indicate a triangular inconsistency. Only a single triangular inconsistency was found in the annotations. This was in the Dough Rolling task. The pairs forming the inconsistency were excluded from the set of *consistent pairs*.

For the Surgery tasks within JIGSAWS, numerical scores are converted to pairwise ranking annotations to match the collected annotations for EPIC-Skills. This allows the same method and evaluation to be used for all tasks.

It is also interesting to obtain pairs which display a similar level of skill. For the Surgery tasks, this is simple, pairs with the same numerical score are considered as the set of

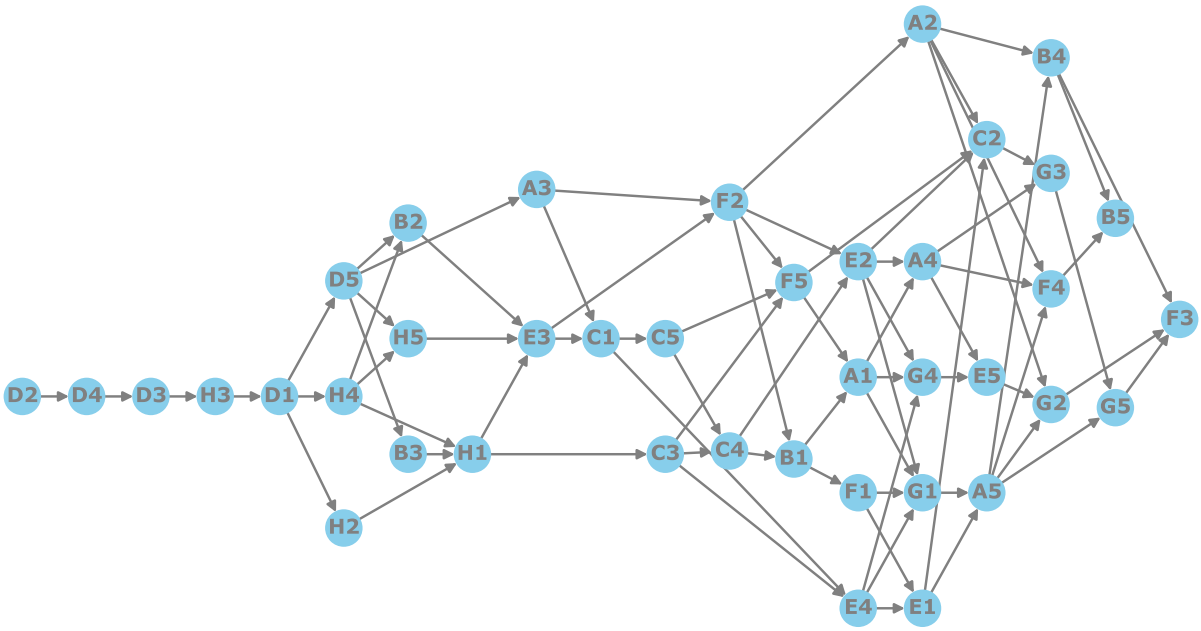

Figure 3.4: *A graph visualising the annotations for the Chopstick Using task. Videos are nodes and edges indicate a pairwise annotation* e.g. *D2 → D4 indicates that video D2 has been annotated as having higher skill than video D4*[3]*. Edges which can be represented as a path through other nodes are removed for clarity.*

similar pairs. For the tasks annotated through AMT, the set of *similar pairs* is not just all the pairs outside the set of *consistent pairs*, *i.e.* the *inconsistent pairs*, as these may be noisy. To find the set of *similar pairs*, the directed graph introduced above is used. Separation between a pair of videos is defined as the difference in the length of the longest walk from any source node in the graph. A source node is a video which is not ranked lower than any other video. For instance, in Figure 3.4 D2 is the source node and the longest walk to H1 is 7. Pairs in the set of *inconsistent pairs* with a separation of 0 or 1 are taken as *similar pairs*. In the Chopstick Using task, H2 and H4 would be considered similar in skill, as would H2 and B3 (see Figure 3.4). Annotators are not asked to indicate pairs they believe are similar, as this would discourage them from distinguishing between closely ranked pairs and would result is a less fine-grained ranking.

Statistics on the tasks, and the *consistent* and *similar pairs* found for each, are available in Table 3.1. The Surgery tasks have a high number of consistent pairs as these pairs come from the scores of a single expert, available with the JIGSAWS dataset. Therefore, pairs are only excluded from the *consistent pairs* when two videos have the same score. For the other tasks, the judgements from multiple AMT workers are used. Dough Rolling has

[3]The letter indicates the participant which is followed by the trial number.

Task	Average Length (s)	Num. Videos	Maximum Pairs	% Consistent Pairs	% Similar Pairs	Total Pairs
Surgery (Knot Tying)	57± 20	36	630	95%	5%	100%
Surgery (Needle Passing)	108± 24	28	378	96%	4%	100%
Surgery (Suturing)	112± 43	39	701	95%	5%	100%
Dough Rolling	102± 29	33	528	34%	18%	52%
Drawing (Sonic)	86± 25	20	190	62%	37%	99%
Drawing (Hand)	117± 58	20	190	68%	26%	94%
Chopstick Using	46± 17	40	780	69%	10%	79%

Table 3.1: *The number of videos and number of potential pairs ($n(n-1)/2$) for each task, alongside the percentage of consistent pairs and similar pairs obtained through the annotation process.*

the lowest percentage of *consistent pairs* with annotators finding these videos harder to separate. The annotators found the Drawing and Chopstick Using tasks less ambiguous, with 60-70% of pairs ranked consistently and many of the *inconsistent pairs* found to be similar.

3.3 Naive Measures of Skill

This chapter proposes to rank skill from videos of people performing a task. This section considers the suitability of different forms of data to ranking skill, namely past experience, time of completion and the end result. Each of these are discussed in turn and the need to rank skill from video is motivated.

3.3.1 Experience

It could be argued that the amount of past experience a person has will indicate how skilled that person is at a task. While experience in a task will be heavily related to skill, there are more factors. People will improve at different rates and some people will be naturally much better than others or can draw on experience from other tasks. Experts may also occasionally make mistakes and thus could have a low-scoring trial.

This is evident in the Surgery tasks from the JIGSAWS dataset [48]. As explained in Chapter 2, this dataset contains eight participants each performing Knot Tying, Needle Passing and Suturing tasks five times using the da Vinci surgical system [54]. These participants are classified into 'Novice', 'Intermediate' and 'Expert' categories based on

	Knot Tying	Needle Passing	Suturing
ρ	0.61	-0.29	0.10

Table 3.2: *Spearman's rank correlation between past experience category and OSATS skill score for the tasks in the JIGSAWS dataset.*

their previous experience[4] with the system. The dataset also contains skill scores from an expert using the OSATS criteria [108].

The Spearman's rank correlation between these scores and the past experience of participants is shown in Table 3.2. This gives a score between 1 and -1, with 1 being perfect positive correlation, -1 perfect negative correlation and 0 no correlation. For Knot Tying the correlation is reasonably high, with intermediates and experts outperforming the majority of novices. However, more information is evidently needed even in the Knot Tying task, as intermediates tend to outperform experts and the highest score is obtained by a novice. For Suturing and Needle Passing, past experience has a much lower correlation with score and is even negatively correlated for Needle Passing.

To obtain finer-grained categorisation between performances from the same participant, it could be assumed that they improve between consecutive trials. This also utilises past experience, but only the recent experience of the exact task being performed. In the Surgery tasks, as well as in the Drawing and Chopstick Using tasks, each participant repeats the task five times. The correlations between each participant's trial numbers and skill ranking are shown for each of these tasks in Figure 3.5. For the Surgery tasks, Spearman's rank correlation is used as above, with 0 indicating no correlation. Since the tasks in EPIC-Skills have pairwise annotations and not a complete ranking, the relationship between trial number and skill is measured by calculating the number of human annotated pairs correctly classified when assuming a larger trial number means the participant will demonstrate higher skill. Note that with pairwise accuracy, random performance is 50%, with $< 50\%$ showing a negative correlation and $> 50\%$ showing a positive correlation.

From Figure 3.5 it is clear that while many participants do increase in skill across trials, many do not. All tasks have at least one participant with little correlation between trial number and skill ranking. In some tasks, such as Chopstick Using, some participants have a strong negative correlation, meaning their performance decreased over the trials. While in a task such as Drawing participants explore different ways to complete the task,

[4]Novices had 1-10 hours experience, Intermediates 10-100 hours and Experts had more than 100 hours.

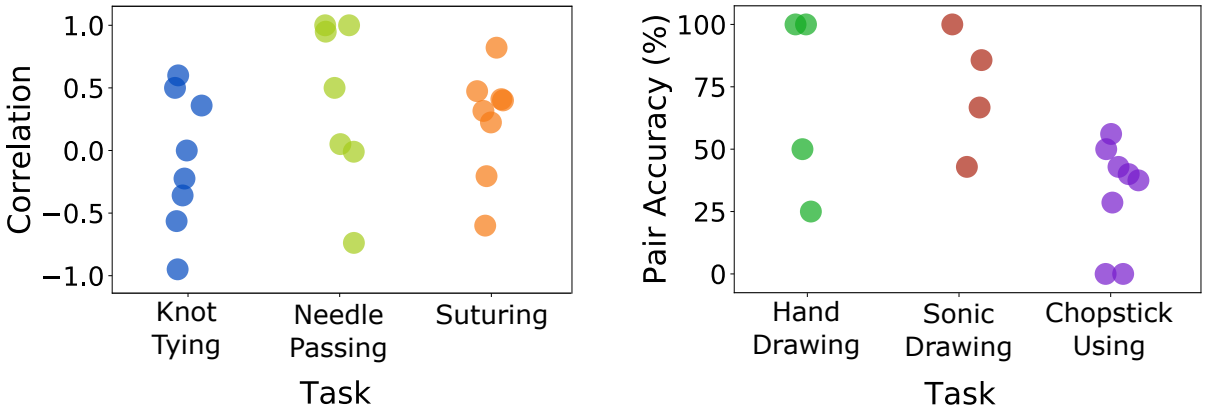

Figure 3.5: *Relationship between skill and trial number per participant. Left: Spearman's rank correlation between skill score and trial number for each participant in the Surgery tasks. Right: Average pairwise accuracy per participant in the Drawing and Chopstick Using tasks when assuming a later trial means the participant is more skilled.*

this exploration does not tend to occur in Chopstick Using. Instead, participants grip the chopsticks in the same way in each trial causing their performance to worsen as their hand becomes fatigued.

In Drawing, novices tend to improve in skill across the trials. This is not true for the other tasks. Some novices improve in the Surgery tasks, but others do not. On the other hand, all experts have a positive correlation between their OSATS skill score and trial number. This is likely because the experts only need a few trials to remember how to use the tools effectively, while novices are still discovering how to control the tools.

In summary, measures of both overall experience and immediate past experience do not correlate with annotations of skill for many tasks. Therefore, past experience is not a good indicator for the skill shown in a particular trial.

3.3.2 Time of Completion

Time of completion could be used as a naive measure of skill. However, someone who is able to complete a task quicker is not always better at the task. When two participants obtain the same result, the quicker participant will often be considered better at the task, however the speed is less important if one outcome is worse. Furthermore, for some tasks, speed will be much less relevant, or even inversely correlated. The correlation between time of completion and skill ranking is shown for all tasks in Table 3.3. As before, Spearman's rank correlation is used to measure the relationship between time of completion and skill score for the Surgery task, while pairwise accuracy is used for the tasks in EPIC-Skills.

	Knot Tying	Needle Passing	Suturing
ρ	0.724	0.211	0.342

	Dough Rolling	Sonic Drawing	Hand Drawing	Chopstick Using
Acc. (%)	85.6	30.5	15.5	96.5

Table 3.3: *Relationship between time of completion and skill. Left: Spearman's rank correlation (ρ) between the skill scores and the time taken to complete the Surgery tasks. Right: Percentage of annotated skill pairs correctly ranked when assuming faster means more skilled.*

Table 3.3 shows that for some tasks, such as Chopstick Using, time is a good measure of skill. Knot Tying, Dough Rolling and Hand Drawing also show a reasonably strong relationship between time and skill, although for Drawing the correlation is negative, meaning the slower performances of the task usually show higher skill. However, there is little relationship in the remaining tasks, indicating that time is unsuitable as a general method of skill determination.

3.3.3 End Result

Another possibility is using the end result of the task. While for many tasks the end result is important, in other tasks it can look very similar. Take the example of Suturing shown in Figure 3.6. There is little difference between the end result of the videos ranked first and third and while the stitching is somewhat misaligned with the guide dots in the second video, other parts of the video are much more informative. The ranking becomes more apparent when looking at the other frames. In the third video, larger holes are made during stitching and the material is often stretched. These differences can also be seen between the first and second ranked video, albeit to a lesser extent.

For other tasks, such as Drawing or Dough Rolling, the end result can be a better demonstration of skill (see Figures 3.7 and 3.8). However, as will be shown in Section 3.5.3, using only the last segment is less effective than using the full video for these tasks. There are many aspects not captured in the final result, such as the smoothness of motion or the method used to complete the task. In Drawing, how the participant starts can be an early indication of skill. Mistakes are also more obvious in the video than the end result and they can be partially corrected. This is the case in Dough Rolling, where participants may cover over holes in the dough. The Dough Rolling task highlights another issue with the end result, where poor lighting or bad camera angles may make it hard to obtain, such as in the lowest ranked video in Figure 3.8. Therefore, it is necessary to use more than just the end result to rank skill.

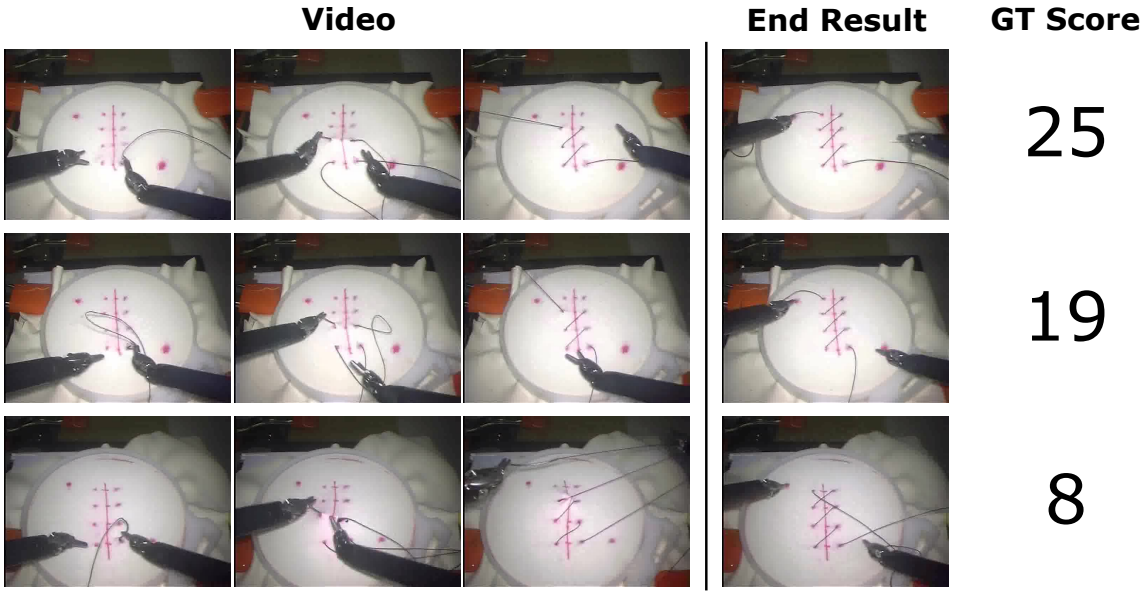

Figure 3.6: *Examples of three Suturing videos displaying the OSATS score and end result alongside informative segments in the earlier parts of the video.*

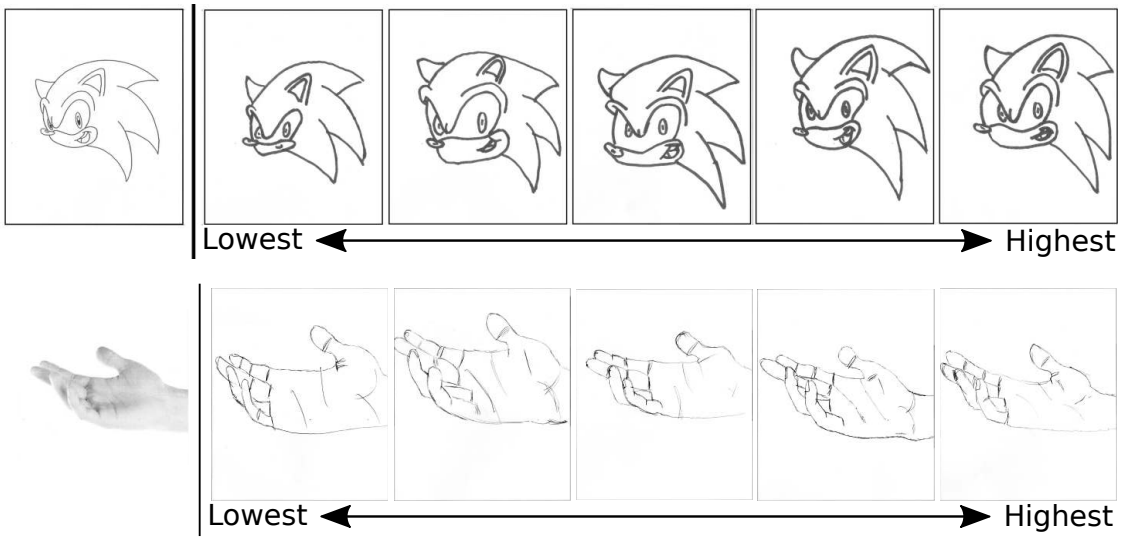

Figure 3.7: *The end result of the sonic drawing (top) and hand drawing (bottom) tasks ranked according the full-video ground-truth. The reference images given to participants are displayed on the left.*

After examining these alternative criteria for ranking skill it can be concluded that none of them are generally applicable to many tasks. While some metrics are useful for certain tasks, such as time of completion in Chopstick Using, in other tasks they are poorly correlated with the human annotations. The criteria used by the annotators is more nuanced and weighs each of the discussed metrics differently for the distinct tasks, while also taking into account extra information which can only be seen in the video. Using the video to determine skill will also allow estimates of skill to be obtained

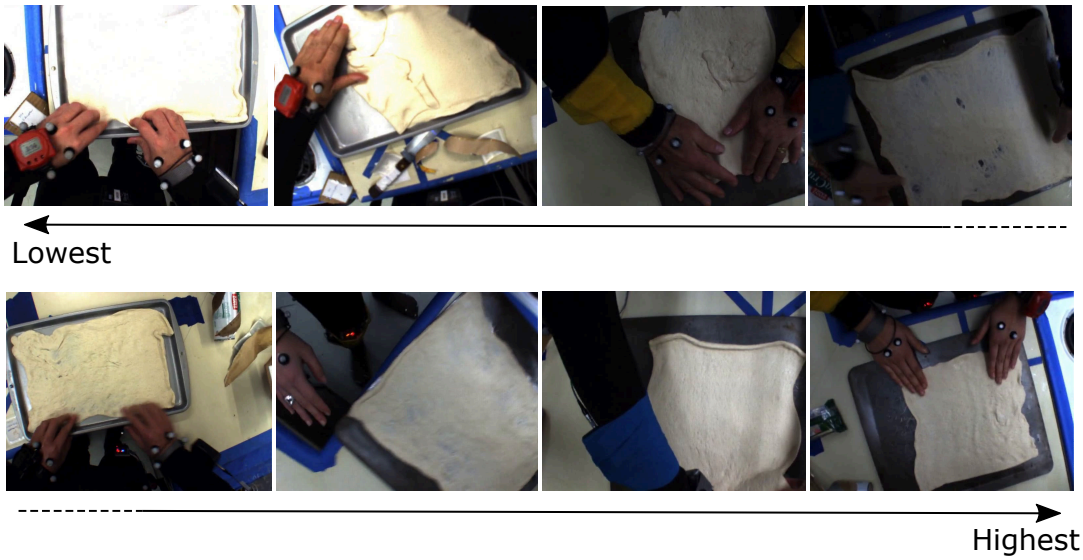

Figure 3.8: *The end result of the Dough Rolling task ranked according to ground-truth pairwise annotations.*

mid-task, enabling future work on intervention and feedback. Therefore, this chapter learns to rank skill from video for each of the different tasks. The proposed method is explained in the following section.

3.4 Pairwise Deep Ranking for Skill Determination

This section first gives an overview of the skill determination problem (Section 3.4.1) and the base network architecture (Section 3.4.2). Section 3.4.3 then describes how this network architecture is adapted for the task of skill ranking. The proposed additions to the loss function are then explained in Sections 3.4.4 and 3.4.5. Finally, Section 3.4.6 describes how the method is used to determine skill for test videos.

3.4.1 Problem Definition

The goal is to be able to learn models to rank skill in different tasks. Within a single task, there is a set of K videos $X = \{x_k, 1 \leq k \leq K\}$, from multiple people. Each video is considered independently, despite some tasks having the same participant in multiple videos. As seen in Section 3.3, people can differ in the skill they display with each instance of the task. Thus, the aim is to obtain a relative ranking of skill per video instead of accumulating a score per person. The relative skill between two videos x_i and

x_j is defined with the following function:

$$E(x_i, x_j) = \begin{cases} 1 & x_i \text{ shows higher skill than } x_j \\ -1 & x_j \text{ shows higher skill than } x_i \\ 0 & \text{no skill preference} \end{cases} \tag{3.1}$$

Note that according to Equation 3.1, $E(x_i, x_j) = -E(x_j, x_i)$. Therefore, only one annotation is needed per pair and pairs in the set $\{(x_i, x_j); E(x_i, x_j) = -1\}$ are not considered. These values are obtained from the annotation process described in Section 3.2). The *consistent pairs* are the set $P = \{(x_i, x_j); E(x_i, x_j) = 1\}$ and the *similar pair* are the set $\Phi = \{(x_i, x_j); E(x_i, x_j) = 0\}$.

3.4.2 Temporal Segment Network Architecture

Tasks differ in how skill can be displayed. There are two main sources of relevant information within videos. First is the quality and types of motion used, *i.e.* the actions used to complete the task and how well these actions are performed. Second is the effect on the environment, *i.e.* the final result and the results at intermediate stages of the task. The latter can be captured through RGB images contained within the video, while optical flow can be used to focus specifically on the motions used and the quality of these motions. Thus, a two stream convolutional neural network is used to learn to determine skill from videos. Specifically, the proposed method is based on Temporal Segment Networks (TSN) [194] with spatial and temporal streams (see Chapter 2). TSN was selected due to its state-of-the-art performance on action recognition at the time.

Each stream of the TSN is trained separately. As in [194], input videos are uniformly divided into three segments. From each segment a 'snippet' is randomly sampled. This snippet consists of either a single RGB frame (for the spatial stream) or a short sequence of horizontal and vertical optical flow frames (for the temporal stream). The TSN produces a prediction per snippet which are combined to obtain the final prediction for an input video. The authors of TSN [194] find average pooling to be the most effective combination method for action recognition. This chapter also chooses average pooling for skill determination, however other methods are tested in Section 3.5.

3.4.3 Pairwise Deep Ranking

The proposed method utilises the pairwise approach for 'learning to rank'. To do this, a Siamese version of the two-stream TSN is used, where weights are shared across both

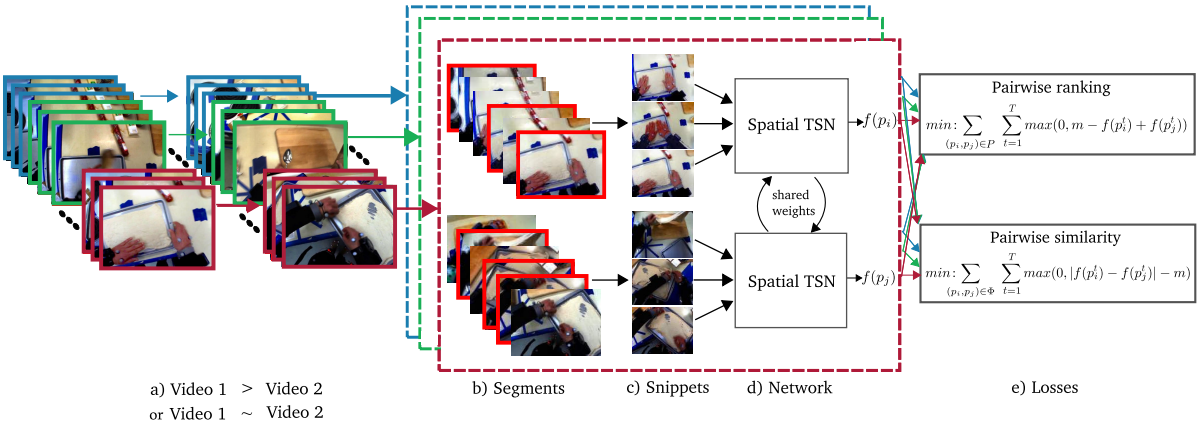

Figure 3.9: *Learning to determine skill. a) All pairs of videos are considered where the first shows a higher level of skill (P) or their skill is comparable (Φ). These videos are divided into T splits to make use of the entire video sequence. b) Each split in the pair is then divided up into 3 equally sized segments as in [194]. c) TSN selects a snippet randomly from each segment. For the spatial network this is a single frame, for the temporal network this is a stack of 5 dense horizontal and vertical flow frames. d) Each snippet is fed into a Siamese architecture of shared weights, for both spatial and temporal streams, of which only the spatial is shown here. e) The score from each video split is given to one of the proposed loss functions: ranking or similarity, depending on the ground-truth.*

sides of the Siamese network. An overview of the proposed network for the spatial stream can be seen in Figure 3.9. The method takes a pair of videos as input, where one video is ranked higher than the other or videos are similarly ranked (Figure 3.9a). These videos are partitioned into multiple splits to make use of the entire video. Each input video split is then further divided into segments (Figure 3.9b) as in TSN [194] and single frame snippets are sampled from each segment (Figure 3.9c). The corresponding snippets sampled from each video are then fed into separate, but identical, TSNs which form the Siamese network (Figure 3.9d). Each side of the Siamese network then outputs a score for its input video which is evaluated by the relevant loss function (Figure 3.9e).

Note that the method aims to rank videos rather than score them according to a predefined scale. Therefore, the predicted score only has meaning in relation to the predicted scores of other videos. Given a pair of videos, where the first is ranked higher than the second in terms of skill, the Siamese network should output a higher score for the first. Formally, there is a set of video pairs $P = \{(x_i, x_j); E(x_i, x_j) = 1\}$ (see Equation 3.1). Let the TSN be represented by the function $f(\cdot)$, where the output for a video x_i is $f(x_i)$. The goal is to learn the function f such that the predicted scores can be used to

rank skill. Specifically:

$$f(x_i) > f(x_j) \qquad \forall (x_i, x_j) \in P \tag{3.2}$$

To learn this function f, a margin loss layer is used to evaluate the ranking of each pair of videos in P. This loss function is an approximation to the 0-1 ranking error loss and has successfully been used for other ranking applications [190, 208]:

$$L_{rank1} = \sum_{(x_i,x_j)\in P} \max(0, m - f(x_i) + f(x_j)) \tag{3.3}$$

With this loss function, the derivatives for the individual examples are:

$$\frac{\partial L_{rank1}}{\partial f(x_i)} = \begin{cases} -1 & f(x_i) - m < f(x_j) \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial L_{rank1}}{\partial f(x_j)} = \begin{cases} 1 & f(x_i) - m < f(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The loss function aims to ensure the score of video x_i is greater than the score of video x_j by at least the margin m. This encourages the network to learn discriminative features to distinguish between the amount of skill displayed in different videos.

3.4.4 Pairwise Deep Ranking with Splits

Traditionally, two stream CNNs are used for action recognition [42, 166, 194] where videos are short and the whole length of the video may need to be considered to recognise the action. This work examines skill, which this chapter assumes could be understood from all (or any) parts of the video. To make the most of the extent of the video, T uniform splits are considered (Figure 3.9a). This method makes the assumption that two videos of the same task have comparable rates of progression through the task, and thus compares the temporal splits across a pair of videos in order. Assuming x_i^t is the t^{th} split of video x_i, the skill annotations are extended such that,

$$E(x_i^t, x_j^t) = E(x_i, x_j) \qquad \forall t = 1, ..., T \tag{3.5}$$

The ranking loss function now becomes:

$$L_{rank2} = \sum_{(x_i,x_j)\in P} \sum_{t=1}^{T} \max(0, m - f(x_i^t) + f(x_j^t)) \tag{3.6}$$

Pairing corresponding splits allows the two videos to be compared at a similar stage of task performance, while still being able to deal with videos of different lengths. For instance, in the Dough Rolling task the last segment of each video is likely to contain the stretching of the dough, while the first will contain the opening of the dough container. It makes more sense to compare the dough stretching in one video to the dough stretching in another, rather than compare the opening of the container in one to the dough stretching in another. More discriminative features are likely to be learned in the proposed way.

3.4.5 Pairwise Deep Ranking with Similarity Loss

The margin loss function in Equation 3.3 only incorporates pairs where one video is consistently ranked higher than another. This section proposes a second loss which aims to utilise similarly ranked pairs to learn commonalities between these videos. The margin loss is modified to learn features which map pairs, indistinguishable in terms of skill, to similar scores. This uses the set of pairs $\Phi = \{(x_i, x_j); E(x_i, x_j) = 0\}$ (see Equation 3.1).

As opposed to making the score of one video greater than another by margin m, the new loss aims to get the difference of the scores of similarly ranked videos within margin m:

$$|f(x_i) - f(x_j)| \leq m \equiv |f(x_i) - f(x_j)| - m \leq 0 \tag{3.7}$$

Therefore, the new loss function for similarly ranked pairs becomes:

$$L_{sim} = \sum_{(x_i,x_j)\in P} \sum_{t=1}^{T} \max(0, |f(x_i^t) - f(x_j^t)| - m) \tag{3.8}$$

This loss has the following gradients, providing a margin $m > 0$ is selected:

$$\frac{\partial L_{sim}}{\partial f(x_i)} = \begin{cases} \frac{f(x_i)-f(x_j)}{|f(x_i)-f(x_j)|} & |f(x_i) - f(x_j)| > m \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial L_{sim}}{\partial f(x_j)} = \begin{cases} \frac{f(x_j)-f(x_i)}{|f(x_i)-f(x_j)|} & |f(x_i) - f(x_j)| > m \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

which simplifies to:

$$\frac{\partial L_{sim}}{\partial f(x_i)} = \begin{cases} 1 & f(x_i) > f(x_j) + m \\ -1 & f(x_i) < f(x_j) - m \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial L_{sim}}{\partial f(x_j)} = \begin{cases} -1 & f(x_i) > f(x_j) + m \\ 1 & f(x_i) < f(x_j) - m \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The above equations show that this loss encourages the set of *similar pairs* Φ to have similar skill scores. The overall loss function then becomes:

$$L_{rank3} = \beta L_{rank2} + (1 - \beta) L_{sim} \tag{3.11}$$

Adding L_{sim} to the ranking loss not only allows extra data to be utilised in the learning process, but also encourages the network to learn features which indicate similarities in skill between similarly ranked videos.

3.4.6 Evaluating Skill for a Test Video

Following training, the learned two-stream CNN is used to evaluate skill for test videos of the same task. In testing, σ snippets are uniformly sampled from each video x_i, again as in [194]. Each snippet x_i^t for $1 \leq t \leq \sigma$ is then fed into the spatial and temporal TSN independently. The output for each snippet is a score $f(x_i^t)$ for both the spatial $f_{spatial}(x_i^t)$ and temporal $f_{temporal}(x_i^t)$ streams. To fuse these two streams and obtain an overall score per video, the weighted average is taken:

$$f(x_i) = \frac{1}{\sigma} \sum_{t=1}^{\sigma} \lambda f_{spatial}(x_i^t) + (1 - \lambda) f_{temporal}(x_i^t) \tag{3.12}$$

where λ is the fusion weighting between spatial and temporal information and σ is the number of snippets in testing.

An overall ranking for a set of videos can be obtained by ordering all videos in descending order based on $f(x_i)$.

3.5 Experiments and Results

This section presents results of experiments which test the proposed method on the Drawing, Dough Rolling and Chopstick Using tasks described in Section 3.2 as well as the

Surgery tasks from the JIGSAWS dataset (described in Chapter 2). First Sections 3.5.1 and 3.5.2 describe the implementation details and the evaluation metric respectively. The different components of the proposed method are then ablated in Section 3.5.3. Section 3.5.4 then compares the proposed method to baselines and qualitative results are presented in Section 3.5.5.

3.5.1 Implementation Details

Dataset Splits. For all tasks, four-fold cross validation is used to report results. For each fold, the pairs between three quarters of the videos are used in training and the remaining pairs are used for testing. This includes pairs where neither video has been seen in training and pairs where one video has been used in training within a different pairing as it is important to evaluate whether the unseen videos are correctly placed within a known ranking. Networks are learnt separately for the different tasks, although the three Surgery tasks from JIGSAWS are trained together, as are the two Drawing tasks. This chapter makes the assumption that these sets of tasks contain similar skill-relevant features and therefore can be learnt together to reduce training time. This assumption will be further explored in Chapter 5. While each task is trained separately, the hyperparameters detailed in the sections below are the same for all tasks.

Network Input. Videos are first resized to 340×256 pixels. To avoid over-fitting, the same data augmentation techniques as in the original TSN network [194] are used, namely horizontal flipping, corner cropping and scale jittering. The cropped regions are 224×224 pixels. To extract optical flow frames for the temporal stream of the two-stream CNN, the TV-L^1 algorithm [211] is used.

Architecture Details. Both the spatial and temporal networks use AlexNet [82] as this gave better results with a shorter training time than the BN-Inception [65] originally used in TSN. Dropout, with a ratio of 0.5, is used before the first and second fully connected layers of both the spatial and temporal networks. Both networks are initialised with weights from pre-trained ImageNet [32] models[5]. The weighting parameter β between L_{sim} and L_{rank2} in Equation 3.11 is set to 0.5 in all experiments. $T = 7$ video splits are used in training and $\sigma = 25$ snippets are used in testing unless otherwise specified (as in [194]). All loss functions use margin $m = 1$. The fusion weighting between spatial and temporal streams is $\lambda = 0.4$ (Equation 3.12) as in [194], however the sensitivity of the results to this parameter is tested in Section 3.5.3

[5]The work in this chapter was carried out in 2017, before pre-training on large-scale videos datasets such as Kinetics became available.

Training Details. The model parameters are learnt using mini-batch stochastic gradient descent with a batch size of 128 and momentum of 0.9. In the spatial network, the learning rate begins at $1e^{-3}$ and decreases by a factor of 10 every 1.5K iterations, with the learning process finishing after 3.5K iterations. The temporal network's learning rate is initialised as $5e^{-3}$, decreasing by a factor of 10 after 10K and 16K iterations, with learning ending at 18K iterations.

3.5.2 Evaluation Metric

To evaluate the proposed method, pairwise accuracy is used. Pairwise accuracy is defined as the *percentage of correctly order pairs*. A pair is correctly ordered if the method outputs $f(x_i) > f(x_j)$ for a pair (x_i, x_j) where $E(x_i, x_j) = 1$ in the ground-truth. The task annotations may not have a complete ranking as ground-truth (as explained in Section 3.2). Pairwise accuracy allows the method to be evaluated on these tasks.

3.5.3 Ablation Study

Table 3.4 shows the results of four-fold cross validation with each loss function on each of the four types of tasks. This table shows the proposed loss function L_{rank3} outperforms the standard margin loss function L_{rank1} on almost all combinations of modality and task. An improvement over L_{rank2} can also be seen for all but the temporal results in Surgery and Dough Rolling. This improvement is particularly noticeable in the two-stream results for Drawing (79.1% to 83.2%) and Chopstick Using (68.8% to 71.5%). From the improvement of L_{rank3} over L_{rank2}, it can be concluded that the inclusion of similar pairs with L_{sim} is effective. This has the largest impact on the spatial network, where the results for skill determination are generally lower. L_{sim} increase the performance of the spatial network causing improvement in the two-stream result.

From Table 3.4 it can also be concluded that the temporal network is generally more useful for determining skill in the presented tasks, with the temporal result outperforming the spatial result in all but the Chopstick Using task. Although, this gap is reduced with the inclusion of L_{sim} (Equation 3.11). This implies the motions performed are more important for determining skill than the current state or appearance of the task (captured in the spatial stream). However, the spatial network is beneficial to the final two-stream result in both Drawing and Chopstick Using.

Loss	Surgery			Dough Rolling			Drawing			Chopstick Using		
	S	T	TS	S	T	TS	S	T	TS	S	T	TS
L_{rank1}	64.7	72.8	69.1	77.6	79.4	78.5	75.6	77.4	78.0	67.2	67.9	68.8
L_{rank2}	64.4	**73.3**	69.0	79.1	**80.4**	78.5	74.9	81.8	79.1	67.2	69.9	68.8
L_{rank3}	**66.4**	72.5	**70.2**	**79.5**	79.5	**79.4**	**77.6**	**82.7**	**83.2**	**70.8**	**70.6**	**71.5**

Table 3.4: *Results of four-fold cross validation on all tasks, for the proposed method with each of the proposed loss functions. For all tasks, L_{rank3} outperforms the original loss L_{rank1}. S=Spatial, T=Temporal, TS=Two-Stream*

The largest difference between the spatial and temporal network is in Surgery. These tasks require quick, smooth motions in order to put minimal stress on the surrounding areas. While the end result of each stage is visually similar, the motions affect the scoring significantly. Surprisingly, the temporal stream also has a large improvement over the spatial stream in Drawing. Although the end result of the Drawing can be seen in the spatial stream, the motions used to create the drawing are a key indicator of skill, this further demonstrates the need for determining skill from the full video.

In Dough Rolling the performance of the two streams are similar. While the motions are important here, there are also several key features which are better seen from the spatial stream, such as holes in the dough. The performance of the two streams are also similar in Chopstick Using, where both streams can be used to identify whether a bean has been successfully picked up and moved. However, the two-stream result improves over either stream independently which demonstrates the features found in each stream are complementary.

Fusion Hyperparameter. The trade-off between the spatial and temporal streams is further examined in Figure 3.10. This figure displays results for all tasks with different values of λ (Equation 3.12) from 0 to 1 at intervals of 0.1. For the majority of tasks, the combination of temporal and spatial modalities is useful, except in the Surgery tasks which peaks at $\lambda = 0$, *i.e.* when no information from the spatial network is included. All tasks benefit from the contribution of the temporal network, albeit to different degrees. Figure 3.10 also demonstrates that the method is quite resilient to the exact value of λ chosen, with $\lambda \in [0.4, 0.7]$ being reasonable choices for the majority of tasks.

Consensus Function. Thus far, the consensus score between the multiple snippets used in TSN has been formed from the mean of the individual snippet scores. This is the same as used in TSN [194] for action recognition. However, this may not be the best choice for skill determination. Skill might be better measured by looking at the highest

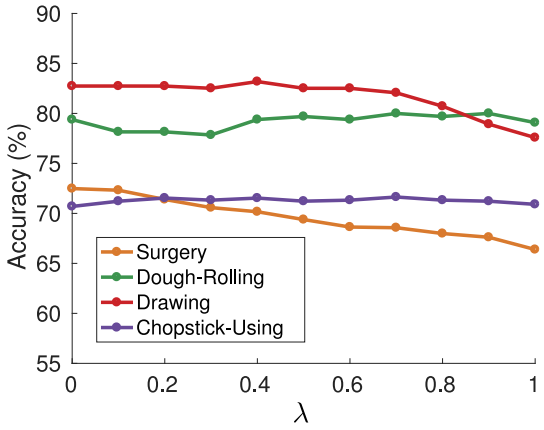

Figure 3.10: *The accuracy for each set of tasks with different λ values. The method is resilient to the value chosen.*

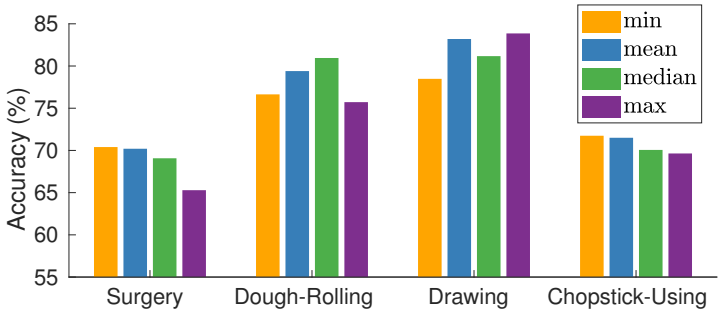

Figure 3.11: *The accuracy per task of the Siamese two-stream CNN with different consensus functions in the TSN.*

or lowest levels of skill displayed in the video. Therefore, different consensus functions: mean, minimum, maximum and median are tested in Figure 3.11.

Figure 3.11 demonstrates that the best consensus function is different for each task. This highlights the diversity between these tasks and demonstrates the challenge in learning to determine skill for a large variety of tasks. Minimum performs well for both Surgery and Chopstick Using. This makes sense, in surgical tasks the severity of the mistakes, or the lack of any mistakes, is critical to measuring the success. This is similarly true for Chopstick Using, where dropping the bean or fumbling to pick it up are key indicators of low-skill. Interestingly, in Drawing the reverse is true, where the most successful consensus function is the maximum. In Dough Rolling, the median and mean are the two best consensus functions. Failures are much less critical in this task, and how failures, such as holes in the dough, are recovered from can be an indication of skill.

Overall, the mean consensus function does obtain the best result as it is the second best aggregation function for each task.

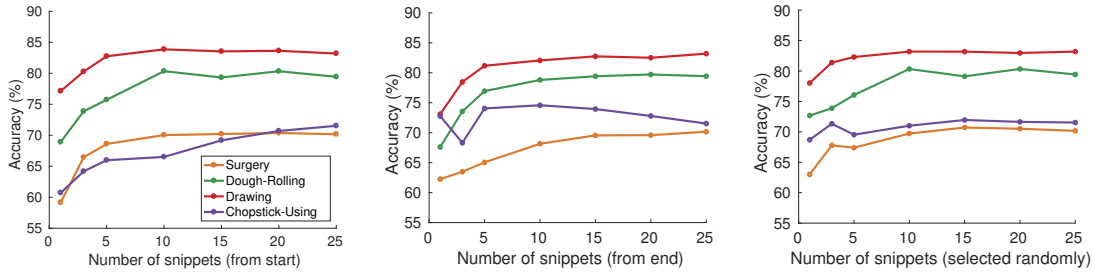

Figure 3.12: *The accuracy achieved when adding snippets in testing, from the start (left), the end (middle) and randomly (right)*

Number of Snippets in Testing. Up to this point, results have been reported using $\sigma = 25$ uniformly sampled snippets in testing (Equation 3.12) in line with previous methods [194]. However, it is interesting to examine how much of a video is needed at test time to gain an accurate evaluation of the skill displayed. The number of snippets and where in the video these snippets are sampled from is tested in Figure 3.12.

Figure 3.12 shows results per task when adding consecutive snippets from the start of the video (left), the end of the video (middle) and from random locations in the video (right). It is evident that good accuracy can be obtained after only seeing a portion of the video, from either the beginning or the end. However, a single snippet is insufficient to measure skill, even if this snippet comes from the end result of the task.

Accuracy converges for the majority of tasks as the number of snippets continues to increase, indicating denser sampling is not needed. For instance, the Surgery tasks achieve near peak accuracy with the snippets sampled from the first 20% of the video, while the latter 20% of snippets appear to be redundant and accuracy increases much slower when starting from the end of the video. This difference is intuitive, as the start of the Surgery tasks are more challenging: due to the repetitive nature of the task novice participants can improve by the end.

The Chopstick Using task performs best when using snippets from the end of the video. This task also has a much more linear increase in performance when snippets are first sampled from the start of a video. This is because the number of beans the participants have successfully managed to pick up can often be counted in the final snippet. With many segments from earlier in the video, this information is diluted and performance is slightly decreased. However, best performance still comes with 10 snippets sampled from the end of the video, meaning a combination of the performance of the task and end result is best. This task highlights the need for future work to identify important frames, both to test with and to learn from, as performance for Chopstick Using degrades as more snippets are added from the beginning of the video.

Method	Surgery	Dough Rolling	Drawing	Chopstick Using
RankSVM [76]	65.2	72.0	71.5	**76.6**
Yao *et al.* [208]	66.1	78.1	72.0	70.3
Siamese TSN with L_{rank1}	69.1	78.5	78.0	68.8
Siamese TSN with L_{rank3}	**70.2**	**79.4**	**83.2**	71.5

Table 3.5: *Results of four-fold cross validation on all tasks, for the baselines and the proposed method with L_{rank1} and L_{rank3}*

3.5.4 Comparison to Baselines

As outlined in Chapter 2, there are no generic prior methods for ranking skill, nor for performing skill determination for non-surgical and non-sporting tasks. For completeness, a method for determining skill in surgery is compared to in Appendix A. However, this thesis focuses on methods generally applicable to a variety of tasks so existing ranking methods, developed for other applications, are used as baselines here.

The first baseline uses the method of **Yao *et al.*** [208] which was originally proposed for highlight detection in video. Although this method was developed for a different purpose, it uses a general method for ranking video which is not specific to highlight detection. First, features are extracted per frame from pre-trained networks. These are then averaged to obtain a single feature vector per video. These features are then passed into a Siamese network with an architecture of six fully connected layers: $F1000$-$F512$-$F256$-$F128$-$F64$-$F1$, where F indicates a fully connected layer and the number is the output dimension. This network uses the same margin ranking loss as in Equation 3.3 (L_{rank1}). Similarly to the proposed Siamese TSN, a two-stream network with late fusion is used by Yao *et al.* with $\lambda = 0.4$ (as in Equation 3.12). AlexNet [82] features pre-trained on ImageNet [32] are used for the spatial network. The temporal network uses features from C3D [182] trained on Sports1M [78].

The second baseline uses **RankSVM** [75], commonly used for ranking problems. This method also uses pairwise ranking to learn an overall ranking function, however an SVM is instead used to learn this ranking function. The implementation from [76] was used. The same features as in the above baseline from Yao *et al.* were used as input to the SVM. Similar to other works [208, 210], these were found to be better representations than hand-crafted features.

Comparative results are available in Table 3.5. As in the proposed method, four-fold cross validation is used to report baseline results. The proposed method outperforms

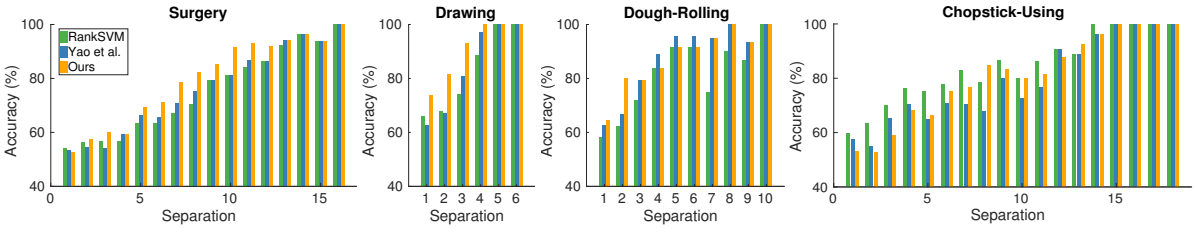

Figure 3.13: *The accuracy of each task by ranking separation between pairs of videos. The accuracy consistently increases as pairs are further apart.*

both baselines on three of the four tasks. RankSVM performs best on Chopstick Using. Chopstick Using is a relatively simple task with little visual variation between instances, therefore it is likely deep learning methods such as the proposed approach and the approach of Yao *et al.* over-fit to the data.

For the other three tasks, Yao *et al.* is the better performing baseline. The main differences between the proposed method and this baseline are the use of extracted features, dense sampling of frames and the proposed loss functions. The results of the proposed network with the same loss function as used in Yao *et al.* are also shown in Table 3.5. This demonstrates that the dense sampling of frames used by Yao *et al.* does not offer an advantage over the sparse sampling in TSN, at least not for the features used by Yao *et al.*. The most significant improvement with the proposed method is in the Drawing task, which benefits equally from the choice of architecture in the proposed method (+6.0%) and the proposed loss function (+5.2%).

To study where the difference in performance lies, the accuracy for each level of separation between the pairs is shown in Figure 3.13. Assume there exists annotations between pairs of videos which results in the partial ranking $x_i > x_{i+1} > ... > x_{i+n} > x_j$, then the separation between x_i and x_j is defined as $n+1$. It is more important that pairs with high separation be correctly ordered than pairs close in the ranking, although close together pairs will be harder to distinguish between. Figure 3.13 shows that the improvement of the proposed method over baselines comes from the mid-level of separation in Surgery and Drawing. Although all methods approach 100% for the most separated pairs, the proposed method approaches this much faster. In the Chopstick Using task, where the proposed method performs below the RankSVM baseline, the performance is comparable at medium and high separation, only falling below for pairs close in the ranking.

3.5.5 Qualitative Results

Output rankings produced by the proposed method are shown in Figure 3.14. The errors (highlighted with red arrows) tend to be between neighbouring pairs. In Dough Rolling,

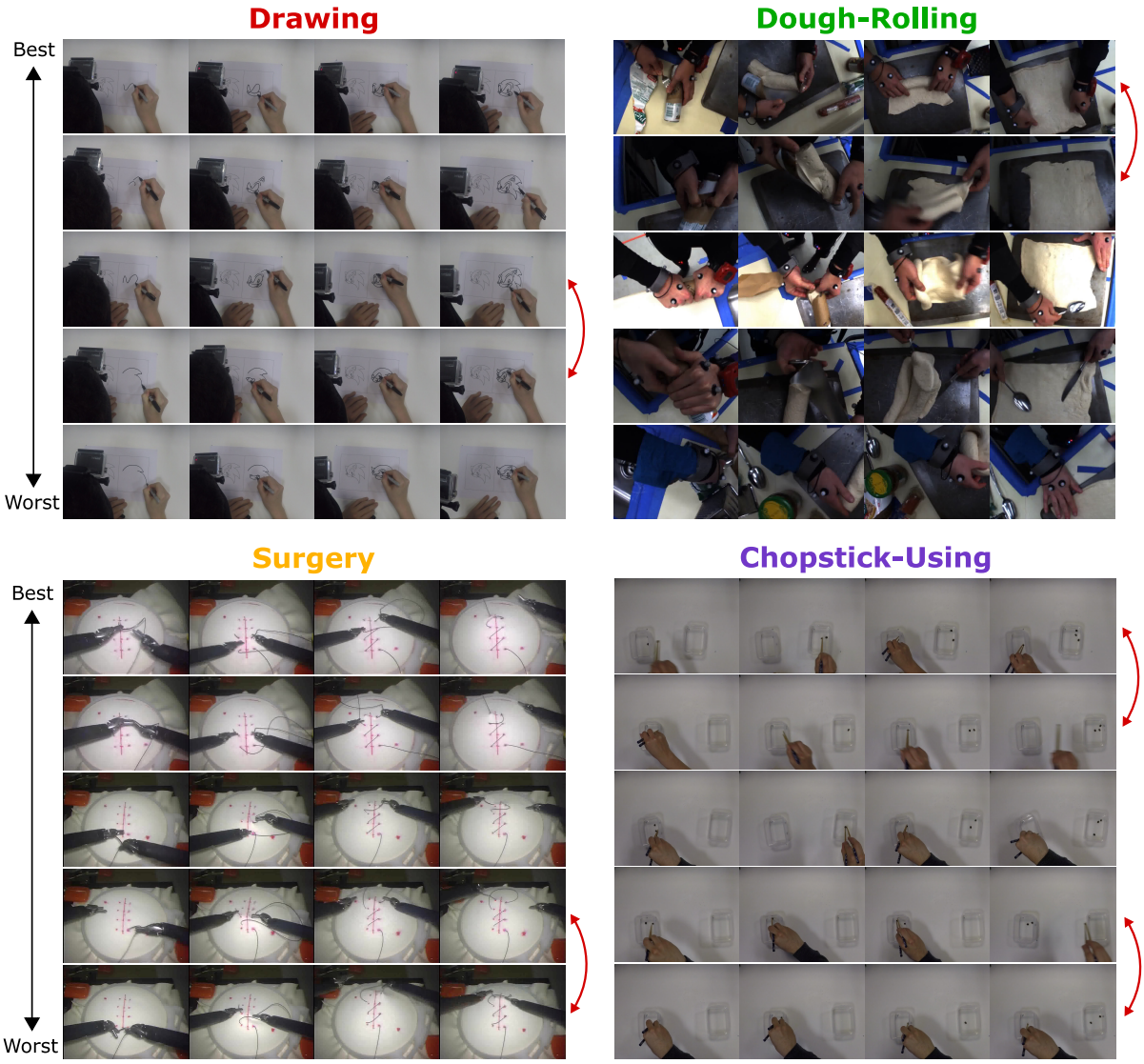

Figure 3.14: *Example rankings produced by the proposed method are shown for the four tasks. Wrongly ordered pairs are highlighted with the red arrows.*

successful participants manage to remove the dough from the can without creating holes and unroll the dough neatly. Participants from higher ranked videos in Drawing produce drawings which more closely resemble the reference image. Lower ranked videos in the Suturing Surgery task tend to not sew through the guide dots and put strain on the surrounding tissue. In Chopstick Using, how the chopsticks are held and how many beans are moved are key factors which determine the rank of a video[6].

A key difficulty in skill determination is capturing the nuance of the tasks. Figure 3.15 visualises top-down attention of the spatial CNN on example frames from each task. This

[6]These results are better viewed in video form: https://www.youtube.com/watch?v=R3QoZ-FltUQ

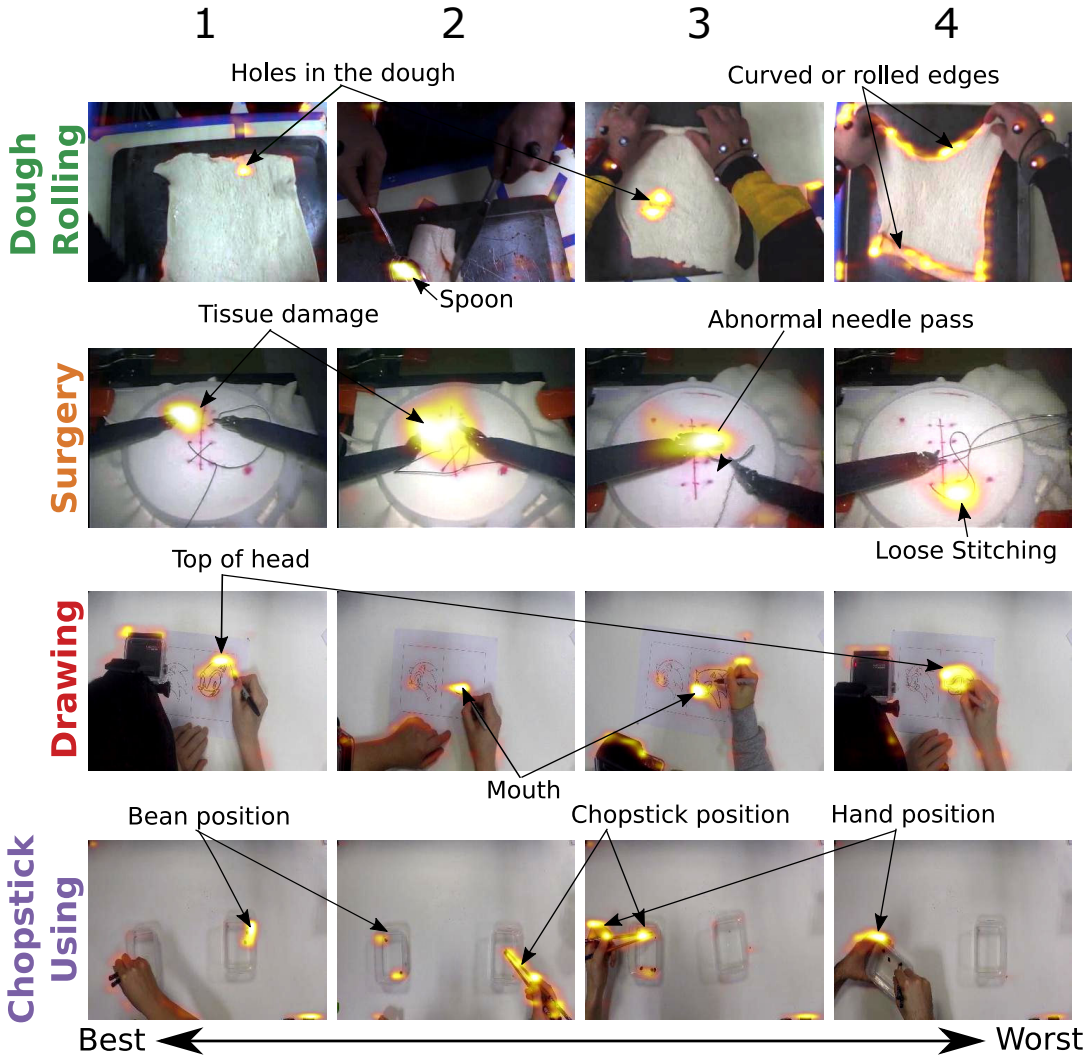

Figure 3.15: *Spatial activations for sample frames at varying ranks.*

uses the implementation of [148] based on excitation backpropagation [214]. For each task, frame-level spatial activations are shown on four videos with varying skill.

Figure 3.15 shows that the model selects details which correspond to what a human would pay attention to. In Dough Rolling high activations occur on holes in the dough (1,3), curved or rolled edges (4) and when using a spoon to spread the dough (2). High activations occur in Surgery when strain is put on the surrounding material (1,2), with abnormal needle passes (3) and when there is loose stitching (4). In Drawing, the model attends to specific parts of the sketch, such as the head (1,4) and mouth (2,3). The high activations in the Chopstick Using task occur on the hand position (3,4), chopstick position (2) and the bean locations (1,2,3). There are also some seemingly erroneous activations such as attending to the head mounted camera in Drawing and Chopstick Using or the hand markers in Dough Rolling.

3.6 Conclusion

This chapter has introduced the problem of determining skill in videos of daily tasks. Pairwise skill ranking annotations were collected for several of these tasks and a method which learns to rank skill from pairwise video annotations was introduced. Unlike prior work, this method is not specific to a particular task, or group of tasks, and can learn discriminative features for each task with a reasonably small amount of data. However, there are several limitations of the current method which highlight potential avenues for future work.

Equal importance of video parts. This work made the assumption that skill can be determined from any or all parts of a video. However, the results indicated that some video parts may be more important than others when determining skill. For example, the later parts of the video are more informative than the beginning in Chopstick Using. The spatial attention results demonstrated that the method can focus on key regions of the frame, however with the current method this cannot be done temporally. In Suturing there are several parts of the action which are indicative of skill, such as pulling the stitching tight. With the sparse sampling of frames, the method will rarely sees these most informative parts. A method which uses temporal attention to identify the video segments most relevant to determining skill would be a necessary extension to deal with longer and more complex tasks. It could also make skill determination models more explainable and therefore more appropriate for feedback systems. Chapter 4 will explore how to learn the most important video parts for skill determination.

Limited data. This chapter has presented the first dataset for skill determination of daily-living tasks, however the EPIC-Skills dataset is somewhat limited in scope. There are many more daily-living tasks which require different skills to the tasks presented in this chapter. It would be necessary to test many more tasks to confirm the method can rank skill in the wide array of tasks covered under 'daily living'. The amount of data per task is also limited. While there is a reasonable number of ranked pairs per task, the number of videos of each task is small. With this data, the method is somewhat prone to over-fitting. Additionally, the fact that dividing the videos into splits with L_{rank2} improves the results, despite evidence that some video parts are more important to determining skill, highlights the method's need for a larger amount of data.

The data is also limited by the environment it is recorded in; each of the takes within EPIC-Skills only takes place in one setting. A large amount of research into domain adaptation [24, 120, 122, 184] has shown that CNNs do not generalise well to new environments unseen in testing. To enable new videos of a task to be accurately ranked,

the model should be trained on more varied data with different backgrounds and more diverse ways of completing the task. Further data will be presented in Chapter 4 which alleviates some of these issues.

Transferable skill. This chapter made the assumption that related tasks (*i.e.* the three Surgery and two Drawing task) could be trained together as they may share features useful for determining skill. Further work is needed to investigate whether this is the case and to what extent tasks need to be similar to share skill relevant features. The current approach of training one network per task (or group of tasks) and requiring pairwise annotations of every task is not scalable. Chapter 5 will explore this problem, however there are many different directions for future work in this area. One possibility is few-shot learning for skill determination, where a ranking for a new task could be learnt from a small number of ranked pairs in combination with existing pair annotations (or learnt rankings) for other tasks.

Task-specific versus general skill determination. The method proposed in this chapter is more successful than prior ranking methods for skill determination in daily tasks. However, methods specific to a certain task, or group of tasks, are able to utilise prior knowledge and therefore have the potential to be more successful in these tasks. Further work is needed to bridge this gap. This could either be through methods which allow learning more fine-grained features specific to a task, or by automatically incorporating external sources of knowledge to learn what it means to be skilled in specific tasks. Chapter 6 will take the latter approach by exploring how instructional videos can be used to automatically assess whether individual actions have been performed in accordance with instructions.

Chapter 4

Attending to Skill-Relevant Video Parts

Chapter 3 proposed the first method to assess skill from videos of daily tasks. Results show that the proposed method is applicable to a variety of different tasks and outperforms existing ranking baselines. However, the the method is limited by the assumption that skill can be determined from any and all parts of the video. The results in Chapter 3 showed that this was not necessarily the case; in Chopstick Using a better accuracy was obtained when only using the end of the video. Many other tasks will also have certain parts or actions which are more indicative of skill than others. For instance, in cooking tasks, collecting ingredients from the fridge is unlikely to demonstrate skill, whilst the way in which ingredients are prepared may be key.

This chapter removes the assumption that skill can be determined from any video part and instead learns which video parts are most informative to skill. Illustrations of the spatial activations in Chapter 3 demonstrated that the Siamese CNN was able to attend to informative parts within a frame, therefore this chapter focuses solely on temporal attention. This is done in two parts. First, this chapter explores how existing temporal attention mechanisms can be incorporated into a Siamese ranking framework and how to stabilise the training of this attention. Second, a loss term is proposed which makes the attention 'rank-aware' and able to attend to video parts particularly indicative of either high or low skill.

This chapter also proposes a second skill determination dataset. While EPIC-Skills allowed skill determination to be explored across various daily-living tasks, it is somewhat limited by the amount of data per task. In this chapter, the BEST dataset is presented. This has an increased number of ranked pairs and videos per task, as well as diversity in environment and viewpoint and an increase in task complexity and length. The

BEST dataset allows skill determination to be explored for further daily-living tasks and increases the challenge for a method which aims to locate the relevant video parts.

The chapter is broken down as follows: Section 4.1 details collection and annotation of the BEST dataset. Section 4.2 discusses the need for temporal attention in skill determination and introduces the concept of 'rank-aware' attention. The method capable of learning this 'rank-aware' temporal attention is then described in Section 4.3, with Section 4.4 presenting the results of this method on EPIC-Skills and BEST.

4.1 Bristol Everyday Skill Tasks Dataset

To better explore skill-relevant parts of a video, the Bristol Everyday Skill Tasks (BEST) dataset was collected. This section first describes the collection of the videos in this dataset (Section 4.1.1) and the annotation of these videos for ranking (Section 4.1.2). Section 4.1.3 then explores the contents of the dataset.

4.1.1 Video Collection

The aim of this new dataset is to provide further tasks to study in the context of skill determination. A new dataset provides the opportunity to test on a larger assortment of skill tasks and better understand how to deal with variety in tasks and irrelevant parts of the video. The videos in this dataset were obtained from YouTube. This allows for variation in background, camera angle and the steps used to complete the task, which would be difficult to obtain in a lab setting.

Five tasks were selected: scrambling eggs, braiding hair, tying a tie, making an origami crane and applying eyeliner. The tasks selected were deliberately varied in their content and also differ from the tasks available in EPIC-Skills, as this allows more thorough testing of a proposed model. The aim was to collect 100 videos per task, over twice as many as the tasks collected for EPIC-Skills. To achieve this, the task name was used as a YouTube query and the top 400 videos were collected. These were then filtered to obtain good quality recordings across a variety of skill levels. Three Amazon Mechanical Turk (AMT) workers watched each video and responded to various questions about the video to determine its suitability for the dataset. The questions were:

- Does the video contain a person ⟨task_name⟩?
- Is the video recorded in a landscape orientation?
- Does the video contain a clear, unobstructed view of the task?

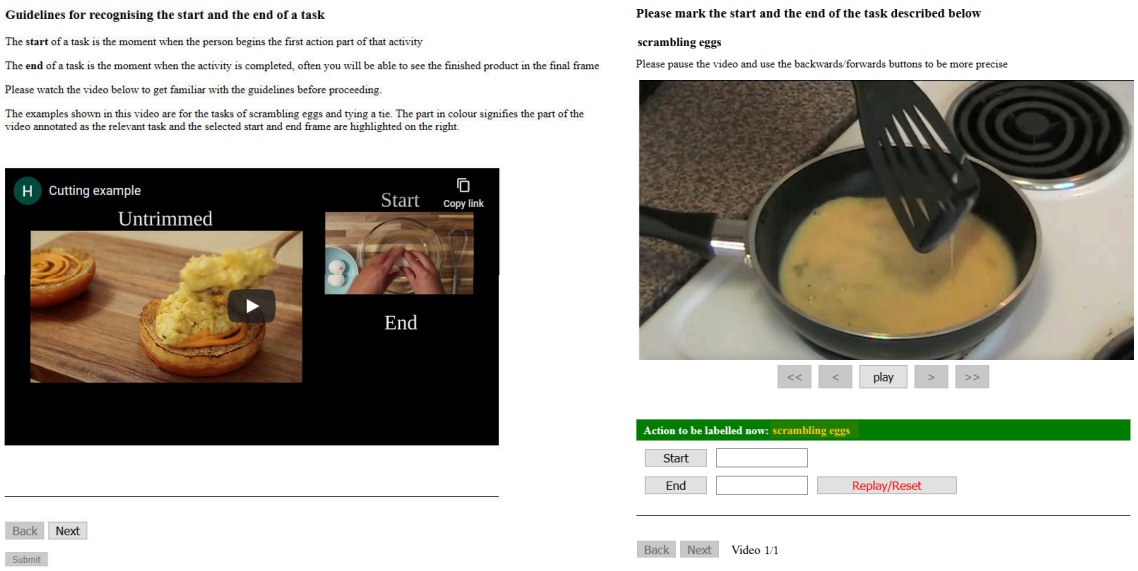

Figure 4.1: *The AMT interface used to trim videos. AMT workers are first shown a video explaining the start and end of the activity (left). They then watch each video in a HIT, noting the start and end times of the task with the buttons below the video (right).*

- Does the portion of the video containing ⟨task_name⟩ contain the full task, without skipping large portions?
- How well did the person perform ⟨task_name⟩?

For the latter question, AMT workers were given a choice of 'Beginner', 'Intermediate' or 'Expert'. This initial labelling was to ensure sufficient low-skill videos were selected before collecting the more demanding pairwise annotations, therefore it is only coarse-grained. For each task, the set of videos which AMT workers agreed met the quality requirements were filtered down to 100 videos. These were selected with a preference for intermediate and beginner rated videos, which were less common.

To focus on the task itself, the videos were trimmed to exclude the introduction and end of the video which typically contain irrelevant title sequences or the presenter talking to camera. AMT was also used to trim the videos. For this annotation, a similar approach to [30] was used. Specifically, each HIT contained 5 videos of the same task for which workers were asked to annotated the start and end time of the relevant activity. The interface is shown in Figure 4.1.

To help AMT workers understand the annotations required, each HIT began with a demonstration of a good annotation for the relevant task. Using the same task for all 5 videos in the HIT allowed workers to become familiar with the task and thus become

more accurate. Four AMT workers annotated each video. To obtain the final annotation for a video, the average agreement $v_i(j)$ was calculated for each of the $M = 4$ annotators using the following equation:

$$v_i(j) = \frac{1}{M-1} \sum_{m=1, m \neq j}^{M} \text{IoU}(V_i(j), V_i(m)) \tag{4.1}$$

$V_i(j)$ is one annotator's annotation for a video, i indexes the video and j indexes the annotator. Workers with an average IoU < 0.2 for multiple videos are rejected and the process is repeated until there is high agreement among the four annotators for each video. The annotation with maximal agreement was selected as the final annotation:

$$\hat{j} = \text{argmax}_j(v_i(j)) \tag{4.2}$$

if $v_i(\hat{j}) \geq 0.85$, otherwise it was manually verified. Examples of the videos obtained from this collection process are shown in Figure 4.2.

4.1.2 Pairwise Annotation

To annotate these videos for skill determination, a process similar to the annotation of EPIC-Skills (Section 3.2.2) was used. AMT workers were asked to watch pairs of videos simultaneously and select the video which displays higher skill. Each HIT is completed by 4 AMT workers and contains 5 video pairs from the same task. One of these was a quality control pair, as in the annotation of EPIC-Skills (Section 3.2.2). A pair was taken as ground-truth only if all four annotators agreed on a pair's ordering.

As the increase in the number of videos per task leads to an quadratic increase in the number of pairs, it is impractical and unnecessary to annotate every video pair. Instead, 40% of the possible pairings were annotated, with each video appearing in an equal number of pairs. This removed the need for exhaustive annotation as the transitive nature of skill ranking could be utilised to obtain pairs outside the original 40%. If a video x_i is annotated to indicate that it shows higher skill than video x_j and in turn x_j shows higher skill than x_k, it is safe to assume x_i shows higher skill than x_k.

After annotating the selected 40%, further annotations were performed to ensure the dataset contained challenging video pairs, close in the ranking. This is done by computing the separation of video pairs, as in Section 3.2.2 to determine similar pairs. First a directed graph of the pairs is created, with the videos $x_i \in X$ as nodes and edges $x_i \rightarrow x_j$ for annotated pairs $(x_i, x_j) \in P$. Refer to Figure 3.4 for an illustration of the graph this

Figure 4.2: *Examples of videos in the BEST dataset. Two videos with a difference in skill are shown for each task.*

	Task	#Videos	#Pairs	%Pairs	Av. Length (s)
EPIC-Skills	Chopstick Using	40	536	69%	46 ± 17
	Dough Rolling	33	181	34%	102 ± 29
	Drawing	40	247	65%	101 ± 47
	Surgery	103	1659	95%	92 ± 41
BEST	Scramble Eggs	100	2112	43%	170 ± 113
	Tie Tie	100	3843	77%	81 ± 47
	Apply Eyeliner	100	3743	76%	122 ± 105
	Braid Hair	100	3847	78%	179 ± 91
	Origami	100	3237	65%	386 ± 193

Table 4.1: *Comparing EPIC-Skills with BEST: number of videos, number and percentage of pairs and average and standard deviation of video length.*

process produces. If the two videos have the same length of the longest walk from any source node, or a difference of only 1, that video pair is sent for annotation.

To assess whether the crowd-sourced annotations are reliable and focus on the skill for the relevant task, a manual inspection of the annotated pairings was carried out. For each task, a random sample of 100 video pairs was selected and ordered by a golden annotator. The golden annotator agreed with 96% of the video orderings. This error rate is comparable to recently published datasets [30, 215].

4.1.3 Dataset Statistics

The number and percentage pairs obtained through the annotation process is shown in Table 4.1, along with the average video length per task. The BEST dataset is considerably larger than EPIC-Skills, both in terms of videos and annotated pairs. The task length in BEST is also generally longer and much more variable.

Potential biases in the data are measured to ensure the dataset is suitable for learning and evaluating skill. Five potential sources of bias are selected, two concerning quality and three relating a video's rank on YouTube. For video quality, bitrate and frame size are assessed. To gauge the relationship between the skill shown in a video and its popularity on YouTube, view count, like count and like/dislike ratio are measured. Table 4.2 displays the pairwise accuracy when using each of these metrics to rank videos. Since the majority of tasks have an accuracy close to random (50%), it can be concluded that there is little correlation between skill and video quality or skill and YouTube popularity for the videos in BEST. There is some inverse correlation with frame size, however this bias should be eliminated in any method which down-sizes the input videos.

Task	Bitrate	Frame Size	View Count	Like Count	Like/Dislike Ratio
Scramble Eggs	51.8	34.5	51.2	52.7	53.3
Braid Hair	48.8	31.0	56.3	54.9	46.2
Tie Tie	54.9	41.5	41.7	40.7	45.6
Origami	53.2	28.2	53.1	50.3	50.1
Apply Eyeliner	47.8	20.1	61.2	61.2	52.4

Table 4.2: *Pairwise accuracy (%) when using quality measures of video size and bitrate or view and like counts to determine skill.*

Figure 4.3: *An example of two scramble eggs videos from BEST. The top video is ranked higher in the ground-truth. Many segments such as adding the milk and placing the pan are not informative for differentiating the videos based on skill. These videos also show variability in skill through the video, with the bottom video showing better cracking of the eggs and the top better stirring.*

4.2 Temporal Attention in Skill Determination

The previous chapter made the assumption that the same level of skill is exhibited throughout the video, and thus skill can be determined by any of its parts. However, this is not the case, particularly with the longer and more complex tasks in the BEST dataset. Take the task of scrambling eggs shown in Figure 4.3, segments such as adding the milk, placing the pan and turning on the cooker may be uninformative when determining a subject's skill. On the other hand, the way in which the eggs are stirred in the pan are key. Additionally, there may be variation in skill across the video: in Figure 4.3 the participant in the higher ranked video begins by cracking the eggs poorly, but produces better scrambled eggs overall. This demonstrates that some actions can be more important than others for determining skill.

Accordingly, this chapter considers skill determination to be a fine-grained video understanding problem, where it is important to first localise relevant temporal regions to distinguish between instances. As in the previous chapter, this chapter targets skill determination for daily-living tasks, where ranking videos is more suitable than estimating an objective score.

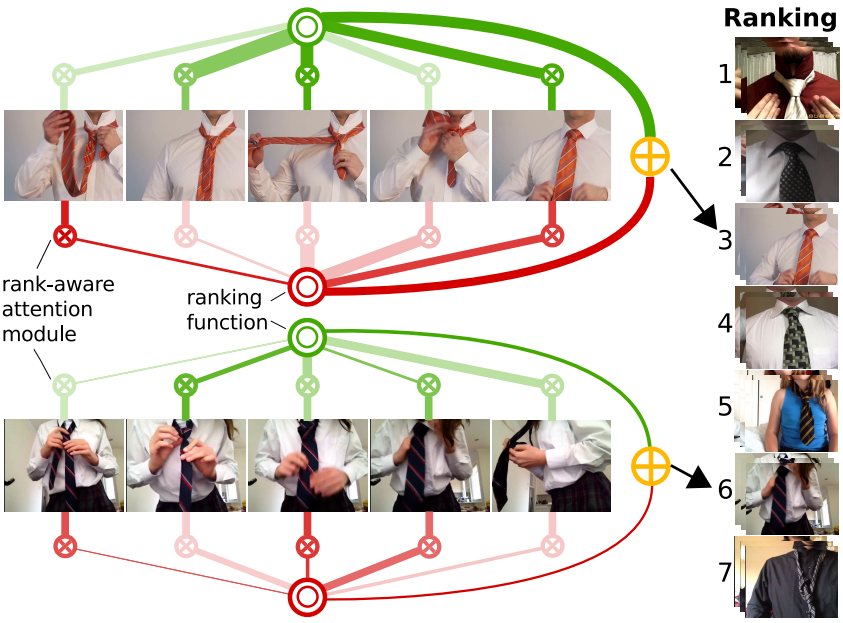

Figure 4.4: *Rank-aware attention for skill ranking. A video's rank is determined by using high (green) and low (red) skill attention modules, which determine each segment's influence to the rank. Both modules are fused (orange) for an overall skill assessment of the video. Line opacity indicates the attention value for a segment and the line thickness indicates the score.*

Attention has been increasingly used to identify important regions in fine-grained recognition problems, where intelligently weighting input is key to distinguish between similar categories. In the video domain, attention has been adopted for action recognition [139, 146] and localisation [92, 125, 138, 162]. See Chapter 2 for a more detailed summary. With only video-level supervision, attention allows a network to learn to focus on the segments containing the action of interest and ignore irrelevant background segments [125, 138]. Although the BEST (Section 4.1) and EPIC-Skills (Section 3.2) datasets proposed in this thesis are fully labelled datasets, there are similarities between weakly-supervised action localisation and skill determination. Both problems have long videos with only video-level labels where parts of the video will not be relevant to this label. Therefore, this chapter aims to use attention to determine the most informative parts of a video and use these parts to rank videos in terms of skill.

Many works which employ temporal attention use class-specific attention [96, 125, 138, 146]. This can attend to segments informative to a particular class unlike class-agnostic attention. However, these techniques are not applicable to ranking. Therefore, this chapter proposes a model to train rank-specific (referred to as rank-aware) attention.

The concept of the proposed approach is depicted in Figure 4.4. A Siamese neural network over temporal segments is devised, which includes attention modules adapted

from [125]. These are trained to be rank-aware using a novel loss function. This is because relevance may differ depending on the skill displayed in the video — *e.g.* mistakes may not appear in higher-ranked videos. When trained with the proposed loss, these modules specialise to separately attend to parts of the video informative for high skill or sub-standard performance. The overall skill assessment of the video is formed from a weighted average of the individual segment skill scores based on the segment's importance to determining skill.

4.3 Rank-Aware Attention Network

This section outlines the proposed rank-aware attention network which can be used to determine skill in long videos by attending to the most informative segments. First the skill determination problem is reformulated in Section 4.3.1, with a focus on long videos. Section 4.3.2 then gives an overview of the network used, with the newly proposed loss functions which make the attention rank-aware covered in Sections 4.3.3 and 4.3.4. The details of the attention modules within the network are then explained in Section 4.3.5.

4.3.1 Problem Formulation

Similar to the previous chapter, this chapter proposes a supervised pairwise ranking approach for skill determination. In this set-up, there is a set of videos $x \in X$ of a particular task. Within each task, there is a set of pairs P between the videos, where each video pair $(x_i, x_j) \in P$ has been annotated to indicate video x_i displays higher skill than video x_j. Such pairwise annotations can be acquired for any task using crowdsourcing (see Sections 3.2.2 and 4.1.2). The aim is to learn a ranking function $f(\cdot)$ for an individual task such that

$$f(x_i) > f(x_j) \qquad \forall (x_i, x_j) \in P \tag{4.3}$$

The previous chapter assumed that these pairwise skill annotations could be propagated to any part of the video (see Equation 3.5). Given x_i^t is the t^{th} video segment, $t \in [1, T]$, skill annotations were propagated and $f(\cdot)$ was learnt so that,

$$f(x_i^t) > f(x_j^t) \qquad \forall t \in [1, T]; (x_i, x_j) \in P \tag{4.4}$$

An alternative approach to deal with long videos [134, 208], is to use a uniform weighting

of feature vectors to learn a video-level ranking. This would assume all parts of the video are equally important for skill assessment, *i.e.* $u(x_i) > u(x_j)$ where,

$$u(x_i) = f(\frac{1}{T}\sum_t x_i^t) \tag{4.5}$$

As noted previously, these assumptions do not hold. First, some parts of the video may not exhibit any difference in skill, such as the end result in Suturing shown in Figure 3.6. Alternatively, video parts may even show reverse ranking — where the overall better video has segments exhibiting less skill (see Figure 4.3). Second, non-uniform pooling should better represent the video's overall skill by increasing the weight for segments more pertinent to a subject's skill. This was the case in the Chopstick Using task, where performance degraded as segments from the earlier parts of the video were added (see Figure 3.12). Third, comparing corresponding video chunks (x_i^t, x_j^t) assumes tasks are performed in a set order at a similar rate of progression. This is also not necessarily the case, especially with more complex and varied tasks, such as those in the BEST dataset. This chapter deviates from these assumptions and instead aims to jointly learn temporal attention $\alpha(\cdot)$, alongside ranking function $r(\cdot)$ such that

$$s(x_i) > s(x_j); \qquad s(x_i) = r(\sum_t \alpha(x_i^t)x_i^t) \tag{4.6}$$

While $\alpha(\cdot)$ is a standard attention module for relevance, segments most crucial to determining skill may differ depending on the subject's skill; a low-skill subject may perform certain actions (*e.g.* mistakes) not performed by a high-skill subject and vice-versa. Therefore, this chapter proposes to train two general attention modules to produce scores s^+ and s^- for all pairs $(x_i, x_j) \in P$, such that:

$$s^+(x_i) > s^+(x_j); \qquad s^-(x_i) > s^-(x_j); \qquad s^+(x_i) \gg s^-(x_j) \tag{4.7}$$

In particular, $s^+(x_i) \gg s^-(x_j)$ will encourage the two attention modules to diverge, such that one attends to segments which display a high skill (α^+) and the other to low skill (α^-), along with separate ranking functions g and h:

$$s^+(x_i) = g(\sum_t \alpha^+(x_i^t)x_i^t) \tag{4.8}$$

$$s^-(x_i) = h(\sum_t \alpha^-(x_i^t)x_i^t) \tag{4.9}$$

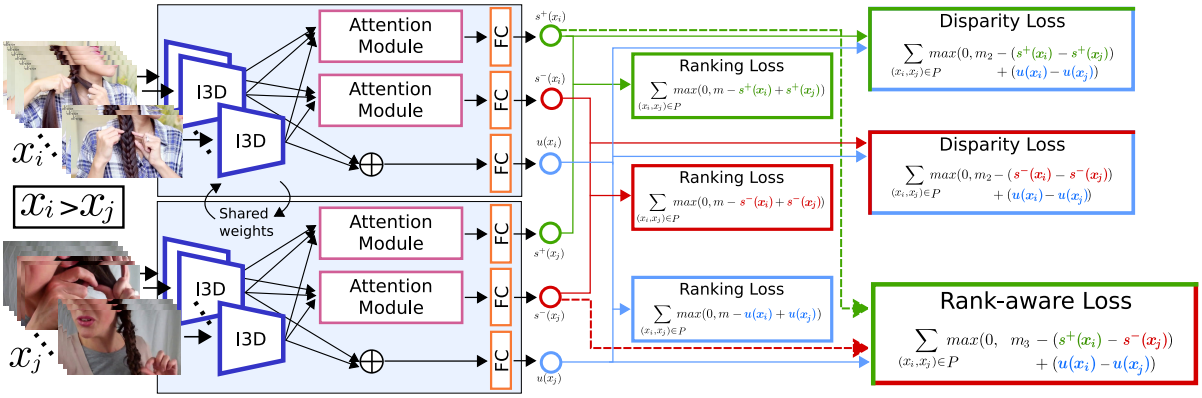

Figure 4.5: *Given a ranked pair of videos (p_i, p_j), where p_i exhibits higher skill than p_j, each video is uniformly split into segments. Extracted features (I3D) are passed into a pair of attention modules to produce video-level representations for the ranking functions (FC layers). Each ranking function produces a score s^+ (green) or s^- (red). Additionally, a uniformly weighted video representation produces a third ranking score u (blue). Three types of losses are defined: the ranking loss maximises the margin (green-to-green, red-to-red, blue-to-blue) between the pair of ranked videos, the disparity loss ensures attention branches outperform uniform (green-to-blue, red-to-blue) and the final loss optimises the attention modules to become rank-aware (green-to-red).*

4.3.2 Overall Network

The overall architecture is shown in Figure 4.5. The Siamese network takes a video pair $(x_i, x_j) \in P$ as input. Each video in the pair is split into T segments of uniform length. The features from all video segments $\{x_i^t\}$ are then passed to the three branches of the network. Within each branch, a video-level representation is obtained by combining the video segments. The weights which indicate how the video segments are combined either come from the learnt attention modules $\alpha^+(\cdot)$ and $\alpha^-(\cdot)$ (see Section 4.3.5) or from the uniform weighting $\frac{1}{T}\sum_t x_i^t$. Each of these branches also contains a ranking function learnt with a fully connected layer. The ranking functions are $g(\cdot)$, $h(\cdot)$ and $f(\cdot)$ respectively for the high-skill, low-skill and uniform branches. These fully connected layers are separate for each branch, but are shared by both sides of the Siamese network. For both videos in the pair, each of these branches produces a score, s^+ for the high-skill branch (Equation 4.8), s^- for the low-skill branch (Equation 4.9) and u for the uniform branch (Equation 4.5).

The ranking function for each branch (and attention function where applicable) is learnt using a margin **ranking loss** function as in the previous chapter. Three within-branch margin ranking loss functions are used to evaluate whether each branch outputs a higher score for x_i than x_j, for a pair (x_i, x_j). For the high-skill branch, this loss is defined as:

$$L^{+}_{rank} = \sum_{(x_i,x_j)\in P} \max(0, m - s^{+}(x_i) + s^{+}(x_j)) \tag{4.10}$$

where $s^{+}(x_i)$ is the final score of video x_i from the high-skill branch and m is a constant margin. The ranking loss is defined similarly, with the same margin, for the low-skill and uniform weighting branches:

$$L^{-}_{rank} = \sum_{(x_i,x_j)\in P} \max(0, m - s^{-}(x_i) + s^{-}(x_j)) \tag{4.11}$$

$$L^{u}_{rank} = \sum_{(x_i,x_j)\in P} \max(0, m - u(x_i) + u(x_j)) \tag{4.12}$$

The weights of the network are also optimised by two cross-branch ranking losses: a disparity loss and a rank-aware loss. These losses further encourage the attention module to attend to the most skill-relevant video segments and will be outlined in the following sections.

When testing the network, a single video is evaluated and its rank is assigned through its ranking score for the high and low skill branches:

$$R(x_i) = s^{+}(x_i) + s^{-}(x_i) \tag{4.13}$$

The uniform branch is used only to aid the learning of the attention module in the high and low skill branches, as Section 4.3.3 and 4.3.4 will explain, and is therefore not used in testing. Section 4.4 will also demonstrate that the attention branches are more informative than the uniform weighting branch. Note that in training, $s^{+}(\cdot)$ and $s^{-}(\cdot)$ are learnt such that $s^{+}(x_i) > s^{+}(x_j)$ and $s^{-}(x_i) > s^{-}(x_j)$ which implies $s^{+}(x_i) + s^{-}(x_i) > s^{+}(x_j) + s^{-}(x_j)$. Although α^{-} will attend to low-skill segments, the overall score s^{-} reflects the correct ranking of the videos.

4.3.3 Optimising Attention with Uniform Weighting

Since the uniform weighting branch is not used in testing it may not be apparent why it is needed. When training the high and low skill attention modules with the margin ranking losses (Equations 4.10 and 4.11), the attention modules frequently fall into local-minima and the results vary between runs. The segments attended to are often uninformative for skill and learnt attention weights for such local-minima perform worse than a uniform weighting of segments. For this reason, the uniform branch is incorporated into the

network along with an attention **disparity loss**. This disparity loss explicitly encourages an attention branch to outperform the uniform weighting branch:

$$L^{+}_{disp} = \sum_{(x_i,x_j)\in P} \max(0, m_2 - (s^{+}(x_i) - s^{+}(x_j)) + (u(x_i) - u(x_j))) \tag{4.14}$$

Here, m_2 is a separate margin from m in Equations 4.10, 4.11 and 4.12. This loss uses the same framework as the margin ranking loss, however, similarly to a triplet loss (see Chapter 2), it optimises the difference between a pair of values to be greater than the difference between another pair. An analogous loss is defined between the low-skill and uniform branches of the network:

$$L^{-}_{disp} = \sum_{(x_i,x_j)\in P} \max(0, m_2 - (s^{-}(x_i) - s^{-}(x_j)) + (u(x_i) - u(x_j))) \tag{4.15}$$

For a video pair (x_i, x_j), this loss causes the difference between scores $s^{-}(x_i)$ and $s^{-}(x_j)$ to be greater than the difference between scores $u(x_i)$ and $u(x_j)$. This enables the network to use the uniform branch as a reference and learn to attend to the most discriminative regions with α^{-}, which the network can use to produce more confident ranking of the videos in each pair. This loss alone could instead cause the performance of ranking function $f(\cdot)$ to degrade, however by jointly optimising with Equation 4.12 this is avoided.

4.3.4 Rank-aware Attention

With the two types of loss functions defined so far, the two learnt attention modules α^{+} and α^{-} are indistinguishable. They will attend to skill-relevant segments, however there is no reason for one module to focus on the high-skill segments and the other to focus on low-skill segments, nor any reason for the two attention modules not to focus on the same segments to one another. To optimise these modules to achieve the desired response, this chapter proposes a **rank-aware loss**:

$$L_{rank\_aware} = \sum_{(x_i,x_j)\in P} \max(0, m_3 - (s^{+}(x_i) - s^{-}(x_j)) + (u(x_i) - u(x_j))) \tag{4.16}$$

This loss ensures that the difference between the high-skill branch score $s^{+}(\cdot)$ of the higher ranked video x_i and the score from low-skill branch $s^{-}(\cdot)$ of the lower ranked video x_j is larger than the difference between the scores of these videos from the uniform branch. With the requirement that $m_3 > m_2$, this loss achieves the initial goal that

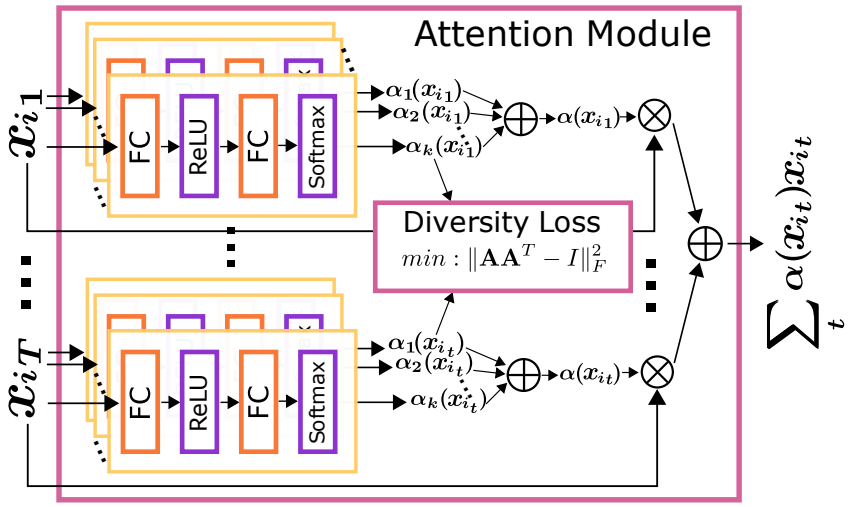

Figure 4.6: *The attention module consists of K attention filters, each outputting a scalar weight per segment. These weights are used to pool segments features into an overall video-level feature.*

$s^+(x_i) \gg s^-(x_j)$. To satisfy this loss, the attention module in the high-skill branch attends to parts of the video where the participant displays higher skill, as these will produce higher scores. Similarly, the attention module in the low-skill branch diverges to attend to video segments which typically produce lower scores.

The overall training is then conducted by combining the losses:

$$L_R = \sum_{i=\{+,-,u\}} L^i_{rank} + \sum_{i=\{+,-\}} L^i_{disp} + L_{rank\_aware} \tag{4.17}$$

As the training iterates through pairs in P, the same video will be considered higher skill in one pair and lower in another (*e.g.* $(x_i, x_j) \in P$ and $(x_j, x_k) \in P$). The network accordingly optimises the *shared weights* so as to learn rank-aware attention modules.

4.3.5 Multi-Filter Attention Module

The attention modules $\alpha^+(\cdot)$ and $\alpha^-(\cdot)$ each take a set of T video segments and learn a weighting of these segments informative for skill ranking. This section will explain the structure of these modules. The two attention modules are encouraged to learn different weights with L_{rank_aware}, however they have the same structure, therefore this section will refer to the generic attention module $\alpha(\cdot)$.

The architecture of an attention module is shown in Figure 4.6. It consists of K filters, each comprised of two fully connected layers. The first of these fully connected layers is followed by a ReLU activation function and the second is followed by a softmax. This

is based on the attention filters used in [101] and [125]. Filters are combined to achieve segment level attention:

$$\alpha(x_{it}) = \sum_{k=1}^{K} \alpha_k(x_{it}) \tag{4.18}$$

where α_k refers to the k^{th} attention filter for the attention module $\alpha(\cdot)$. Importantly, $\sum_{t=1}^{T} \alpha_k(x_{it}) = 1$ for each of the K filters. Multiple attention filters are included to encourage a module to attend to the numerous skill-relevant parts present in lengthy videos; a single filter typically focuses on only one element of the task [101]. To regularise the K filters, a diversity loss is used. The $K \times T$ attention matrix relating to video x_i is defined as:

$$\mathbf{A}_i = \begin{bmatrix} \alpha_1(x_{i1}) & \alpha_1(x_{i2}) & \dots & \alpha_1(x_{iT}) \\ \alpha_2(x_{i1}) & \alpha_2(x_{i2}) & \dots & \alpha_2(x_{iT}) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_K(x_{i1}) & \alpha_K(x_{i2}) & \dots & \alpha_K(x_{iT}) \end{bmatrix} \tag{4.19}$$

and the diversity loss is as follows:

$$L_{div} = \sum_{(x_i, x_j) \in P} \|\mathbf{A}_i \mathbf{A}_i^T - \mathbf{I}\|_F^2 \tag{4.20}$$

where $\mathbf{I}$ is the identity matrix and $\|\cdot\|_F^2$ denotes the Frobenius norm. Similar diversity losses have been used successfully in other applications, such as text embedding [95] — here it is used to regularise temporal attention in video.

The non-diagonal elements a_{kl} in the matrix $\mathbf{A}_i \mathbf{A}_i^T$ correspond to the sum of the element-wise product of the attention values produced by two different attention filters α_k and α_l where $k \neq l$. Due to the softmax on the individual attention filters elements, this sum must also be between 0 and 1:

$$0 \geq a_{kl} = \sum_{t=1}^{T} \alpha_k(x_{it}) \alpha_l(x_{it}) \geq 1 \tag{4.21}$$

When there is no overlap between the distributions of attention values produced by the two filters α_k and α_l, a_{kl} will be 0. Otherwise the value will be positive and will subsequently cause the loss to be greater than 0. The loss will be largest when the two attention filters give identical attention values for each video segment, in such a case $a_{kl} = 1$. As a result, the diversity loss will encourage the matrix $\mathbf{A}_i$ to be full-rank.

Without such a loss function, all filters will attend to the same most discriminative part in the video. This would make multiple filters redundant. The importance of multiple filters is assessed in Section 4.4.

The diversity loss also encourages filters to be sparse and pick the few most informative segments. By subtracting the identity matrix, the elements on the diagonal of $\mathbf{A}_i\mathbf{A}_i^T$ are optimised to be close to 1, while other elements should be close to 0. This will happen when an attention filter α_k focuses on as few video segments as possible, with other video segments given an attention value of 0.

A diversity loss L_{div}^{+} is applied to the attention module in the high-skill branch and a second diversity loss L_{div}^{-} is used in the low-skill attention module. Note that the diversity loss is within an attention module; diversity is not enforced between modules in different branches. Attentions are allowed to overlap and do so when the segment is relevant at all skill levels. The overall training loss then becomes:

$$L_R = \sum_{i=\{+,-,u\}} L_{rank}^{i} + \lambda \sum_{i=\{+,-\}} L_{div}^{i} + \sum_{i=\{+,-\}} L_{disp}^{i} + L_{rank\_aware} \tag{4.22}$$

A weighting factor λ is introduced to down-weight the diversity losses. This loss begins quite large, therefore can cause the network to initially only focus on diverse segments instead of skill-relevant ones. Empirically, a weighting coefficient was not found to be necessary for the other losses as the margins m, m_2 and m_3 already indicate the relative importance of these losses.

4.4 Experiments and Results

This section reports results with the proposed method on both EPIC-Skills (Section 3.2) and BEST (Section 4.1). First, the implementation details are described in Section 4.4.1. The results of the proposed method are then compared to baselines in Section 4.4.2. Section 4.4.3 studies the effect of the different loss functions and branches in the proposed method as well as investigating other parameters such as the number of attention filters. Qualitative results of the method are then shown in Section 4.4.4.

4.4.1 Implementation Details

Video Features. To capture the nuances of skill from a video, dense sampling is needed. Therefore, $T = 400$ stacks of 16 frames are uniformly sampled at 10fps for each video as in [125]. Images are re-scaled to have a height of 256 pixels, then centre cropped

to 224 × 224 pixels. Features are extracted using an I3D [18] network first pre-trained on ImageNet [32] and then fine-tuned on Kinetics-400 [79]. Including features from an optical flow I3D network gave little improvement, so for efficiency only features from an RGB network were used. This gives $T = 400$, feature vectors per video, each of which is 1024 dimensional.

Architecture Details. Attention filters consist of two fully connected layers, where the first has an output size of 256. In all experiments the weight of the diversity losses (Equation 4.20) is $\lambda = 0.1$. The margins for the ranking losses, disparity losses and rank-aware losses are set to $m_1 = 1$ (Equation 4.10), $m_2 = 0.1$ (Equation 4.14) and $m_3 = 0.3$ (Equation 4.16) respectively. $K = 3$ filters are used in both attention modules.

Training Details. All models are trained using the Adam optimiser [80] with a batch size of 128 and learning rate of 10^{-4} for 2000 epochs. To prevent over-fitting, the features are augmented by adding a small amount of noise, $\mathcal{N}(0, 0.01^2)$, as in [125]. For stable training, the network's parameters are optimised iteratively. For one epoch the attention module parameters are fixed and the ranking fully-connected layer weights are optimised using the L_{rank} losses (Equations 4.10, 4.11 and 4.12). For the next epoch the ranking fully-connected layers are fixed and the attention module weights are learnt using the remaining losses (L_{div}, L_{disp} and L_{rank_aware}). This continues to alternate until the training is complete.

Evaluation Metric. The same evaluation metric as in Chapter 3 is used, namely *pairwise accuracy*. This is the percentage of correctly ordered video pairs.

Dataset Splits. For EPIC-Skills (Section 3.2) the four-fold cross validation training and testing splits from Chapter 3 are used. For BEST (Section 4.1) a single 75%:25% split per task is used, as the number of pairs per task is much larger. The videos in each split are selected randomly, although it was checked to ensure there was a reasonable distribution of skill levels in both train and test. Unlike EPIC-Skills, the test set consists exclusively of pairs where neither video is present in training. This is a somewhat more challenging test set, as it only measures test videos against each other and not where they are placed within a known ranking.

4.4.2 Comparison to Baselines

Table 4.3 shows the mean pairwise accuracy for the proposed method for each dataset. Four temporal attention baselines are compared to, two learnt and two predefined. Furthermore, the approach from Chapter 3 is compared to, which is the only prior work to determine skill in daily-living tasks. The rank-aware attention network proposed in

Method	EPIC-Skills	BEST
Chapter 3	76.0	75.8
Last Segment	76.8	61.0
Uniform Weighting	78.8	73.6
Softmax Attention	74.5	72.3
STPN [125]	74.3	70.0
Rank-aware Attention Network	**80.3**	**81.2**

Table 4.3: *Results of the proposed rank-aware attention network in comparison to baselines for the EPIC-Skills and BEST datasets. The proposed method outperforms every baseline for both datasets.*

this chapter outperforms the approach proposed in **Chapter 3** by 4.3% and 5.4% on EPIC-Skills and BEST respectively.

Last Segment is a hard-coded attention baseline. It learns a ranking function, with a fully connected layer, using only the I3D features extracted from the last segment of each video. Note that videos in both datasets are trimmed such that the last segment will generally contain a clear view of the task's end result. As explained in Section 3.3, it could be argued that the final outcome is sufficiently informative to demonstrate a participant's level of skill, without having to view the rest of the task's performance. However, the results demonstrate that the last segment is insufficient for determining skill for the tasks in EPIC-Skills and BEST. The last segment baseline performs worse than the proposed method for both datasets, particularly on BEST (-20.2%).

Uniform weighting is another hard-coded attention baseline. In this baseline it is assumed that every video segment is equally important for determining skill. A ranking function is learnt with a fully connected layer after average pooling all I3D features extracted from a video. The improvement of the proposed method over this baseline is particularly large for BEST, where the videos tend to be longer and contain more segments either irrelevant to skill or the task itself.

Softmax attention is a basic form of temporal attention without the proposed additions, *i.e.* a single attention branch is optimised with only L_{rank}. Interestingly, the inclusion of softmax attention decreases the accuracy for both datasets from a naive uniform weighting of segments (-4.3% and -1.3%). Although softmax attention achieves higher accuracy than uniform for several tasks, the softmax attention is highly inconsistent between runs (see the consistency experiment in Section 4.4.3), often falling into local minima early in training.

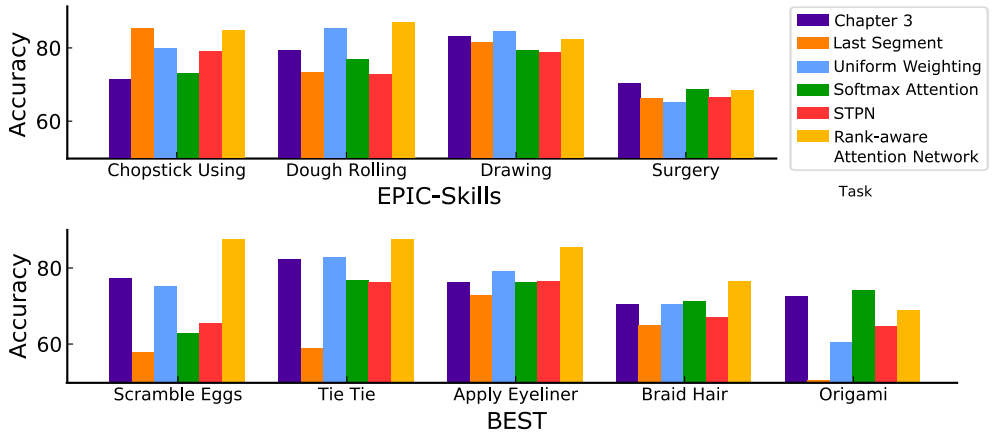

Figure 4.7: *Individual baseline results for each task in EPIC-Skills and BEST.*

The final baseline is the Sparse Temporal Pooling Network (**STPN**) [125], originally proposed for weakly-supervised action localisation. It uses a temporal attention filter optimised by a sparsity loss to encourage only a few video segments to be considered relevant. For comparison in skill determination, the class-agnostic attention from this method is adapted into a Siamese network optimised by the margin ranking loss L_{rank}.

The STPN baseline performs worse than both the proposed method and the softmax attention baseline. There are two main difference between STPN and the softmax attention: the number of attention filters (1 versus 3) and the inclusion of the additional sparsity loss in STPN. The sparsity loss encourages STPN to select only a small number of video segments with a high weighting. As the skill tasks are quite complex, particularly in BEST, this harms the networks performance. A full evaluation into the effect of the number of filters can be found in the ablation experiments (Section 4.4.3).

In general, the baselines struggle more on BEST as they are affected by the lengthy videos and increase in irrelevant segments, while the last segment baseline is affected by variations in environment and viewpoint. By focusing on key segments indicative of skill, the proposed method is able to combat these difficulties and gain a larger increase on this dataset.

Per Task Results. Figure 4.7 shows the performance of the proposed method and baselines for each task in EPIC-Skills and BEST. Overall the proposed method outperforms all baselines on both datasets, however Figure 4.7 demonstrates that the best performing baseline varies between tasks.

The uniform weighting and the method from Chapter 3 perform well in simpler tasks such as Dough Rolling and Drawing, where many different parts of the video demonstrate skill. The method from Chapter 3 also performs well for Surgery and Origami,

while the uniform weighting performs poorly for these tasks. This is due to the method from Chapter 3 allowing back-propagation through the features, which is beneficial to identifying the fine-grained details relevant to skill in Surgery and Origami.

As was the case in the previous chapter, the last segment baseline is most successful in Chopstick Using. The last segment baseline uses the I3D features used by the rank-aware attention network, thus it incorporates more temporal information than previously. Therefore, it will show the participant successfully picking up or struggling to grasp hold of a bean in addition to the number of beans moved. The last segment baseline also performs well in Drawing, where it will display the final drawing. However, like the other tasks, it is beneficial to assess skill with more than the last segment in Drawing.

Although there is significant difference between the best performing baselines for each task, the proposed rank-aware attention network does outperform both learnt attention baselines (softmax attention and STPN) in nearly every task. The per-task results highlight the difficulty of skill determination, where methods need to be applicable to determining skill in very different tasks. By learning to attend to the skill-relevant parts of a video, the proposed method is able to cope with the challenging diversity of tasks and consistently perform well.

4.4.3 Ablation Study

This section examines the contribution of each model component to the performance of the proposed method. First, the effect of the proposed loss functions is looked at, followed by the contribution of each network branch. The improved consistency with the uniform branch is also explored, as is the effect of the number of attention filters.

Loss Functions. Figure 4.8 shows a per-task ablation of the individual terms in the final loss function (Equation 4.22). The inclusion of the diversity loss increases the result by 2% for both datasets. It is particularly useful for Drawing (+7.3%) and Tie Tie (+6%), as videos in these tasks consistently have many skill-relevant segments.

Figure 4.8 also shows that using the disparity loss generally improves the result further. L_{disp} encourages the network to attend to segments which are better at discriminating between skill levels than the uniform weighting. In tasks such as Chopstick Using and Scramble Eggs, where the basic softmax attention performs similarly to uniform, this can help significantly.

The proposed rank-aware loss further improves the results, particularly for BEST (average improvement of +5%). This is especially true for Scramble Eggs and Apply Eye-

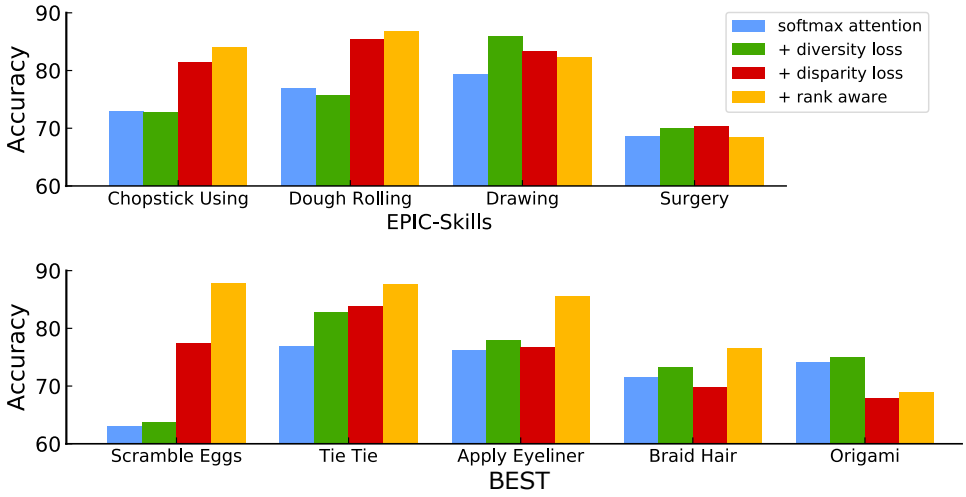

Figure 4.8: *Ablation study of loss functions on all tasks. In general, each additional loss term gives an improvement. The most significant improvement comes from the rank-aware loss which gives an average increase of 5% on BEST.*

liner (+10.4% and +8.8% respectively). These tasks contain many instances of actions and behaviours specific to subjects with higher or lower skill, as will be seen in Section 4.4.4.

There are three exceptions to this overall trend: Drawing, Surgery and Origami. Surgery maintains a similar score throughout the ablation test and has the lowest final score of all tasks. This is likely due to the I3D features not capturing the skill-relevant detail. One explanation for this is the domain shift between Kinetics, which I3D was pre-trained on, and the Surgery tasks. Since the Surgery tasks are performed via robotic arms, they appear quite different to the human-object interactions and human-human interactions found in Kinetics.

Drawing and Origami both drop with the addition of L_{disp}. In Drawing, the attention branches struggle to outperform the uniform branch, indicating that most segments are relevant for determining skill. In Origami the movements are incredibly subtle. This, combined with the longer length of the Origami videos, makes it one of the most challenging and complex tasks. The proposed method performs well on the majority of tasks, however the challenges presented by the Origami task highlight that skill determination in long video is not yet solved.

Branch Contribution. Having trained the proposed model with the overall loss (Equation 4.22), this section assesses the performance of the individual branch scores. Figure 4.9 shows that for many tasks the proposed model is able to learn high and low-skill branches which are both more informative than uniform. This is particularly true for

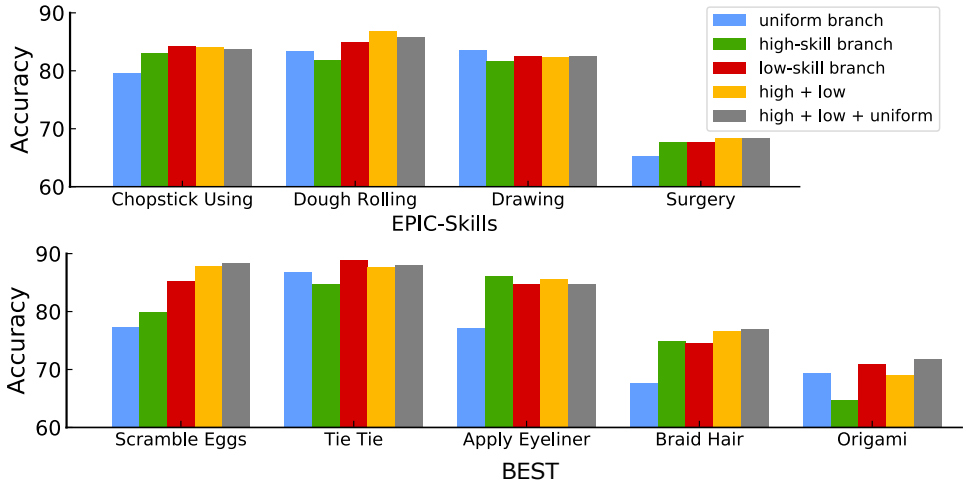

Figure 4.9: *Contribution of different branches in the network. The addition of L^{+}_{disp} and L^{-}_{disp} cause both the high and low skill branches to perform better than uniform in most tasks. These branches offer complementary information causing an improvement in the final result.*

tasks such as Chopstick Using and Scramble Eggs which do not benefit from attention until the disparity loss is introduced (see Figure 4.8). Within tasks, the performance of high and low skill branches can vary. This is evident in the Tie Tie task, with the low-skill branch performing best (+4.3%). Here, the presence of hesitation in lower-ranked videos proves effective for skill ranking.

The fusion of high and low skill branches further improves the average result (EPIC-Skills +2.9% and BEST +3.2%). In many tasks, the branches offer complementary information, as each branch can attend to separate video segments, specific to either high or low skill. However, in some tasks, such as Origami and Apply Eyeliner, either the high or low-skill branch on its own is more effective than the combination.

While instrumental in stabilising the training, as will be shown next, adding the scores produced by the uniform branch when testing offers little improvement to the combination of high and low-skill branches. Since the high and low skill branches are explicitly encouraged to pick segments more informative than the uniform weighting, there is little information that the uniform branch is able to add. Origami is the only task for which this is not the case, due to the high-skill branch's poor performance in comparison to uniform.

Consistency. The purpose of incorporating the uniform branch is to stabilise training of the attention. Without the uniform branch and the disparity loss (Equations 4.14 and 4.15), the performance of the network is much more variable between runs. Figure 4.10 displays this variability. The full model is tested against a model without the

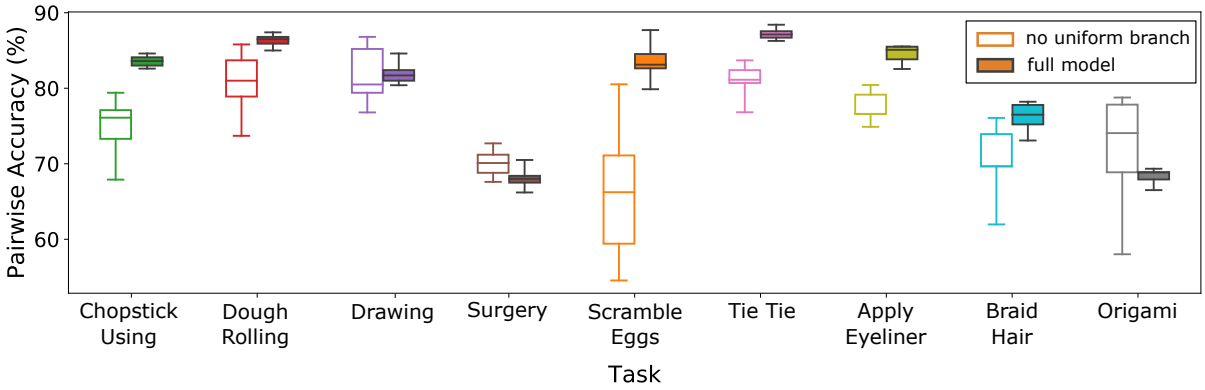

Figure 4.10: *The variability in accuracy when using the proposed model with and without the uniform branch and related losses. The inclusion of the uniform branch has a large improvement on the consistency of the model.*

uniform branch, and therefore without the disparity and rank-aware losses. Both settings are repeated 5 times for every task in EPIC-Skills and BEST.

For every task, regardless of the improvement in accuracy obtained by the proposed network, the consistency is significantly improved. Some tasks, such as Scramble Eggs and Braid Hair do still have a large range in performance with the proposed method, although the variability is much lower than previously.

Number of Filters. Figure 4.11 tests the effect of K, the number of filters per attention module (see Section 4.3.5). The previous sections report results using $K = 3$, which shows an improvement over a single filter in the majority of tasks. However, with $K > 3$ the accuracy generally does not increase further, as less informative segments are included. Only for Drawing does the result continue to increase with further attention filters, as this task has the best performance with a uniform weighting of segments.

Although almost all tasks improve from the addition of multiple attention filters, the amount each task improves is different. In general, the additional filters are more beneficial in the longer, more complex tasks in BEST, as the complexity of these tasks cannot be captured with a single attention filter.

Separate to the ablation of K, it is important to assess how the two rank-aware attention modules compare to a single attention module with double the number of filters. Table 4.4 compares the results of a single attention module with $K = 6$ filters to the proposed two attention modules optimised by the rank-aware loss with $K = 3$ filters. These results show there are significant improvements for the rank-aware attention modules in many tasks, particularly those in BEST. It is clear that the proposed rank-aware attention is doing more than simply doubling the number of attention filters.

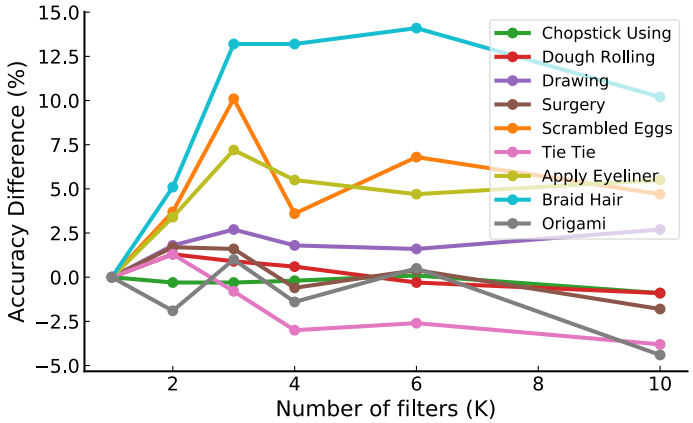

Figure 4.11: *The number of filters per attention module (K) is tested for all tasks. Increasing the number of filters over $K = 1$ shows a clear improvement in performance for many tasks, with the majority of tasks peaking at $K = 3$.*

Task	1 Module	2 Modules
Chopstick Using	80.1	**84.7**
Dough Rolling	85.2	**86.9**
Drawing	**82.1**	**82.3**
Surgery	**70.0**	68.5
Overall	79.3	**80.6**

Task	1 Module	2 Modules
Scramble Eggs	75.3	**87.7**
Tie Tie	79.4	**87.6**
Apply Eyeliner	79.2	**85.5**
Braid Hair	65.8	**76.5**
Origami	**71.2**	68.9
Overall	74.2	**80.2**

Table 4.4: *Testing the proposed two rank-aware attention modules against a single attention module with double the number of attention filters.*

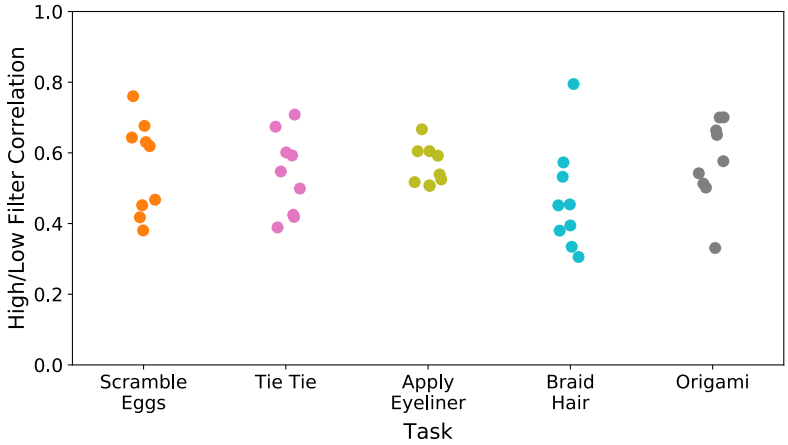

Figure 4.12: *The correlation of high and low skill attention filters for all tasks in BEST. This demonstrates attention modules attend to different video segments.*

Filter Correlation. Figure 4.12 shows the Spearman's rank correlation between pairs of high and low skill attention filters. These are averaged over the videos for each task. Most filter pairs in BEST have a low correlation, meaning they attend to different video

segments. There are some cases where filters in BEST have a higher correlation (*e.g.* Braid Hair at $\rho = 0.8$) as some video segments are relevant at all levels of skill.

The correlation between the high and low skill filters is also measured for the EPIC-Skills tasks in Figure 4.13. As seen in Figure 4.11, increasing the number of filters in an attention module has little effect on these simpler tasks; the additional filters focus on the same video parts as the original high or low skill filters. This means the correlations between each pair of high and low skill filters are similar. Despite filters within a module being similar, the low correlations in Dough Rolling and Drawing indicate that the high and low skill attention modules focus on very different parts of the video.

For some tasks in EPIC-Skills, such as Chopstick Using, the correlation between pairs of high and low skill filters is large. The Chopstick Using task is relatively short and many video segments will display features relevant to both high and low skill. The Surgery tasks also have a reasonably strong correlation between high and low skill filters. Videos in Surgery are typically longer than Chopstick Using, however many of segments displaying entry and exit of the needle are relevant to all levels of skill. Despite the correlations for both of these tasks being high, it is still < 1. This demonstrates that the two attention modules do indeed learn to focus on different parts of the video.

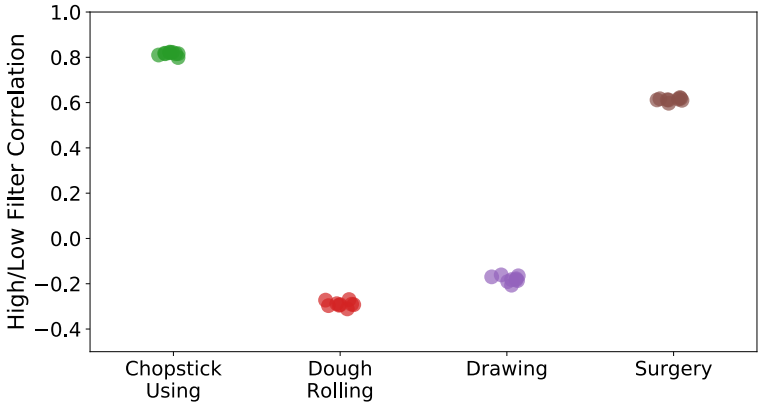

Figure 4.13: *Correlation between high and low skill attention in EPIC-Skills.*

4.4.4 Qualitative Results

This section first looks at example ranking results which illustrate the video segments attended to. Then the distributions of attention values throughout each task are shown. Finally, individual high-skill and low-skill attention filters are examined.

Ranking Examples. Figure 4.14 shows example rankings for three tasks: Chopstick Using (top), Scramble Eggs (middle) and Tie Tie (bottom). Attention weights for both

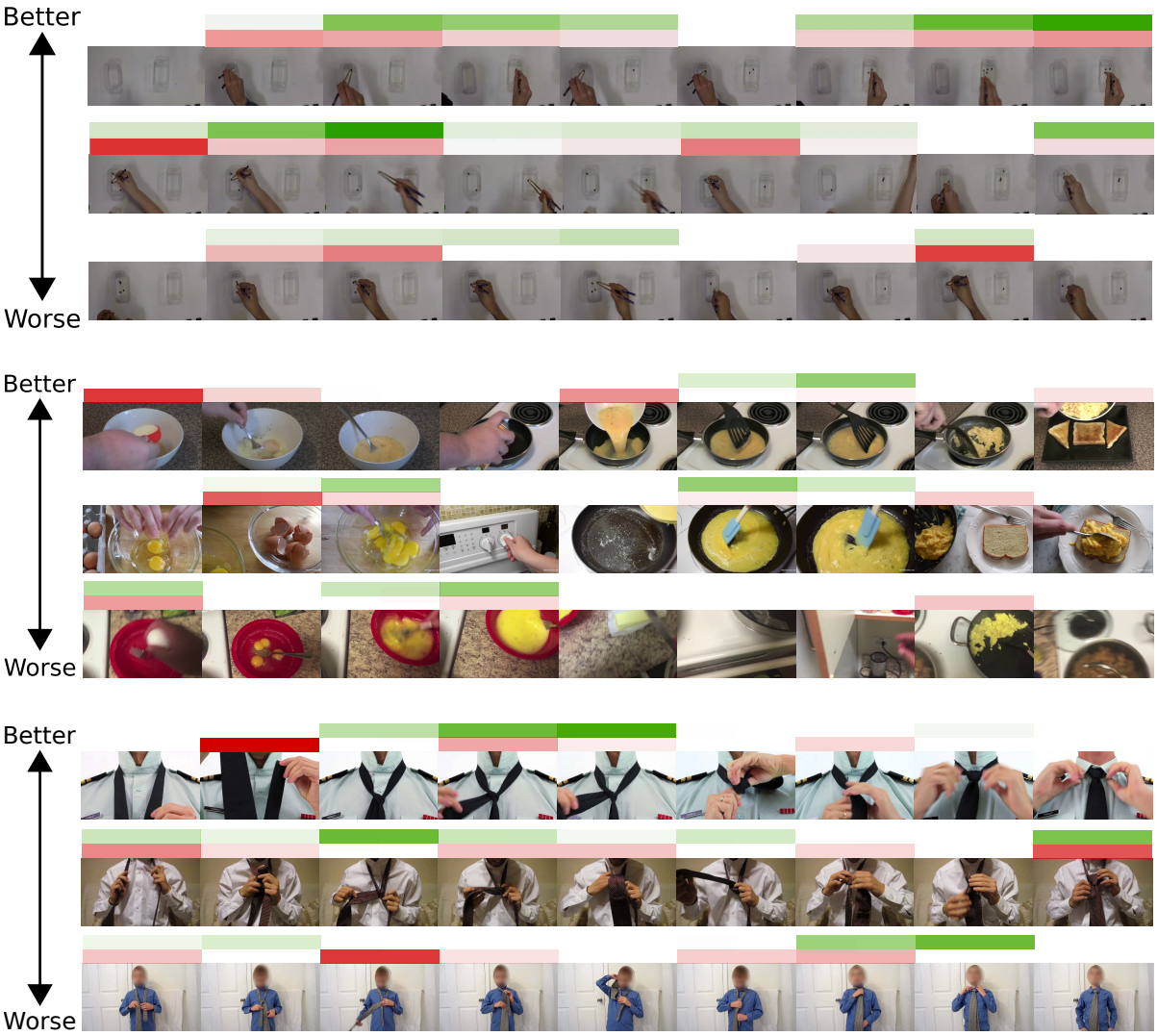

Figure 4.14: *Attention values of the high-skill (green) and low-skill (red) modules with the corresponding video segments for examples from Chopstick Using, Scramble Eggs and Tie Tie. The intensity of the colour indicates the attention value. The videos in each task are ordered using ranking from the combination of high and low-skill branches.*

the high-skill and low-skill attention modules are shown above the corresponding frames of each video. From these examples, several behaviours of the attention can be observed. Firstly, the proposed model is able to filter out irrelevant segments. For instance, turning on the stove-top and opening the cupboard in Scramble Eggs.

Secondly, the rank-aware attention allows the attention modules to focus on different aspects of the video to one another. This is perhaps easiest to observe in the Chopstick Using task. The high-skill module (green) shows increased attention when a bean is being picked up. The low-skill module (shown in red) attends to difficulties in gripping beans.

Similar trends can be seen in the other tasks. In Scramble Eggs, the high-skill module consistently focuses on whisking the eggs in the bowl and gently stirring the mixture in the pan, while the low-skill module attends to adding milk or cream to the eggs and pouring the eggs into the pan. For Tie Tie, the high-skill module gives a high weight to segments displaying a tight inner knot and straightening the tie before folding across, while the low-skill module focuses mainly on hesitation[1] and repetition.

Cases where the filters attend to segments seemingly irrelevant to skill can also be observed; in Scramble Eggs the low-skill module attends to video segments containing bread. This demonstrates a shortcoming of the proposed method. Due to the nature of supervised training and the inability of an attention filter to focus on many different aspects of the video, the model will attend to segments which can be consistently seen across many videos as demonstrating high or low skill. This means less relevant correlations, like attending to the bread, may appear.

A limitation of the proposed approach is that this may also cause rarer mistakes and displays of high skill to be missed. An example can be seen in Chopstick Using in the seventh segment of the second video, where the participant has dropped a bean while moving it between tubs and is now trying to retrieve it. Since this type of mistake was not seen in training, both high and low-skill attention values are low.

Distribution of Attention Values. How the temporal distribution of the attention values vary by task is examined in Figure 4.15. This shows the average temporal attention values per task, over both positive and negative attention modules, for the $T = 400$ temporal segments. Attention values are normalised between 0 and 1 and smoothed with a sliding window of 20.

For some activities, such as Surgery, Scramble Eggs and Braid Hair, the beginning of the task is generally most informative to skill. In Braid Hair, how well the hair is separated and the tightness of the initial braids are particularly useful for determining skill. Although Surgery is repetitive, the early parts of a video tend to be most informative to skill as novices make more mistakes before they become familiar with the control of the robotic arms. Other tasks which consist of the same actions throughout the video, such as Chopstick Using or Drawing, tend to have a more uniform distribution of attention values. Participants may improve in these tasks, however this improvement is less apparent within each video than in Surgery.

[1] This can be seen in the video version of the qualitative results available at https://www.youtube.com/watch?v=ILvowKqiALU.

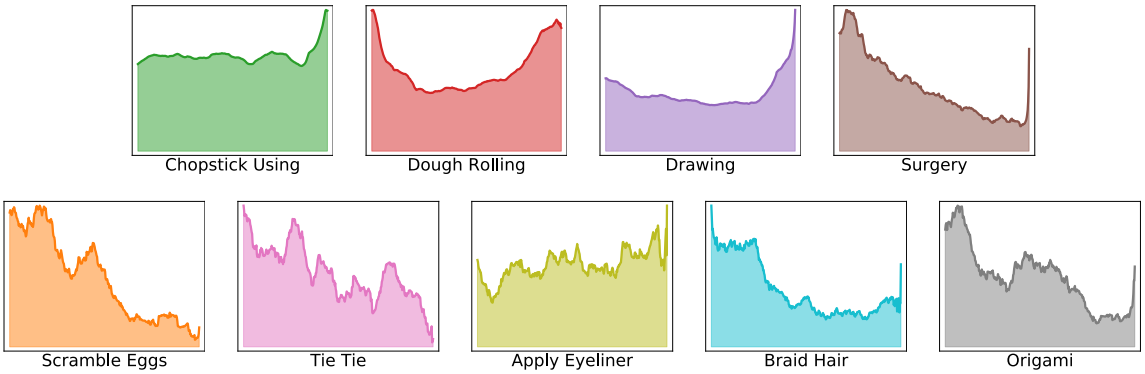

Figure 4.15: *Distribution of attention values over time for each task in EPIC-Skills (top) and BEST (bottom).*

The majority of tasks end with a peak over the final few segments where the end result of the task can seen. As shown by the results in Table 4.3, the last segment alone is not effective for determining skill, however for most tasks it has a useful contribution in combination with other video parts. Scramble Eggs and Tie Tie do not have this peak at the end of the task. In both of these tasks, earlier segments which display the method used to complete the task are more informative than the end result itself.

Attention Filter Examples. Figure 4.16 shows examples of segments with high attention values from an individual low-skill filter. The full 1.6 second segments can be viewed in video form[2], however exemplar frames are also shown here. Within each task, a single low-skill attention filter consistently attends to similar video parts.

In Surgery, one filter attends to strain on the material as a sign of low-skill; the surgery segments in Figure 4.16 show the central stick bending as the knot is tied. In Apply Eyeliner, segments are indicative of mistakes as the frames show eyeliner being removed or covered over. Mistakes are also attended to in Origami, where one filter focuses on the paper being unfolded. For Braid Hair, one low-skill attention filter picks up on messy hair. In Drawing, a low-skill filter attends to the segment where the participant has finished drawing and the end result can be viewed.

Example segments with strong responses from high-skill filters are shown in Figure 4.17. In Apply Eyeliner, the high-skill attention chooses the end result as an informative segment for determining skill. These segments mostly demonstrate neat eyeliner with flared edges extending beyond the end of the eyelid. For Braid Hair, segments which show neat separation of the hair have a strong response from one high-skill attention

[2]Video version of qualitative results available at https://www.youtube.com/watch?v=ILvowKqiALU

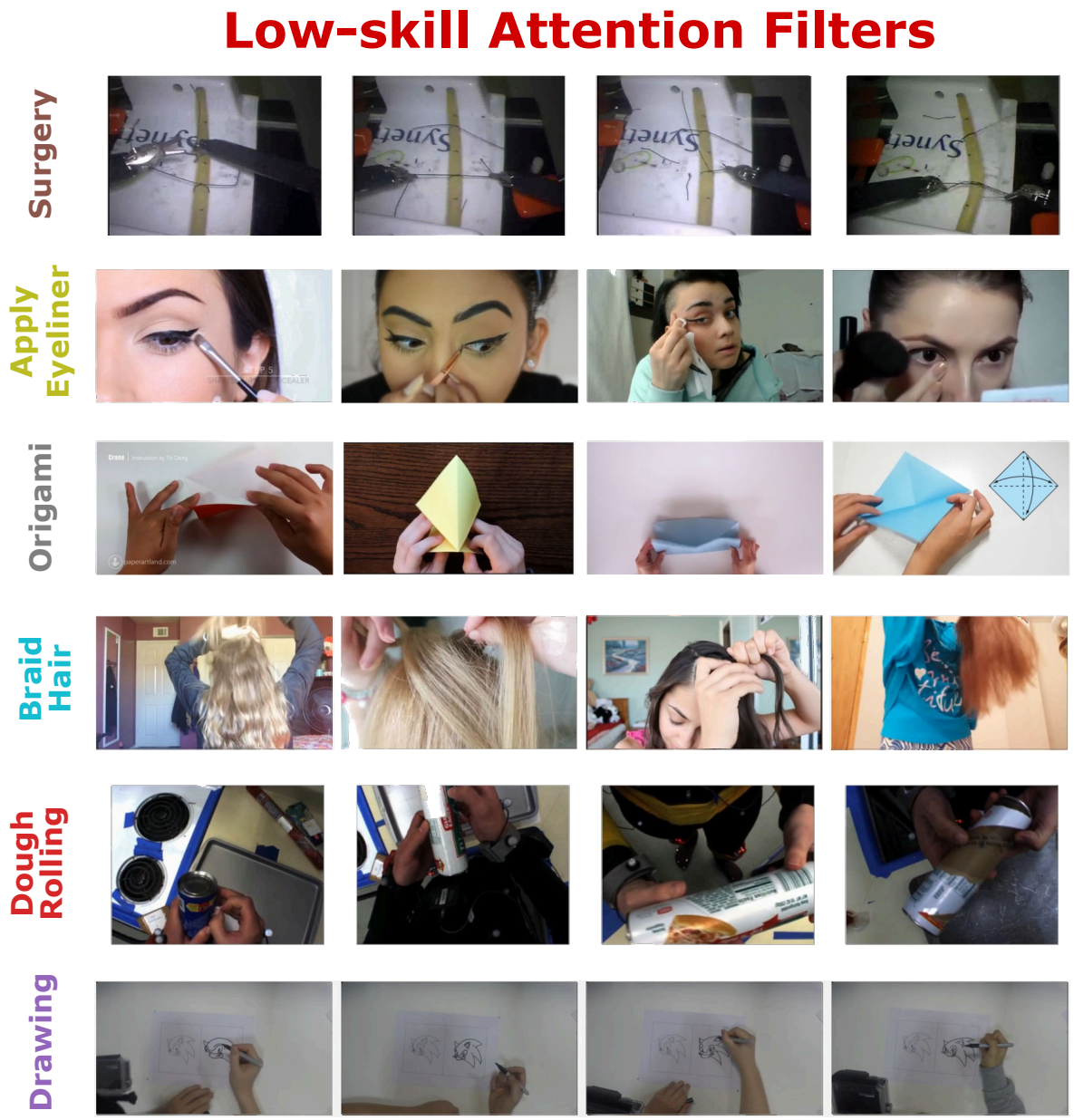

Figure 4.16: *Example video segments attended to by a low-skill attention filter for each task.*

filter. Looping the string to form the knot is attended to in Surgery, where higher ranked videos manage to tie the knot quickly without moving the central pole.

Interestingly, segments common across multiple tasks can be see. In both Dough Rolling and Origami, straightening of the edges is picked up as a sign of high-skill. For Drawing pausing to think, *i.e.* hesitating, is indicative of high-skill. However, hesitation is a sign of low-skill in Dough Rolling, where participants struggle to work out how to open the can of dough.

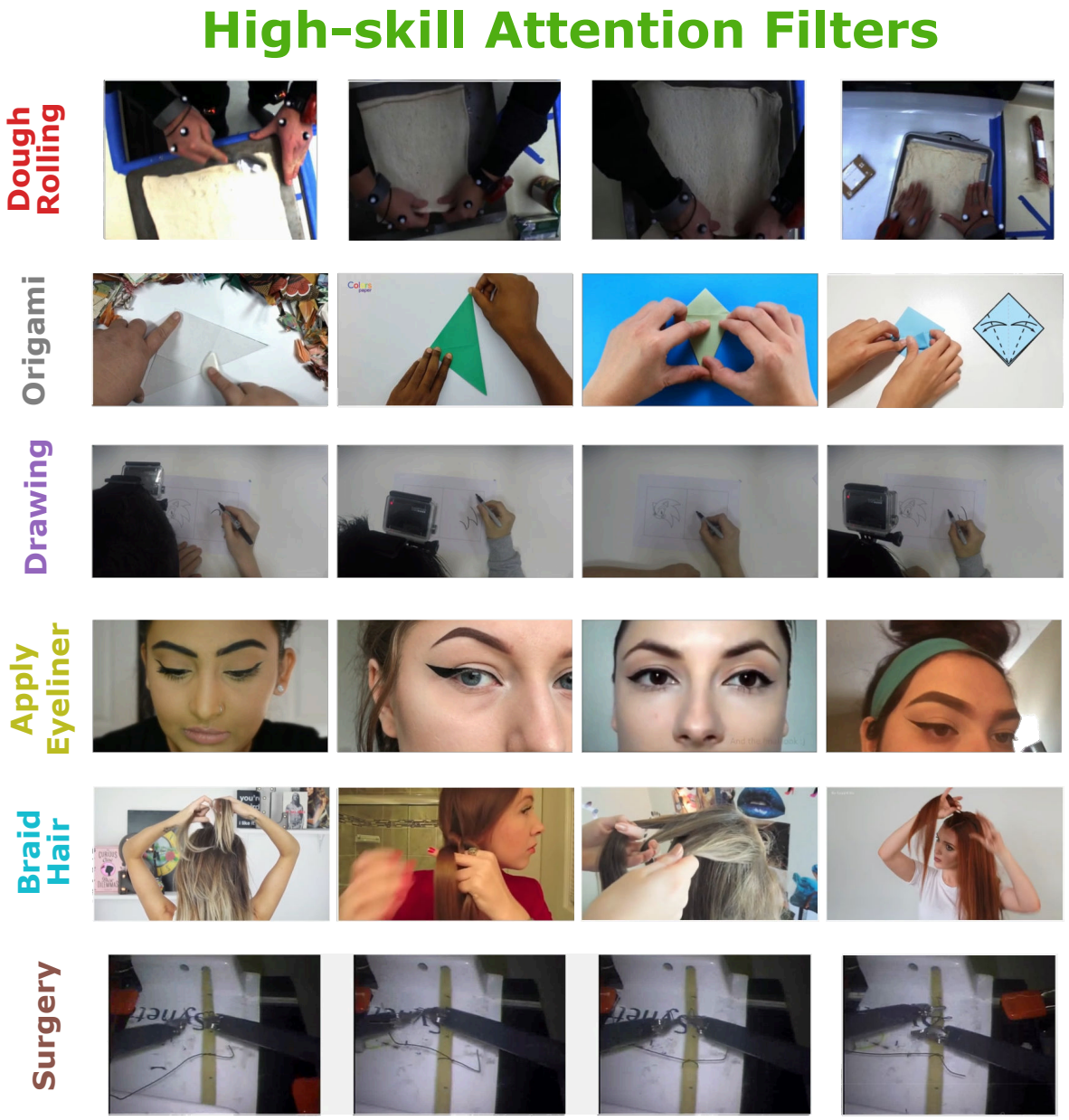

Figure 4.17: *Video segments with maximal response from one high-skill attention filter for various tasks.*

4.5 Conclusion

This chapter has presented rank-aware temporal attention, trained using a novel loss function. The rank-aware loss enables the model to learn the most informative segments for determining the skill shown in a video. This chapter has also proposed the disparity loss, which directly optimises the attention to pick more informative segments than the uniform weighting. This solves the instability in optimising the standard softmax attention in a ranking framework.

The proposed method has been tested on two datasets: EPIC-Skills and BEST, the latter of which was also presented in this chapter. This new dataset allows for further exploration of skill determination in daily-living with longer and more complex tasks. It can also enable future work which aims to discover similarities between skill in different tasks. The proposed method outperforms existing temporal attention methods as well as the method from Chapter 3, the only previous method for skill determination in daily-living tasks. With an average performance of over 80% in both datasets and an increase by as much as 12% on individual tasks, it can be concluded that the new method is useful for skill determination and is able to attend to informative parts of the video. However, there are still a few limitations of the proposed approach which are outlined below.

Temporal Dependency. This chapter has proposed an attention-based approach which re-weights the importance of video segments based on their content. However, it predicts attention with only local context from each individual segment and does not take into account the importance of that segment in regards to the others attended to. Furthermore, an action may be considered indicative of high or low skill depending on when it occurs in relation to other events in the task. For instance, in Scramble Eggs the point at which milk or salt is added can have an effect on the outcome. Thus, future works could explore how to incorporate temporal dependencies to determine skill.

Task Complexity. The proposed method improves the performance for many tasks, however it struggles to find the most informative segments in tasks such as Surgery, where movements are subtle and in a different domain to the pre-training datasets. The proposed method uses pre-extracted I3D features to avoid issues with training on small amounts of data, however these features are not suitable for all tasks. Further work is needed to be able to learn and extract the best features for each task, while still being able to learn with relatively little data per task.

Despite the proposed method being designed to cope with long videos, it struggles with the longest task, Origami, where videos are up to 10 minutes in length. Future work could investigate how to better cope with tasks of varying lengths and remove the need for a set number of pre-extracted features.

Feedback and Explainability. This chapter takes a step towards the goal of providing users with feedback, by identifying the video parts particularly relevant to the skill ranking. However, it can be hard to interpret what the model is picking up on within these segments and what improvements could be made to increase skill. More work is needed in this area to achieve further explainability and be able to provide feedback to users

identified as having low-skill. Further explainability could be achieved by incorporating spatial attention, so the particular motions and attributes the model is picking up on can be identified. One way to provide feedback could be to retrieve the same action in a higher ranked video without the same mistake. Alternatively, future work could study how to learn to provide automatic text-based feedback for segments identified as low-skill. Chapter 6 will look at using narrations in instructional videos to identify how an individual step should be performed and determine whether it has been performed in this way.

Chapter 5

Knowledge Transfer for Determining Skill in Novel Tasks

Chapters 3 and 4 both presented methods to rank videos of daily-living tasks in terms of skill. As demonstrated in the previous chapters, the presented methods are applicable to a wide variety of different tasks. The methods do not rely on prior knowledge about the tasks however, both methods are trained on tasks individually to learn task-specific network weights. Chapter 4 also incorporates attention to learn the relevance of video parts, again for each task individually. This approach of training one network per task (or group of tasks) and requiring a sizable number of annotations for each task limits the scalability of skill determination methods. It would be ideal to be able to utilise previously learnt knowledge from skill determination on other tasks to adapt to a new task with relatively little (or optimally no) labelled data.

This chapter investigates to what degree sharing is beneficial between different tasks for skill determination and whether it is possible to transfer information from previously seen tasks to rank videos in a new task. Specifically, Section 5.1 introduces the concept of multi-task learning in skill determination and explores the situations when this is and isn't effective. Section 5.2 then establishes that, in some cases, information can be transferred to new skill tasks with little or no annotated data. However, it is hard to predict when these transfers are possible. The difficulties with sharing features and transferring information to new skill determination tasks are then discussed in Section 5.3.

5.1 Multi-task Learning

Classically, machine learning has focussed on learning a function to solve one problem on a particular dataset. To learn a function for another problem, further annotations or new data are collected and the model is retrained. Prior works have shown that it is beneficial to learn to solve related problems jointly and share the learnt representations [21, 212]. For instance, explicitly learning to predict edges may help a model learn to segment objects in a scene. This is known as *multi-task learning*.

In skill determination, a 'task' refers to an activity in which skill can be ranked, such as tying a tie or using chopsticks. The end goal for each task is to rank the videos of that task in accordance with the skill they display, but the criteria for determining skill may differ across the different tasks. Thus, the goal of multi-task learning in skill determination is to learn common features to rank skill from videos of multiple tasks. This is distinct from the usual multi-task learning set-up, where the aim is to learn a generalisable feature representation which can be applied to different computer vision problems on the same data.

This section will explore whether jointly learning to rank multiple skill tasks simultaneously is beneficial for the daily-living tasks in EPIC-Skills (Section 3.2) and BEST (Section 4.1). The methods presented in Chapters 3 and 4 will be used to investigate this. Section 5.1.1 will first report results when all tasks in a dataset are learnt jointly. Section 5.1.2 then considers the assumption that similar tasks share skill-relevant features and demonstrates that it is effective to learn the subtasks within Surgery and Drawing jointly.

5.1.1 Sharing All Tasks

The skill tasks within the EPIC-Skills and BEST datasets are deliberately diverse in their content as this allows skill determination methods to be more thoroughly tested. However, the current method of training tasks individually is not scalable and is a hurdle for extending skill determination to further daily-living tasks. Therefore, this section investigates how necessary the separate training of tasks is by jointly training tasks within EPIC-Skills and BEST.

These experiments use the same training and test splits for both datasets as the previous chapters. That is, four-fold cross-validation for EPIC-Skills and a single training and test split for BEST which contain 75 and 25 videos per task respectively. A single network learns to determine skill using video pairs from all tasks. There are no pairs of videos between different tasks and the rank across tasks is assumed meaningless. Batches are

	Surgery			Dough Rolling			Drawing			Chopstick Using		
	S	T	TS	S	T	TS	S	T	TS	S	T	TS
Separate	66.4	72.5	70.2	79.5	79.5	79.4	77.6	82.7	83.2	70.8	70.6	71.5
Joint	65.9	73.3	69.9	77.5	76.3	77.8	76.7	76.5	76.7	66.0	68.7	69.4

Table 5.1: *Results of four-fold cross validation on the tasks in EPIC-Skills with tasks either learnt jointly, with a single network, or separately, with each task having its own network. Results are shown for the spatial (S), temporal (T) and two-stream (TS) end-to-end models from Chapter 3.*

	Scramble Eggs			Tie Tie			Apply Eyeliner			Braid Hair			Origami		
	S	T	TS	S	T	TS	S	T	TS	S	T	TS	S	T	TS
Separate	67.5	77.3	77.3	74.7	80.7	82.4	81.3	68.4	76.2	62.4	71.0	70.5	67.5	71.2	72.7
Joint	**67.5**	68.8	69.5	66.5	83.7	83.7	76.2	74.9	76.6	73.1	56.8	66.3	64.2	68.4	69.8

Table 5.2: *Comparison of pairwise accuracy when learning tasks in the BEST dataset jointly or separately with the end-to-end model proposed in Chapter 3.*

balanced such that each batch contains an equal number of video pairs from each task. Otherwise, the hyperparameters used are also the same as in the previous chapters and pairwise accuracy is again used to evaluate performance.

Both the end-to-end method of Chapter 3 and the Rank-aware attention method proposed in Chapter 4 are tested. With the latter method sharing of the attention is tested separately to sharing the ranking function as the parts pertinent to determining skill are generally quite task specific and therefore may require individual attention modules. However, there may be common features which indicate skill in different tasks.

End-to-End Multi-task Learning Results

Tables 5.1 and 5.2 report results of training tasks within a dataset jointly in EPIC-Skills and BEST respectively. These results are obtained using the end-to-end method proposed in Chapter 3 and are compared to separately training each task with its own network. For EPIC-Skills none of the tasks benefit from joint training. The amount to which jointly training hurts the result is dependent on the task and modality, with some tasks like Surgery performing comparably to individual task training. In other cases there is a decrease in results. This is largest in the temporal stream of Drawing (-6.2%) and the spatial stream of Chopstick Using (-4.8%).

In BEST the majority of tasks also do not benefit from joint training. Apply Eyeliner and

Tie Tie have comparable two-stream performance, while the accuracy drops for Scramble Eggs, Braid Hair and Origami. Again, the amount each task is affected differs between the modalities, although the majority of results on individual modalities also decrease. One exception to this is Braid Hair, which has a significant improvement in the spatial stream (+10.7%). Without further inspection of which features are shared between tasks it is hard to conclude why this happens as Braid Hair is not visually similar to any other task. There may be some similarities in motions between Braid Hair and Tie Tie as both involve neat looping of separate components, however this information would likely be in the temporal stream, which decreases in performance with joint training.

Multi-task Learning Results with Rank-aware Attention Networks

It is also interesting to know to what degree attention can be shared between tasks and whether using separate attention for each task can enable feature sharing in the ranking layers. Figure 5.1 compares three different sharing settings:

- **Separate**: completely separate rank-aware attention network weights are learnt per task, as was done in Chapter 4
- **Shared Ranker**: a rank-aware attention network where the ranking function is shared across all tasks (for both the high-skill and low-skill network branches), but separate attention modules are learnt for each task.
- **Shared Attention and Ranker**: both the attention modules and ranking functions are shared between all tasks.

In general, sharing the ranker causes a drop in performance as with the method from Chapter 3. Unlike the end-to-end method from Chapter 3, the rank-aware attention network learns a ranking on top of pre-extracted features. This network has much less capacity than the end-to-end method tested above, which is likely the reason for the larger drop in performance as the network no longer has enough capacity to learn separate ranking criteria for each task within one model.

Sharing the attention in addition to the ranker causes a further drop in performance. For the tasks in EPIC-Skills, the sharing of the ranker causes a bigger drop than sharing the attention. In BEST, both cause a sizable drop as the tasks in this dataset are more complex and thus more reliant on the attention to discover the key video parts. From these preliminary results it can be concluded that it is not beneficial to learn all available tasks jointly, instead task-specific features are required when the tasks are diverse.

These overall trends are not true for Origami, where performance increases when sharing the ranker. Interestingly, in the separate attention and shared ranker model, the per-

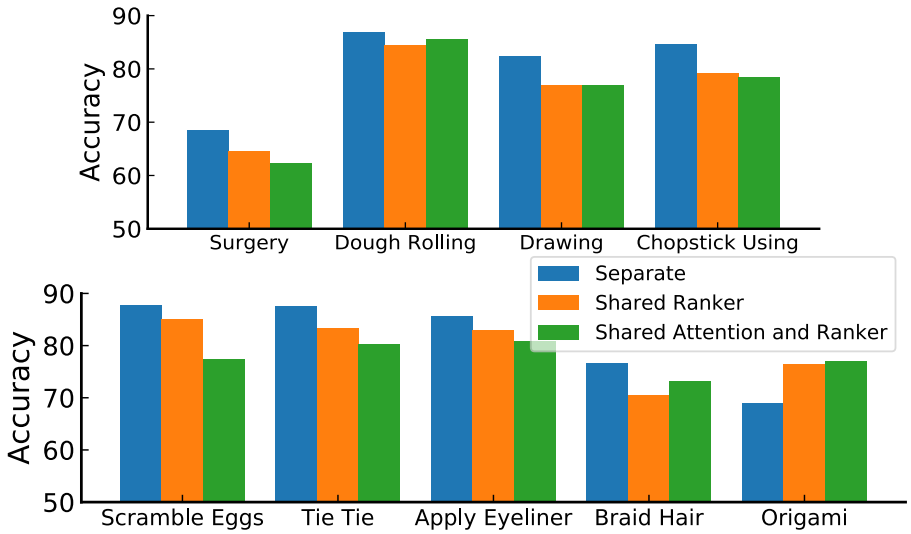

Figure 5.1: *The effect of sharing the ranking layers and attention modules in the rank-aware attention network. Tasks within EPIC-Skills (top) and BEST (bottom) are trained jointly.*

formance for Origami gains a further 1% improvement if the attention learnt for Braid Hair is used. All other tasks perform considerably better when using their own attention modules. Origami is the most challenging task for the rank-aware attention network due to the differences in skill being visually very subtle. Whilst other tasks perform better with task specific features, the features learnt for Origami struggle to rank the videos correctly. This task therefore benefits most from more general skill determination features learnt by training tasks together.

5.1.2 Sharing Related Tasks

While the previous experiments show jointly training a variety of distinct skill tasks is not beneficial, this thesis has thus far assumed that highly related tasks can be trained jointly. For instance, the Surgery and Drawing tasks within EPIC-Skills each contain multiple subtasks. Surgery consists of Knot Tying, Needle Passing and Suturing (see Figure 2.9), while Drawing contains a Sonic Drawing subtask and a Hand Drawing subtask (shown in Figure 5.2). It seems reasonable to assume that these subtasks can be trained jointly, as they are recorded in the same environment and skill performances share many high-level properties. This is highlighted in the Surgery tasks from JIGSAWS [48], where experts use the same OSATS criteria [108] to grade performances of the different subtasks.

This section verifies that the subtasks within both Drawing and the Surgery can be trained jointly. The rank-aware attention network is used to test this as it is generally the best performing model and clearly established the benefit of training unrelated tasks

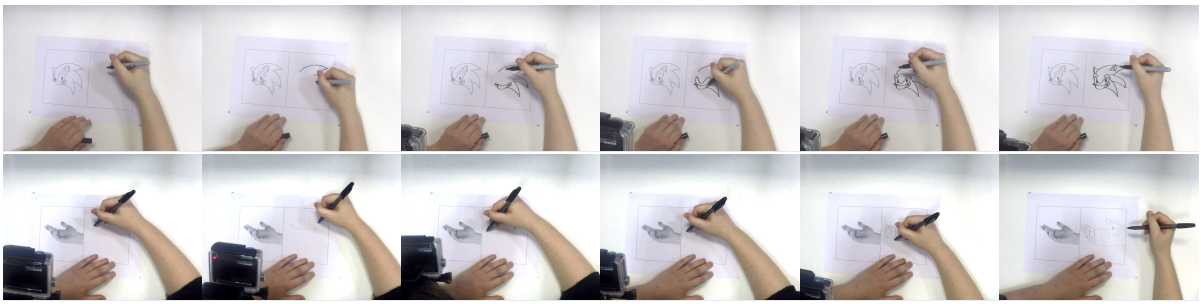

Figure 5.2: *The Sonic (top) and Hand (bottom) Drawing subtasks.*

separately in the previous section. If training the subtasks within Drawing and Surgery jointly limits the performance, this model will demonstrate it. The rank-aware attention network also offers the opportunity to test which aspects of the model are beneficial to share, *i.e.* the attention and/or the ranking.

The same three settings as in the previous section are tested: separate, shared ranker and shared attention and ranker. Figure 5.3 shows the results for the three subtasks within Surgery. The Surgery tasks generally benefit from sharing with Needle Passing and Suturing increasing in performance. This increase comes from sharing the ranking layers. Knot Tying however decreases in performance when sharing the ranking layers. Figure 5.4 shows a similar pattern for Drawing, with the fully shared model being better than training each subtask with a separate model.

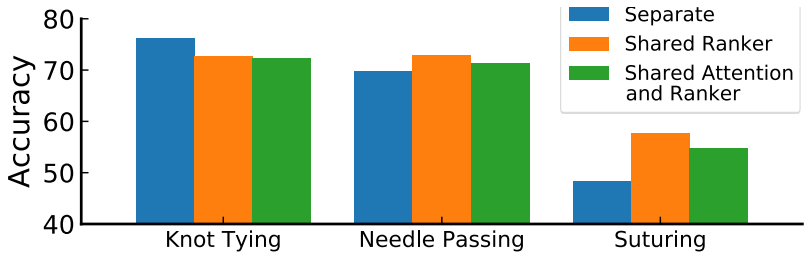

Figure 5.3: *The pairwise accuracy for subtasks within Surgery when training tasks separately or with shared components in the rank-aware attention network.*

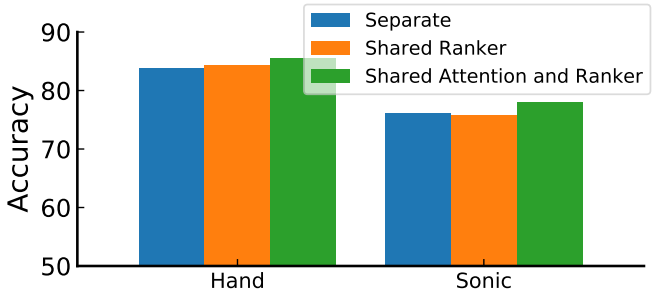

Figure 5.4: *Comparison of the accuracy for subtasks within Drawing when training tasks separately or with shared components.*

5.1.3 Multi-task Learning Conclusion

This section has demonstrated that joint training can be helpful for highly related tasks recorded in similar environments. The amount of annotated data available for a single skill task is limited, thus training with tasks which share properties can help a model learn more generalisable features.

For unrelated tasks, joint training is generally not beneficial and leads to a degradation in performance. There are some cases where tasks can benefit from joint training with seemingly unrelated tasks. For instance, Origami in the rank-aware attention network or Braid Hair in Chapter 3's end-to-end network. However, it not obvious which tasks will benefit and the improvements are not consistent over different models. Therefore, with the current data available, it is recommended to train tasks separately.

These tasks may share some high-level features such as fluidity of movement or neatness, however it is hard to learn these with the limited amount of data available in each task. Such properties will manifest themselves in different ways in different tasks, meaning the models will instead learn task specific features which are irrelevant to determining skill in other tasks. It may be possible to learn general features applicable to skill determination in many tasks, however many more labelled tasks would be needed.

A more viable approach would be to share parameters within subsets of tasks and to learn which features are shareable and which are task-specific. Future work which learns to predict which tasks are relevant to one another would therefore be very valuable. This could be investigated in multi-task learning with soft-parameter sharing methods [100, 103, 114] which aim to learn which tasks or parameters can be shared, as opposed to the hard parameter sharing investigated in this section.

5.2 Transfer Learning

Training individual models for each skill task is only one obstruction of applying skill determination to a greater number of daily-living tasks. Another obstruction is the need for a sizable amount of labelled data per task. It is therefore necessary to reduce the amount of labelled data required when learning to determine skill for a new task. One way to do this is by transferring knowledge from previously collected tasks to a new related task. This is known as *transfer learning*.

The most widely used form of transfer learning is fine-tuning, where a network is initialised with weights learnt for a different problem on another, often larger, dataset. Chapter 3 used this technique by initialising the TSN network with weights trained on

ImageNet [32]. Fine-tuning is usually done with a large number of labelled samples, however it can also be studied in *few-shot* learning scenarios where there are only a handful of examples available for adapting a model. Another possibility is the *zero-shot* setting where the transfer is performed without using any examples to adapt the model.

In the context of skill determination, the aim is to transfer knowledge from learnt rankings in one or more known (*source*) tasks to predict a ranking in a previously unseen (*target*) task. This section will investigate whether this is possible with current models and datasets in two ways. First zero-shot transfers between tasks in EPIC-Skills and BEST will be studied in Section 5.2.1. While information can be transferred between tasks, this section will demonstrate that these transfers are hard to predict and not consistent between models. Section 5.2.2 will then summarise recent work in zero-shot transfers on the shorter, more related AQA-7 dataset [135]. Second, the problem will be reformulated as a meta-learning problem in Section 5.2.3 and experiments will be performed to demonstrate that it is incredibly difficult to learn generalisable features for skill determination, even with highly related tasks.

5.2.1 Zero-shot

This section examines whether it is possible to determine skill in a new task with zero-shot transfers *i.e.* directly applying a learnt model to the new task without fine-tuning. Figure 5.5 shows the results of this type of zero-shot transfer for both EPIC-Skills and BEST when using the rank-aware attention network from Chapter 4. Models are trained on a single source task before being applied to a target task. Note that the performance in these tasks is measured with pairwise accuracy, thus random performance is 50%.

There are several zero-shot transfers between tasks with surprisingly high performance. For instance, Chopstick Using transfers well to Dough Rolling and vice-versa. In BEST, the model trained on Origami transfers particularly well to Tie Tie and also to Scramble Eggs. However, not all transfers are symmetrical and the reverse is not true for these transfers. A good transfer does indicate that the two tasks share some common properties which can indicate skill. Although, it is not guaranteed that a model will learn these properties as opposed to some other task-specific features.

Figure 5.5 also shows a large number of negative transfers. These are the results below 50%, where the zero-shot transfer gives a performance worse than random. Since skill determination is a ranking problem, these negative transfers could still be useful. Negative transfers indicate that the trained model does have features useful for separating the videos in the new task based on skill, although they give the reverse ranking.

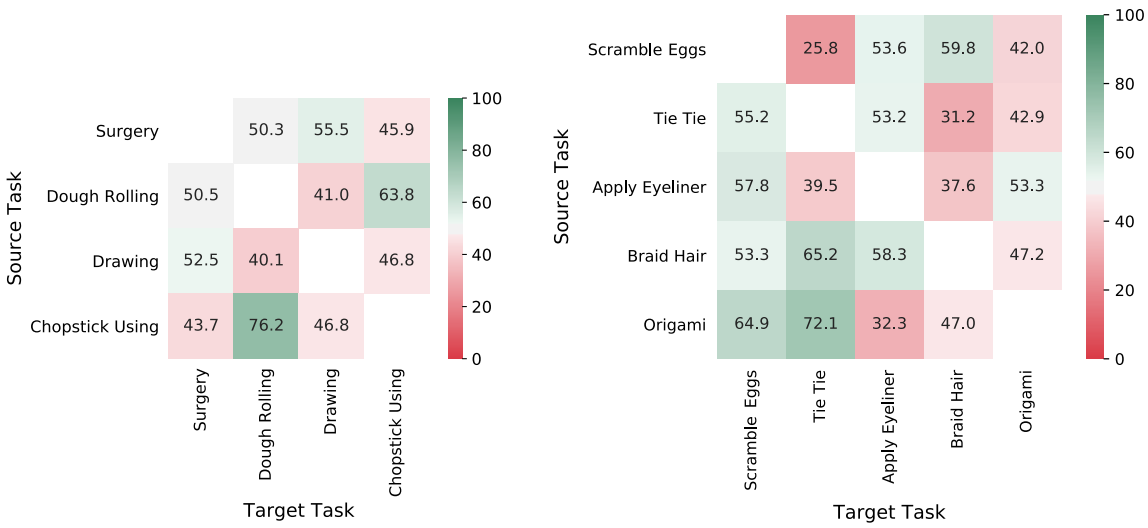

Figure 5.5: *Zero-shot transfer between individual tasks in EPIC-Skills (left) and BEST (right) with the rank-aware attention network.*

To compare differences in transferability between models, Figure 5.6 shows the zero-shot task transfer results for BEST with the end-to-end method proposed in Chapter 3. The uniform baseline from Chapter 4 is also tested as this could give better understanding of the effect of applying learnt attention to a new task. In general, the end-to-end method from Chapter 3 has more positive transfers, but also more transfers close to the random 50% result. The successful transfers are also quite different between the three models. With Tie Tie as the source task, the features learnt with the method from Chapter 3 are much more transferable. The direction of the transfer can also differ between models. The Origami to Apply Eyeliner transfer is highly negative with the rank-aware attention network and uniform baseline, but positive in with Chapter 3's method.

In summary, these results demonstrate that there is potential for transferring information between skill tasks. Some of the transfers are surprising, with no obvious relationship between the tasks which transfer well. Further work is therefore needed to understand the relationships between tasks by examining the features which cause the positive and negative transfers. With a larger number of skill tasks, it may be possible to build a taskonomy [212] of skill tasks which identifies the relationships between tasks. The above results have also highlighted the issue of negative transfer in skill, where features may have a strong negative correlation with the ground-truth ranking. It is likely that within a task there is a mix of positive and negative transfers from different features, as well as many features irrelevant to the target task. Without any labelled data it would be difficult to identify whether the correlation is positive or negative, thus few-shot transfer between tasks seems the most promising direction.

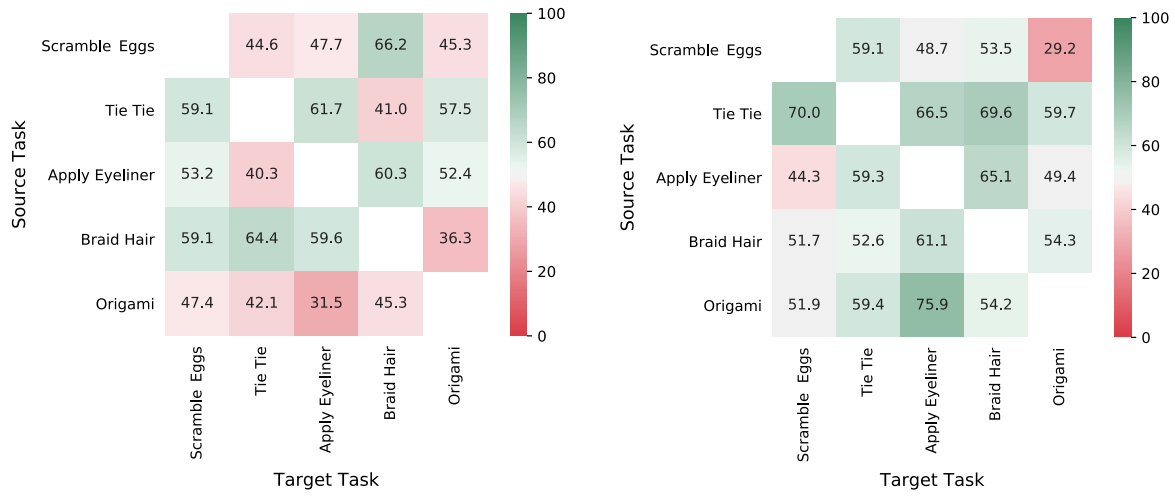

Figure 5.6: *Zero-shot transfer between individual tasks in BEST for the uniform weight baseline from Chapter 4 (left) and the end-to-end model proposed in Chapter 3.*

5.2.2 Related Work on AQA-7

In their recent work on action quality assessment, Parmar and Morris [133] also examined whether knowledge transfer was possible between six different Olympic sports in the AQA-7 dataset. A more detailed summary of the dataset can be found in Chapter 2. The six sports studied are: Diving (individual 10m), Synchronised Diving 3m, Synchronised Diving 10m, Gymnastics Vault, Ski Big Air and Snowboard Big Air. The tasks have many similar properties. All of the tasks involve some sort of launch into the air where the participant(s) have to perform a sequence of somersaults and twists before successfully landing. The performance in each task is judged by the how well the sequence of somersaults and twists were performed and the difficulty of that particular sequence, although the aggregation of these aspects may differ between tasks.

Since these tasks are only seconds in length and are more highly related than the daily-living tasks examined in this thesis, several of the issues with transferring information between tasks are reduced. The shorter duration of the tasks means temporal attention is no longer needed and thus it is more likely existing transfer learning methods can pass useful information between these tasks. The more related nature of the tasks also increases the potential for successful transfers. This is evidenced by experiments in multi-task learning where Parmar and Morris found that jointly training on multiple tasks with their C3D-LSTM model was beneficial to the majority of tasks.

Parmar and Morris also perform zero-shot transfer experiments shown in Table 5.3. Since the tasks all have ground-truth scores from Olympic judges they are trained in

Target → Source ↓	Diving	Gym Vault	Ski	Snowboard	Sync. Dive 3m	Sync. Dive 10m
Diving	**0.6997**	-0.0162	0.0425	0.0172	0.2337	0.0221
Gym Vault	0.0906	**0.8472**	0.0517	0.0418	-0.1642	-0.3200
Ski	0.2653	-0.1856	**0.6711**	0.1807	0.1195	0.2858
Snowboard	0.2115	-0.2154	0.3314	**0.6294**	0.0945	0.1818
Sync. Dive 3m	0.1500	-0.0066	-0.0494	-0.1102	**0.8084**	0.0428
Sync. Dive 10m	0.0767	-0.1842	0.0679	0.0360	0.4374	**0.7397**
Multi-task	0.2258	0.0538	0.0139	0.2259	0.3517	0.3512

Table 5.3: *Zero-shot Transfer Between Tasks in AQA-7. Correlation between the ground-truth and predicted scores is measured with Spearman's rank correlation which ranges between -1 and 1, with 0 indicating no correlation. Results from [133].*

a regression framework and evaluated using Spearman's rank correlation. This metric ranges between -1 and 1, with 0 indicating no correlation between the predicted and ground-truth ranking. Transfers seem to be most successful between the tasks which are performed in the same environment. For instance, there is a positive result when transferring from Ski Big Air to Snowboard Big Air and vice-versa. There are also some positive transfers between diving tasks: models trained on Diving and Synchronised Diving 10m both give a positive Spearman's rank correlation when testing on Synchronised Diving 3m. Table 5.3 shows that transferring from a multi-task model works reasonably well, often performing better than individual transfers. However, when there is a particularly good transfer existing between individual tasks, as in the case of Synchronised Diving 10m to Synchronised Diving 3m, the multi-task transfer is worse.

Despite the tasks within AQA-7 being more related to each other than the tasks within EPIC-Skills and BEST, it is still not possible to naively transfer information between any but the most related tasks without any labelled data. This is evident from the Gym Vault task where models trained on any of the other tasks give negative Spearman's rank correlation. There are also some surprising results between highly related tasks. While Synchronised Diving 10m transfers well to Synchronised Diving 3m, the reverse is not true. The asymmetry of this relationship demonstrates how hard it is to determine the relatedness of tasks without better insight into the features the model is learning.

Parmar and Morris also experiment with fine-tuning these multi-task models to new tasks. They find fine-tuning from the model trained on the other five tasks is generally better than training from a random initialisation, although with 25 or more video being used for fine-tuning, this still requires a sizable annotation effort per task.

Parmar and Morris demonstrated AQA-7 to be a promising starting point to study the transferability between skill tasks. However, there are many opportunities for further work. Further understanding of the features which are transferred between tasks is needed to discern why zero-shot transfers such as Synchronised Diving 3m to Synchronised Diving 10m are unsuccessful. While fine-tuning from other skill tasks was shown to be helpful, it would be necessary to use less labelled data to be able to transfer between a wider variety of tasks. It is also clear that being able to learn more generalisable features in a multi-task model would be helpful as the zero-shot results on Gym Vault show the model struggles with anything but the most related tasks.

5.2.3 Meta-learning

This section explores whether meta-learning can be used to learn more generalisable features and better transfer information between tasks in a few-shot setting.

Meta Learning Overview

Meta learning, often also referred to as learning to learn, aims to learn a model which is capable of generalising or easily adapting to new tasks. To achieve this, meta-learning models are trained over a variety of different learning tasks and optimised for the best performance over the distribution of tasks, rather than on an individual task. The goal is outlined with the following equation:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}[\mathcal{L}_\theta(\mathcal{T})] \tag{5.1}$$

where $\mathcal{T}$ is an individual task, $p(\mathcal{T})$ is the distribution of tasks, $\mathcal{L}$ is the loss and θ is the possible model parameters. This looks quite similar to a standard learning problem, except the optimisation occurs over tasks not individual data samples.

There are several different approaches to meta-learning: model-based [121, 156], metric-based [169, 174, 186] and optimisation-based [45, 149]. Metric-based meta-learning methods aim to learn a good distance function during training. An example is Prototypical Networks, which learn an embedding space for classification where few-shot classes can be recognised by their distance to the prototype of a class. These prototypes are calculated from the mean of the embedded examples for a few-shot class. Optimisation-based approaches aim to adjust the optimisation algorithm so that the model is able to better learn from few examples, something which gradient methods cope poorly with. Model-based meta-learning uses models specifically designed to update their parameters quickly with only a few training steps. For instance, Meta Networks [121] combine two different

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \{x^{(j)}, y^{(j)}\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 5.2
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample datapoints $\mathcal{D}_i' = \{x^{(j)}, y^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 5.2
end while

Algorithm 5.1: *MAML for Few-Shot Supervised Learning taken from [45]*

types of weights. Fast weights are predicted by a network per task and slow weights are trained with REINFORCE [175] across all tasks. The network also stores the fast weights in an external memory module.

A good distance function is problem dependent and as the majority of methods have been designed for classification problems, thus metric-based approaches are not applicable for skill determination. From model-based and optimisation-based approaches, MAML [45] is chosen, a popular optimisation-based approach, as model-based methods usually struggle to generalise to out-of-distribution tasks [44]. MAML (Model-Agnostic Meta-Learning) aims to learn a good initialisation of a model's parameters so a new task can be learnt from only a few examples and a small number of gradient updates.

Applying MAML to Skill Determination

The MAML algorithm for few-shot learning is shown in Algorithm 5.1. The basic idea is to find the optimal model parameters θ which achieve good generalisation across a variety of tasks. These parameters should make task-specific fine-tuning more efficient. In the inner loop, K examples are selected per task $\mathcal{T}_i$ and the parameters θ are adapted over several iterations to task-specific parameters θ_i. After training on the individual tasks, the parameters θ are updated in the meta-update phase. This update uses the error of the task specific parameters on new data points sampled from each task. To adapt to a target task $\mathcal{T}$, K data points are sampled from that task and the model is fine-tuned before testing.

This section looks specifically at applying MAML to the tasks in AQA-7. These tasks are more similar to each other than the tasks within EPIC-Skills and BEST, thus it is more likely a general initialisation can be found where features can be easily adapted to

determine skill in a new task. If it is not possible in AQA-7, it is highly unlikely to be possible in videos of longer, less related, daily-living tasks. Using AQA-7 also removes the issue of how to incorporate temporal attention in a meta-learning framework which this thesis leaves for future work.

In Algorithm 5.1, $\mathcal{T}$ is the set of source skill tasks, *i.e.* if the aim is to adapt to Gym Vault as a target task, Diving, Ski Big Air, Snowboard Big Air, Synchronised Diving 3m and Synchronised Diving 10m are the source tasks. The AQA-7 dataset uses Olympic judges scores to label a video's skill, thus a data point $\mathcal{D} = \{x^{(j)}, y^{(j)}\}$ consists of a video $x^{(j)}$ and skill score $y^{(j)}$. The model f consists of two fully connected layers on top of pre-extracted C3D features as temporally averaging the features obtains better performance than the C3D-LSTM model [133, 134]. Mean Squared Error (MSE) is used to obtain the gradients:

$$\mathcal{L}_{\mathcal{T}_i}(f_\theta) = \sum_{x^{(j)}, y^{(j)} \sim \mathcal{T}_i} \left\| f_\theta(x^{(j)}) - y^{(j)} \right\|_2^2 \tag{5.2}$$

Experiment Details

C3D features are extracted from the fc6 layer of a model trained on Sports1M [182]. A feature is extracted from each 16 frame chunk of a video, thus each video has six 4096-dimensional features. The network trained on top of these features consists of two fully connected layers where the first has an output size of 256. The meta learning rate β is 1×10^{-5} and the base learning rate α is 0.001. $K = 8$ datapoints are sampled from every task in each iteration.

Results for each task are reported on the standard AQA-7 test set. None of these videos are seen in the training or meta update phases. Training is performed by sampling videos from the standard training set of the five remaining tasks. As in the previous section, performance on AQA-7 is reported using Spearman's rank correlation.

Results and Discussion

Table 5.4 shows results when using MAML for few-shot transfer learning in skill determination. The first key takeaway is that the results for MAML with and without this fine-tuning, *i.e.* few-shot versus zero-shot, are comparable. This demonstrates that the features learnt by MAML are not easily adaptable to new skill determination tasks with only a few examples. Skill determination is a much more fine-grained task than the typical 5-way classification tasks performed with meta-learning. It is also likely that the performance is dependent on the set of examples chosen, as the features which separate

Method	Fine-tuning?	Diving	Gym Vault	Ski	Snowboard	Sync. Dive 3m	Sync. Dive 10m	Avg.
Multi-task [133]		0.226	0.054	0.014	0.226	0.352	0.351	0.204
Multi-task (re-imp.)		**0.002**	**0.115**	0.048	0.074	0.424	**0.487**	0.192
MAML		0	0	0.297	0.297	**0.5**	0.376	**0.245**
MAML	✓	-0.001	0	**0.316**	**0.309**	0.486	0.338	0.241

Table 5.4: *Results of meta-learning to adapt to new tasks on the AQA-7 dataset using MAML. This is compared to MAML without fine-tuning and the re-implementation of the multi-task model used for zero-shot transfer by Parmar and Morris [133].*

these examples need to be applicable to the rest of the task's ranking. Future work could explore this variability and predict the best videos for fine-tuning.

These results are also compared to a naive zero-shot transfer from a multi-task model. This multi-task baseline is a re-implementation of the multi-task zero-shot transfer performed by Parmar and Morris in Table 5.3, using the same network structure and extracted C3D features as used in the MAML results.

Overall, zero-shot transfer with MAML performs better than with the multi-task model, although the results vary considerably with individual tasks. The results per task are quite different from the multi-task model, demonstrating that the meta-learning has quite a large effect on the learnt features. However, some tasks decrease in performance. For Gym Vault the performance with MAML is equivalent to random ordering of the videos. This may be because the training tasks have strong similarities with each other, but are not directly related to Gym Vault, as there are three diving and two snow sports tasks. This may also explain the success of MAML on the Ski Big Air and Snowboard Big Air tasks. With the other snow sport present in training, the model is able to learn features which are more applicable to the target task. This was not the case in the standard multi-task model, where the diving tasks likely outweigh Ski Big Air or Snowboard Big Air. For meta-learning to be successful for skill determination it seems that many more tasks are needed.

To a certain extent, MAML does learn more generalisable features. However, with only a handful of tasks it is evident that one or more tasks highly related to the target task are needed in training. Therefore, this approach would not work in daily-living tasks without annotating many more tasks for skill determination. An alternative approach could be to identify the similarities between potential source tasks and the target task to determine which will best transfer information to the target task.

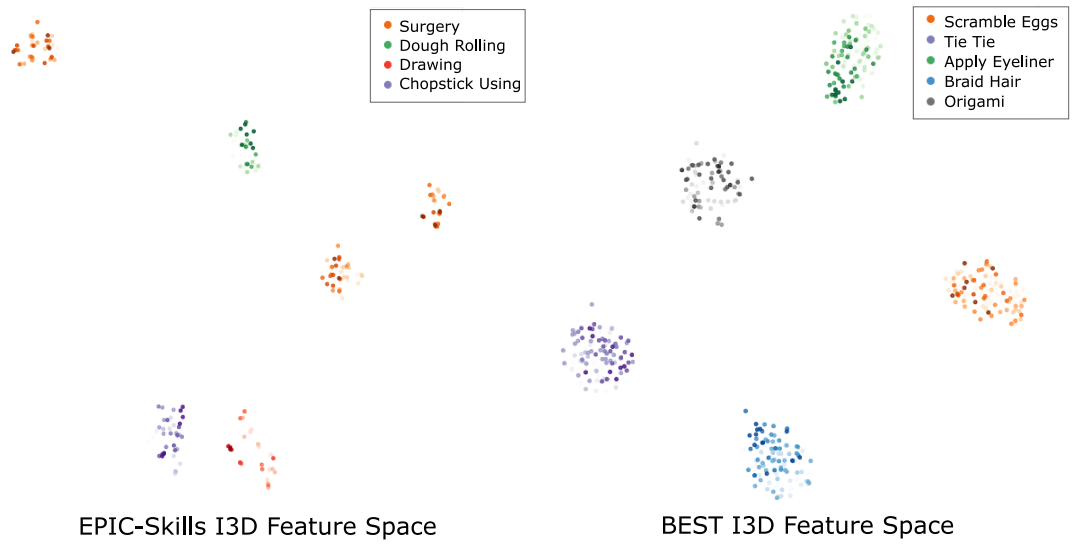

Figure 5.7: *Visualisation of the I3D feature space for all tasks in EPIC-Skills (left) and BEST (right). The colour intensity indicates the ground-truth rank, higher ranked videos are displayed with a higher colour intensity.*

5.3 Discussion

The previous sections reported results when learning skill tasks jointly. This proved to be challenging, with joint-learning of the tasks within EPIC-Skills and BEST decreasing the result in the majority of cases. Transferring knowledge from existing skill tasks to new ones also proved difficult. While knowledge can be transferred between highly related tasks in AQA-7 [133] and some less related tasks, such as Dough Rolling and Chopstick Using, it is hard to predict when a transfer is possible. This section discusses the reasons why it is challenging to share and transfer representations between tasks for skill determination.

Generalisation. With the current data, it is incredibly hard for a model to generalise across different tasks. Figure 5.7 shows t-SNE plots of the I3D features used to train the rank-aware attention network proposed in Chapter 4. The colour indicates the task and the intensity of the colour demonstrates the approximate ranking, with darker shades corresponding to higher ranked videos. Each task is in its own distinct part of the space and there is very little overlap between any of the tasks. Even the Surgery subtasks, which are recorded in the same environment, are in their own separate clusters. The exception is the two Drawing tasks which do overlap. Unlike the Surgery subtasks, all objects present in one Drawing subtask are present in the other. The feature space looks largely similar when jointly training all tasks, with each task still existing in its own separate space regardless of the model used.

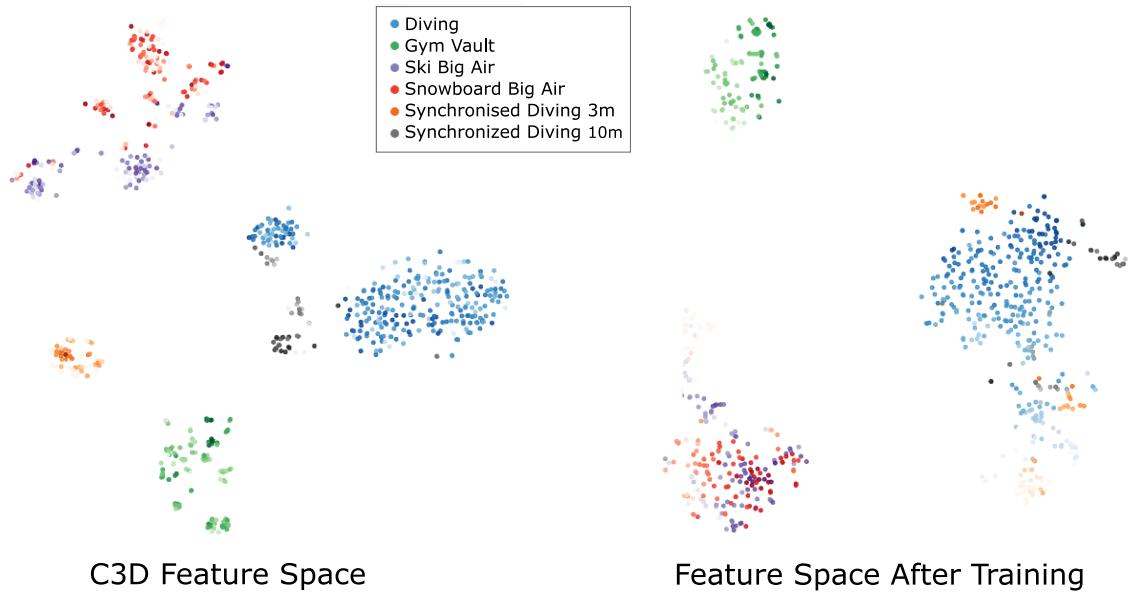

Figure 5.8: *Visualisation of the feature space for tasks in the AQA-7 dataset with C3D features before training and after training with C3D-LSTM [133, 134]. Ground-truth ranking is shown with the colour intensity, with a higher intensity meaning a higher ground-truth rank.*

There is much more overlap in the C3D feature space for the tasks in AQA-7. Figure 5.8 visualises these features when C3D is pre-trained on Sports1M. As would be expected, there is overlap between the two snow sports: Ski Big Air and Snowboard Big Air, as well as Diving and Synchronised 10m Diving. Figure 5.8 also shows the feature space after jointly training on all tasks. As these tasks already overlap with each other in the feature space it is much easier to share features when jointly training these six tasks. The overlap between Ski Big Air and Snowboard Big Air increases, as does the overlap between the three diving tasks. However, there is little overlap between these two clusters and the Gym Vault tasks despite this task containing similar motions as shown in Figure 5.9.

It is incredibly hard for the method to find commonalities between tasks, even when they share common movements and criteria for determining skill. One reason for this is likely the differences in domain between the tasks, as each task takes place in a particular environment. Thus, there is little sharing between tasks even when training on the full amount of data. With meta-learning methods such as MAML struggling to learn features which can generalise to these highly similar tasks, it would not be possible to learn generalisable features in BEST or EPIC-Skills where motions in the tasks themselves are incredibly different.

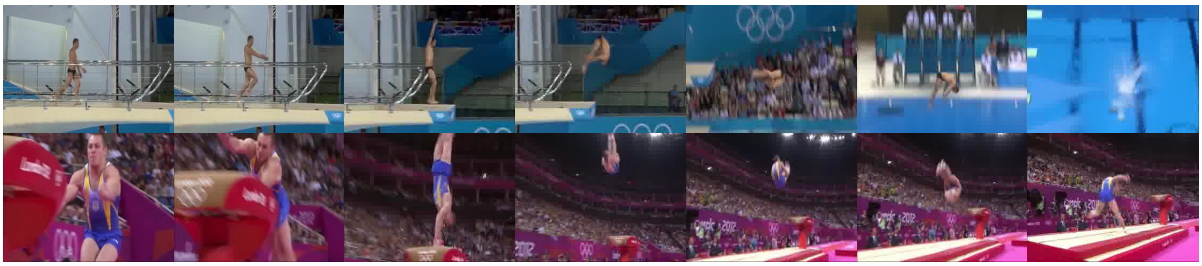

Figure 5.9: *A comparison of the Diving and Gym Vault tasks. Note that while the take off and landing methods are different, both videos involve performing multiple somersaults in a pike position.*

Complexity. Part of the reason that models are unable to generalise when jointly training on multiple skill tasks is the complexity of the tasks. High and low skill can manifest in very visually distinct ways. It is already challenging to cope with this diversity when learning to determine skill separately with an individual task.

The feature space of AQA-7 in Figure 5.8 demonstrates this complexity. After training there is some overlap between the three diving tasks, however this overlap is mainly between videos which demonstrate lower skill. This is likely due to splashing being a common feature across poorly scoring dives in each of the three tasks. There will be less commonalities between highly scoring dives as the types of dives possible are different in the 3m and 10m tasks. In addition, the synchronisation of the two divers is not something which needs to be accounted for in the Diving task.

The complexity becomes even more challenging with long videos of daily-living tasks, where Chapter 4 has shown it is important to locate the most relevant parts of a video to determining skill. The rank-aware attention network was able to successfully locate relevant video parts using only video-level labels indicating the overall skill shown in a video. However, this was learnt with many video examples. As shown by the experiments in Section 5.1, this attention is highly task specific and it would not be possible to learn attention for a target task without additional priors about the relevant actions.

Negative Transfer. There is also the question of which features within other tasks are relevant to determining skill in a new task. Section 5.2.1 showed many examples of negative transfer in BEST. However, there are also examples in AQA-7 where tasks share more similarities. It is also likely that negative transfers are present at a feature level, as well as a task level. For example, Chapter 3 demonstrated that quicker instances of Chopstick Using were generally considered more skillful, while the reverse was largely true for Drawing. Time is only one feature useful for determining skill in these tasks, many others will be more task-specific and not transferable.

While negative transfer has been observed in other transfer learning problems [17, 35, 49] it is not well studied. Instead, many works which aim to transfer knowledge between different problems or domains assume that the same classes are present in both source and target tasks and therefore aim to transfer knowledge by minimising the disparity between the feature spaces of the two tasks. Wang *et al.* [201] formalised the concept of negative transfer and presented a method which reweights the most relevant samples on the source tasks. This type of approach is unlikely to be helpful in skill determination as irrelevant and negatively correlated features will likely be present in all videos of a task. For instance, the amount of splash during the water entry in Diving is not useful for determining skill in Gym Vault. Regardless of the videos used, the concept of the water entry is irrelevant, although the higher-level concept of a good landing is present in both tasks.

Unlike classification tasks, negative transfers could be useful in skill determination. A negative correlation with the ground-truth demonstrates that there are some features useful for separating the video. However, a method is needed to identify these negative transfers so they can be utilised to improve the predicted ranking.

5.4 Conclusion

This chapter has explored whether models can be shared between different skill tasks and why transferring information to new tasks is particularly challenging in skill determination. The assumption that highly similar skill tasks could be trained jointly was verified and jointly training on less related tasks was also tested. While multi-task training was generally harmful to the results on BEST and EPIC-Skills, there were some surprising results when performing zero-shot transfers on these datasets. Seemingly unrelated tasks such as Dough Rolling and Chopstick Using obtain good results when a model trained on one task was applied to the other without any fine-tuning. This demonstrates that sharing of skill determination features should be possible, however it is difficult with current methods.

There are several steps which can be taken to overcome the challenges outlined in Section 5.3. First, better understanding of the relatedness of skill tasks is needed. With a larger number of annotated tasks it would be possible to build a taskonomy [212] of skill tasks and investigate transfers from subsets of source tasks. Knowledge of which groups of tasks share common skill features would be useful to explore future work in predicting relevant source tasks. Examining these task relationships will explain why the multi-task training works so poorly despite some of the zero-shot transfers being successful.

With this understanding, future directions in both multi-task learning and transfer learning could be explored. This chapter investigated multi-task learning with hard parameter sharing of all tasks within a dataset. It is likely many tasks will share some low-level features while certain subgroups of tasks will share higher-level features. Thus more adaptable feature sharing methods [100, 103, 114] could be applied to the skill determination problem to identify relationships and better cope with the limited amount of data available per task.

Future work in transfer learning could aim to predict which tasks and features within these tasks are the most useful to learn from. Recent work has explored attentive feature distillation and selection in the context of classification [191]. It would be interesting to adapt this to a ranking problem with a method which is able to make use of negative transfers. Another future direction is to investigate the transfer of temporal attention separately to the ranking. This would allow transfer learning in skill determination of daily-living tasks where only certain parts of the video are informative to skill.

To explore each of these directions further annotated skill tasks would be needed, as even with the short, highly related tasks in AQA it is difficult to transfer information between tasks. More data could also increase the ability of models such as MAML to learn general skill-relevant features which could easily be adapted to new tasks. Another possibility is to incorporate external knowledge about the skill-relevant parts of a video and what properties skillful performances of these subtasks have. Chapter 6 takes this direction by identifying skill-relevant actions as those described with adverbs in the narrations of instructional videos. The chapter then proposes a weakly-supervised method to learn a representation for these adverbs shared across tasks and actions.

Chapter 6

Learning from Adverbs in Instructional Videos

Chapters 3 and 4 both proposed methods which are capable of learning to determining skill in many different tasks. This is a step forward from prior work which focussed on skill in particular domains such as surgery and sports, thus proposed task-specific methods. However, the methods from Chapters 3 and 4 are still limited by the need to train skill tasks separately, with each new task currently requiring its own labelled data. Chapter 5 explored whether features could be shared or transferred between tasks, however this is incredibly challenging without additional information about what attributes skillful performances in different tasks share.

In Chapter 4, ranked videos from YouTube were used to learn to determine skill. Many of the higher ranked videos in the BEST dataset are instructional videos, made for the purpose of demonstrating how the task should be completed. These videos contain accompanying narrations indicating the way in which the instructor is completing the task. Often in these narrations, the instructor highlights key parts of the task which need to be completed in a certain way to achieved the desired outcome.

This chapter explores how instructional videos can be used as a source of external information about a task to identify and evaluate its skill-relevant parts. Specifically, it focuses on adverbs, which indicate how an action should be performed in order to complete the instructed task well. Therefore, a method which correctly associates relevant adverbs with an action can be used to identify whether this action has been performed in a skillful manner. Such a method is presented in this chapter along with an adverb retrieval dataset containing six adverbs (‘partially’, ‘completely’, ‘quickly’, ‘slowly’, ‘finely’

and 'coarsely') across 72 distinct actions. The representation for these adverbs is shared across tasks and actions, thus this is a step in the direction of sharing skill determination features across different tasks.

The problem of weakly-supervised adverb retrieval is introduced in Section 6.1. Secondly, an adverb dataset and accompanying semi-automatic method to collect such labels are presented in Section 6.2. Section 6.3 proposes a weakly-supervised embedding approach for adverb retrieval. This is then evaluated in Section 6.4.

6.1 From Narrated Adverbs to Action Modifiers

As discussed in Chapter 2 many works have utilised the narrations of instructional videos as a form of free supervision. For instance, several works aim to learn the key steps necessary to complete a task from instructional videos [2, 105, 161, 218]. While identifying the steps necessary to complete a task (or their order) can be relevant to determining skill, this is not all that is needed. As shown in the BEST dataset (Chapter 4), the participants displaying little skill often still complete the task. It is how or how-well the steps within the task are performed which differentiates videos in terms of skill.

Instructions, such as recipes, identify these key steps and indicate how these steps need to be performed to achieve the desired outcome. As well as including these kind of directions, instructional videos contain visual example of the instructed way to perform a step. This makes these videos an ideal form of data for learning characteristics of skillful performances in a task.

An example of an instructional video for the task of 'making a meringue' is shown in Figure 6.1. In some cases the key steps are indicated by the desired state of an object *e.g.* 'whisk the egg whites until stiff'. In other cases, adverbs illustrate how an action should be performed *e.g.* 'slowly add the sugar'. As discussed in Chapter 2 many works have tackled the problem of recognising attributes of objects, however few works have looked at adverbs in video. This chapter thus focusses on the latter type of instruction.

This chapter aims to utilise instructional videos and accompanying narrations to learn to retrieve adverbs which describe how the task is being performed. However, this form of supervision is weak and noisy as instructional videos are created to instruct people in a task, not to provide a supervision for computer vision algorithms. The narrations are only roughly aligned with the actions in the video. In some cases the narration will occur before the step is demonstrated so viewers know what they are about to see. In others, the description of the step will be given after the action has finished. This

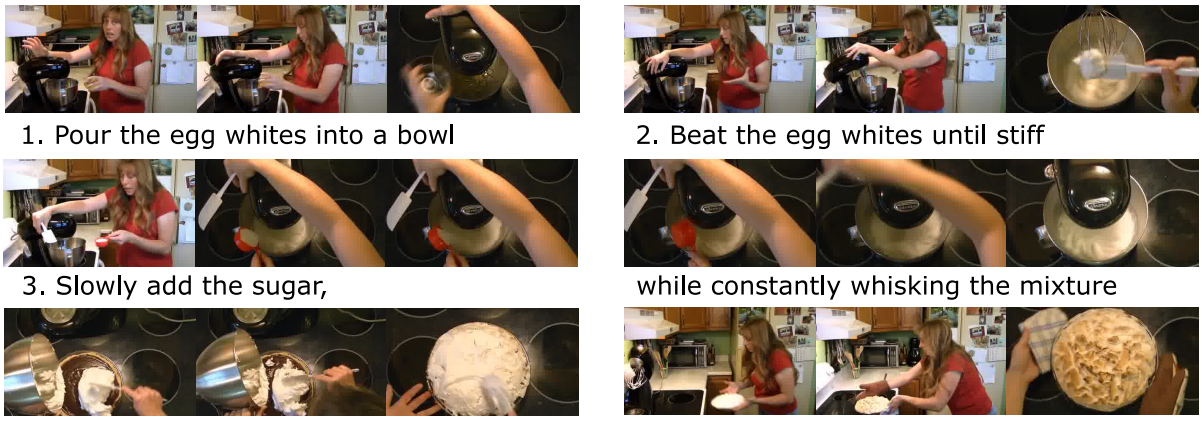

Figure 6.1: *The key steps for making a meringue as shown in an instructional video.*

Type	Gives information about	Example
Manner	How something happens	chop the onion **finely**
Place	Where something happens	just strain the mixture **here**
Time	When something happens	I made this bread **earlier**
Degree	How much something happens	we've **almost** added everything we need
Focusing	Something specific	**only** mix in the sugar
Evaluative	The speaker's opinion	**surprisingly** it will combine together
Viewpoint	The speaker's perspective	**personally** I don't remove the seeds
Linking	Relationships between clauses	**therefore** we need to loosen the screws

Table 6.1: *Types of adverb, taken from the Cambridge English Dictionary* [1]

visual resolution is an insignificant effort for humans, but still a challenging problem in automatic video understanding. Additionally, the narrated actions may not be captured in the video altogether. For example, an instructional video might be narrated as 'pour in the cream quickly' but the visuals skip to the cream already added as it is a trivial step for humans to perform. In this case the video would not be useful to learn the adverb *quickly*.

There are many types of adverbs and not all adverbs are relevant to skill. Table 6.1 shows the types with examples of each. This chapter focuses on 'manner' and 'degree' adverbs as these are the skill-relevant adverbs which describe how an action should be performed. Even within these categories, many adverbs are ambiguous or subjective and not pertinent to skill, *e.g.* 'beautifully' or 'cheerfully'. Therefore, this chapter works with a manually filtered subset of adverbs and focuses on the challenges of retrieving the

[1] https://dictionary.cambridge.org/grammar/british-grammar/adverb-phrases

relevant adverb(s) while coping with weak-supervision and learning representations for these adverbs which are shared across different tasks and actions.

This newly presented task of *weakly-supervised adverb retrieval* attempts to learn a representation for these adverbs from noisy and poorly aligned narrations. An illustration of the proposed approach is shown in Figure 6.2. This approach tackles two of the main challenges in learning a representation for adverbs from weak supervision: learning the visual representation only from the relevant parts of the video and disentangling the adverb from the action allowing adverbs to be applied to multiple actions. First action-adverb pairs from the narrations are identified. These are then used to embed portions of the video relevant to the action in a joint video-text embedding space. Since adverbs generalise to different actions and modify the manner of an action, they are learnt as action modifiers, *i.e.* transforms in the video-text embedding space, which are shared between actions. From this space, the relevant adverb can be retrieved by measuring the proximity of the modified action to the embedded video.

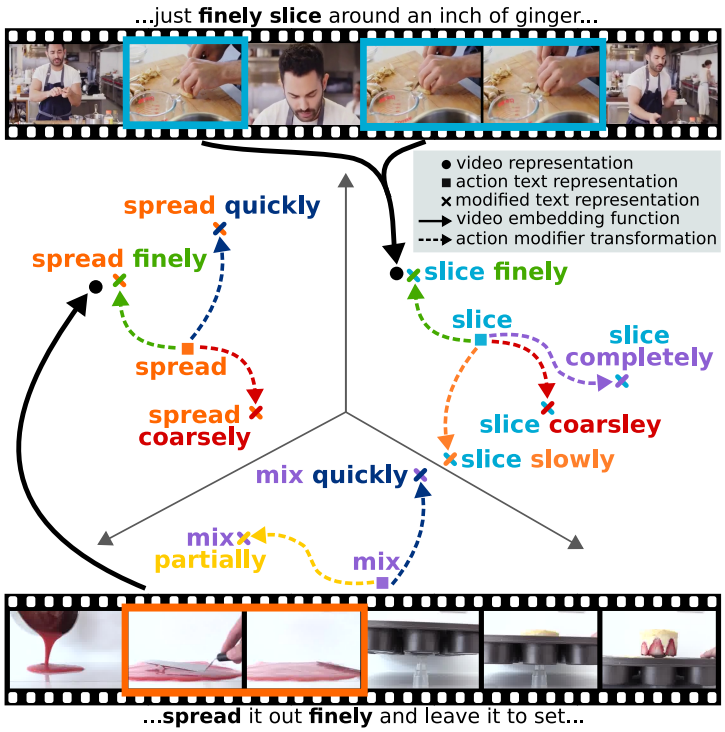

Figure 6.2: *A joint video-text embedding space can be learnt from instructional videos and accompanying action-adverb pairs in the narrations. Within this space, adverbs are learnt as action modifiers — that is transformations which modify the embedding of an action. Video portions relevant to the narrated action are attended to and these are embedded close to the action text embedding modified by the narrated adverb.*

6.2 An Adverb Retrieval Dataset

This section details the semi-automatic annotation procedure used to obtain annotations of action-adverb pairs from instructional videos. The source for the instructional videos is explained in Section 6.2.1. Section 6.2.2 then introduces the method use to obtain action-adverb annotations from these videos and Section 6.2.3 explores the contents of the dataset.

6.2.1 Instructional Videos

HowTo100M [110] is a large-scale dataset of instructional videos collected from YouTube, primarily used for pre-training video understanding models. Each video has a corresponding narration, either from manually entered subtitles or YouTube's Automatic Speech Recognition (ASR). Videos are sourced by querying YouTube for a variety of "how to" tasks (*e.g.* "how to prune a tree"), sourced from WikiHow [1]. The dataset consists of 1.22 million videos of tasks from a variety of domains ranging from cooking and DIY to health and education.

When initially querying the videos from HowTo100M for adverbs, many of the adverbs found came from completely irrelevant videos, such as documentaries or people reading books aloud. The videos in HowTo100M are obtained automatically and are not filtered manually, therefore many are not relevant to the queried "how-to" task. While adverbs used in instructional videos often indicate an important instruction relevant to the task, the use of adverbs are relatively infrequent within a video. Thus the adverbs found in the irrelevant videos outweigh the skill-relevant adverbs from instructional videos. This is less of an issue when performing video-text retrieval on HowTo100M with all parts-of-speech, as the focus is more on verbs and nouns which are frequent within all types of video, therefore the noise is proportionally much smaller.

To avoid a large proportion of the noise, this chapter considers the set of tasks previously explored in CrossTask [218] and uses all available videos from HowTo100M for these 83 tasks. The tasks predominately come from the domains of cooking, drink-making, DIY and car maintenance. These are divided into 65 tasks for training and a disjoint set of 18 'related' tasks for testing. Each training task consists of between 100 and 500 videos. In total this gives 24,558 videos in training and 1,280 in the test set.

Examples of these tasks can be seen in Table 6.2. Videos of the same task will often contain the same adverbs describing the same steps, as shown in Figure 6.3. This will allow a model to learn a generalised representation of the adverb which is not dependent on the viewpoint or appearence of the object being used. Adverbs are also shared across

Training Task	Av. adverb
Make Tiramisu Coffee	1.73
Make a Christmas Cake	1.72
Make a Black Forest Cake	1.70
Make Peppermint Meringue Cookies	1.41
Make Battenburg Cake	1.28
Make Vanilla Custard	1.11
...	
Make Chilli Con Carne	0.74
Make Blueberry Pancakes	0.67
Build a Desk	0.66
Change a Hubcap	0.65
Grill Kabobs	0.64
Make Mocha Nutella Gelato	0.63
...	
Make an Americano	0.21
Make a Frappe	0.20
Make Limeade	0.19
Make Bicerin	0.14

Test Task	Av. adverb
Make Meringue	1.63
Make French Strawberry Cake	1.61
Make Kerala Fish Curry	0.98
Change a Tire	0.85
Grill Steak	0.80
Make Bread and Butter Pickles	0.78
Make French Toast	0.72
Make Pancakes	0.65
Jack Up a Car	0.64
Build Simple Floating Shelves	0.62
Add Oil to Your Car	0.57
Make a Latte	0.51
Make Kimchi Fried Rice	0.51
Make Jello Shots	0.49
Make Banana Ice Cream	0.44
Make Taco Salad	0.43
Make Irish Coffee	0.31
Make Lemonade	0.30

Table 6.2: *Example "how-to" tasks in the CrossTask dataset for training (left) and testing (right). For each task the average adverbs found per video is displayed.*

Figure 6.3: *Examples of how action-adverb pairs vary within and between tasks. Adverbs often modify the same action in different videos of the same task, however the viewpoint and objects used can be visually diverse. The same action-adverb pair can also be found in different contexts in other tasks.*

different tasks, where they appear modifying the same action applied completely different objects as well as other actions. Separating the dataset into training and testing splits by task allows the generalisation of the proposed action modifier model to be tested, as the actions and adverbs will have been previously seen, but in different contexts.

The aim of this chapter is to learn a representation for adverbs from the weak-supervision of an adverb's presence in a video's narration. Therefore, the next step is collect all examples of adverbs from the above tasks, where each adverb is paired with the action it applies to and the accompanying video.

6.2.2 Collecting Adverb Annotations

The narrations of the instructional videos are used to discover action-adverb pairs for both training and testing. English subtitles containing transcriptions of these narrations are extracted from the YouTube videos. In many cases, accurate subtitles are provided by the video owner. Where these are not available, YouTube's ASR is used to generate subtitles automatically.

To locate adverbs, Part-of-Speech (PoS) tagging and dependency parsing are performed with SpaCy's English small core web model. PoS tagging predicts the type of each word within a sentence, while dependency parsing builds a graph of the relationship between words. The output of this step is displayed in Figure 6.4. Since YouTube's ASR subtitles are not punctuated, T-BRNN [180] is employed to punctuate these before PoS tagging to improve the result.

Adverbs (tagged with 'RB') are selected if they have an *advmod* dependency to a verb. For example, 'slowly stir' in Figure 6.4. Verbs are indicated by any of the tags listed in Table 6.3, however verbs with 'VBD' and 'VBZ' tags are removed as these correlate with actions not being shown in the video. After observing many examples, verbs tagged with 'VBN' were not removed, despite being a form of past tense. This form is often used as the instructor is finishing the step, therefore the action is often present in the video. It is also necessary to check whether an adverb is negated. For instance, 'don't chop the pepper finely' is entirely different to 'chop the pepper very finely'. This can be detected by inspecting whether any words are connected either to the adverb or corresponding verb with a 'neg' dependency.

However, not all adverbs and actions obtained from this process are visually relevant. As mentioned previously, 'manner' and 'degree' adverbs have the most potential to be skill-relevant. Even within these categories many adverbs are not useful for determining skill as some 'manner' adverbs describe the emotions with which an action is performed

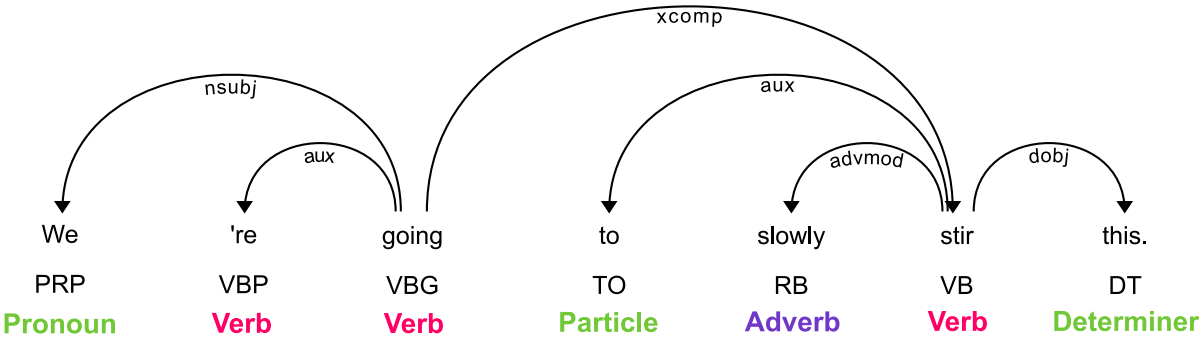

Figure 6.4: *An example output from the dependency parsing. Verbs and adverbs with the 'advmod' dependency are extracted as annotations.*

Tag	Description	Example
VB	Base form	**spread** all this evenly
VBD	Past Tense	I already **chopped** this coriander finely
VBG	Gerund or present participle	I'm **stirring** these together gently
VBN	Past participle	once all of your sugar has been **incorporated** slowly
VBP	Non-3rd person singular present	now I **wash** my ingredients thoroughly
VBZ	3rd person singular present	everything **fits** together neatly

Table 6.3: *Verb tags alongside examples found in instructional video narrations*

e.g. 'happily'. Therefore, adverbs, as well as actions, are manually filtered. Automatic approaches for this filtering were explored, namely using concreteness scores [13] of each word, however this was unreliable. Concreteness scores seemed to have little correlation with how visual an adverb or action was. For instance, *globally* had a higher concreteness score than *quickly*.

Actions and adverbs are also grouped into clusters to avoid synonyms, *i.e.* words with the same meaning. For example, 'put' and 'place' are considered to be the same action. Again, automatic approaches were attempted for this, specifically WordNet [113] hierarchies and Word2Vec [112] embeddings. Both of these approaches have successfully been used to group objects in computer vision, however they do not work well for grouping verbs [30] nor, as found here, adverbs.

WordNet is a hierarchical dataset of English words grouped into sets of cognitive synonyms (synsets) which each expresses a distinct concept. Words have multiple synsets for each use case of the word. Without knowing the correct synset it is not possible to group the actions or adverbs correctly. Furthermore, the adverbs and verbs do not have as deep and as fine-grained a hierarchy as nouns, for instance *easy* and *tardily* are considered synonyms of *slowly*.

Adverb	Count	Antonym	Count
completely	2045	partially	763
gently	1545	firmly	319
quickly	933	slowly	878
properly	1734	improperly	62
finely	804	coarsely	78
separately	737	together	67
evenly	748	unevenly	42
carefully	460	carelessly	8
easily	408	strenuously	30
continuously	319	periodically	20
tightly	271	loosely	59
gradually	219	instantly	1
forward	98	backward	16
generously	66	sparingly	0
vertically	40	horizontally	20
underneath	46	above	2
neatly	27	messily	7
upward	14	downward	5
inward	2	outward	2

Table 6.4: *Counts of adverbs and their antonyms found by the narration parsing process from the training set videos.*

Word2Vec is an unsupervised method which uses word co-occurrence to learn an embedding space of words. Within this space it is possible to calculate the similarity of two words uses cosine similarity of the vectors which represent the words. However, adverbs are often similar to their antonyms, *i.e.* words with completely opposite meanings. For example, *quickly* is more similar to *slowly* than to *fast*.

Clustering was thus performed manually and resulted in 38 adverb clusters and 72 action clusters. The adverbs are organised into adverb-antonym pairs where the adverbs have opposite meaning, *e.g.* *completely* is the opposite of *partially*. Instances of negated adverbs are placed in the cluster corresponding to the antonym of that adverb, *e.g.* 'don't mix it completely' is considered an instance of *mix partially*.

From this process, 15,266 instances of action-adverb pairs are obtained. The counts per adverb in the training set can be seen in Table 6.4. However, these have a long tail of adverbs which are mentioned only a handful of times. Furthermore, the problem of recognising adverbs across different actions is incredibly challenging due to the noisy, weak supervision from the narrations and the adverb's visual appearance being highly dependent on the action. Therefore, this chapter restricts the learning to 6 commonly

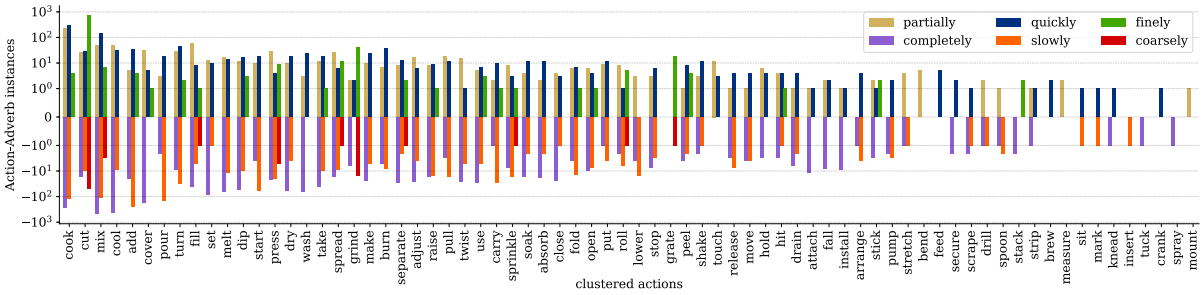

Figure 6.5: *Log-scaled y-axis shows instances of each adverb plotted per action. Adverbs are displayed against their paired antonym.*

used adverbs, that come in 3 pairs of antonyms: *partially*/*completely*, *quickly*/*slowly* and *finely*/*coarsely*. These are selected as they are relatively unambiguous (unlike *properly* versus *improperly*) and the adverb and antonym have a large amount of overlap in the actions they apply to (unlike *gently* and *firmly*[2]). The aim is to focus on the problem of using weak supervision to learn a representation of adverbs across many actions and leave the issue of few-shot adverb retrieval for future work.

The mean of the timestamps for the detected verb and adverb is used as the weak supervision for the action's location. As the data is noisy, the test set is cleaned. The main source of noise is whether the action and adverb occurs around the weak timestamp. Thus, videos are only included in the test set if the action is present with the corresponding adverb within $\pm$ 10 seconds around the weak timestamp.

6.2.3 Dataset Makeup

The final distribution of the action-adverb pairs are shown in Figure 6.5. In total, adverbs appear in 264 unique action-adverb pairs with 72 actions. There are 5,475 individual instances in training and 349 in the cleaned test set. This corresponds to 44% of the original test set. A higher level of noise than HowTo100M's report 50% is obtained [110], even after restricting the tasks examined to those in the CrossTask subset. The authors of HowTo100M measure a clip as relevant to its caption if *any* verb or noun in the caption is present in the video clip. In this chapter, an annotation consists of a single verb and a single adverb and a clip is only deemed relevant if *both* the action and the adverb are present.

Examples of the adverbs and actions obtained through the narration parsing process are shown in Figure 6.6 alongside the corresponding video. In some cases, the action is

[2]Firmly is used predominately in combination with *cut*, *press* and *fill*, whilst gently is mainly used with *add*, *fold*, *cook*, *mix* and *dip*

Figure 6.6: *Example videos and narrations, highlighting the action and adverb discovered through the narration parsing pipeline.*

well aligned with the weak timestamp, such as in the top row of Figure 6.6. However in many cases the weak timestamp is not a good localisation, as demonstrated by the remaining rows. The action could be long (second row) and thus a single timestamp is a poor representation. Alternatively, the timestamp could be a poor match as in third row, where the narrated action happens earlier and a different action is occurring at the weak timestamp. In other cases the action may not be captured in the video at all: in the fourth row only the end result of dicing of the spring onion is shown, while in the fifth, the instructor is discussing something irrelevant to completing the task. The next section explains how these challenges are overcome to learn a representation for adverbs from instructional videos.

6.3 Learning Action Modifiers

This section introduces the method which is able to learn representations for adverbs from weak supervision in instructional videos. An overview of the method can be seen in Figure 6.7. The inputs to the model are action-adverb narrations and the accompanying instructional videos, *e.g.* "...start by *quickly rolling* our lemons..." (Figure 6.7(a)), from which the action *roll* and the adverb *quickly* are identified (see Section 6.2). The aim after training is to be able to assess whether other *roll* actions and entirely different

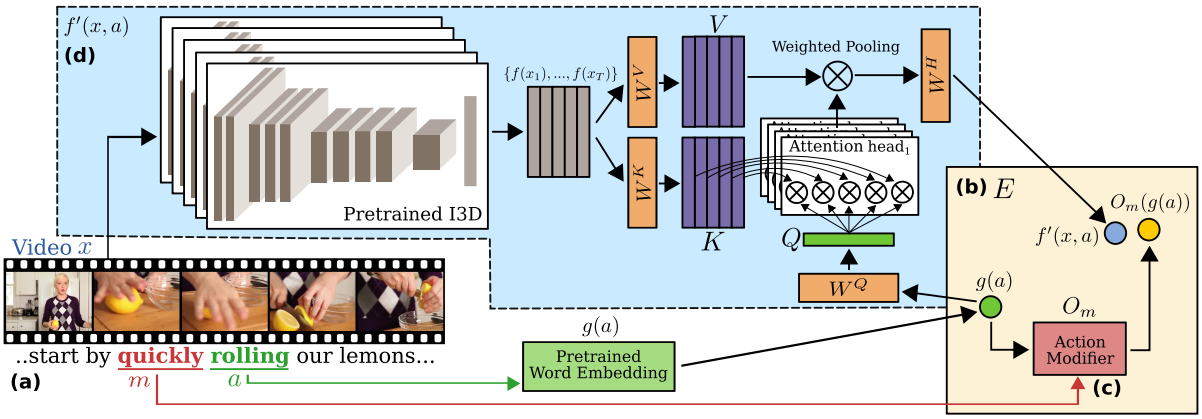

Figure 6.7: *(a) The input is a video x with the weak label (a, m) for the action and adverb respectively. (b) The aim is to learn a joint video-text embedding space for adverb and video retrieval where the embedded video (blue) and action-adverb text representation (yellow) are close. (c) Adverbs are learnt as action modifiers which are transformations in the embedding space. (d) The visual representation of the relevant video parts, $f'(x, a)$ is embedded using multi-head scaled dot-product attention where the query is a projection of the action embedding $g(a)$.*

actions from different videos have been performed *quickly* among other adverbs.

The joint video-text embedding space learnt to achieve this goal is shown in Figure 6.7(b). Section 6.3.1 reviews how joint video-text embeddings are typically learnt from this type of input and introduces notation for the rest of this chapter.

Two prime challenges exist in learning the embedding space for the problem of weakly-supervised adverb retrieval. The first is being able to disentangle the representation of the adverb from the action, allowing the method to learn how the same adverb applies across different actions. In Section 6.3.3 action modifiers are introduced, represented as transformations in the embedding space, which allows this disentanglement by learning one action modifier per adverb, as in Figure 6.7(c).

The second challenge is learning the visual representation from the relevant parts of the video in a weakly-supervised manner, *i.e.* without annotations of temporal bounds. This challenge is addressed in Section 6.3.4 with a weakly-supervised embedding function which utilises multi-head scaled dot-product attention. This uses the text embedding of the action as a query to attend to relevant video parts, as shown in Figure 6.7(d).

Finally, Section 6.3.5 describes how the model learnt from the previous sections can be used for video-to-adverb and adverb-to-video retrieval.

6.3.1 Learning an Action Embedding

Previous works aiming for action retrieval have learnt a joint video-text embedding [109, 202, 205]. In these video-text embedding spaces, video clips and text which correctly describes the videos, *e.g.* a caption or an action's verb, are close together and video-text pairs with low relevance are far apart. This chapter follows a similar approach for the base model. Specifically, given a set of video clips $x \in X$ with corresponding action labels $a \in A$, the goal is to obtain two embedding functions, one visual and one textual, $f : X \rightarrow E$ and $g : A \rightarrow E$ such that $f(x)$ and $g(a)$ are close in the embedding space E and $f(x)$ is distant from other action embeddings $g(a')$.

These functions f and g can be optimised with a standard cross-modal triplet loss:

$$\mathcal{L}_{triplet} = \max(0, d(f(x), g(a)) - d(f(x), g(a')) + \beta) \; s.t. \; a' \neq a \tag{6.1}$$

where a' is an action different to a, d is the Euclidean distance function and β, set to 1 in all experiments.

The GloVe [140] embedding of the verb describing the action is used for $g(a)$, this process is described in Section 6.3.2. Section 6.3.4 describes how $f(x)$ is replaced by $f'(x, a)$ to allow for weak supervision.

6.3.2 The Text Embedding Function

Global Vectors (GloVe) [140] is an unsupervised method for learning vector representations of words. The aim is to have similar words clustered together in the space. It uses global word-word co-occurrence statistics to guide this learning. First, probabilities of two words appearing in the same context are calculated, *e.g.* $P(ice|water)$. The aim is then to learn vectors w_i and w_j as points in the embedding space E to represent the ith and jth word in the corpus such that the following is true:

$$F(w_i, w_j, \tilde{w_k}) = \frac{P(k|i)}{P(k|j)} \tag{6.2}$$

where $\tilde{w_k}$ is a separate context word vector and F is the function which generates the vectors. Predicting the co-occurrence probabilities, rather than the probabilities themselves, allows the model to separate whether two words co-occur because they are related or because a word is used often. For instance, $\frac{P(solid|ice)}{P(solid|steam)}$ will be high as ice is a solid whereas steam is not. In GloVe, F is modelled with weighted least squares regression.

A word embedding method is used for the text embedding function $g(\cdot)$ in the proposed

model instead of a one-hot encoding as the word embedding should aid generalisation. Related actions should be close in the embedding space, therefore the video embedding function will be able to share information for similar actions. GloVe was chosen as it has successfully been used for many other visual-text embedding problems including the related problem of learning attributes of objects [123], although other word embedding methods such as Word2Vec [112] or FastText [53] would also be suitable.

6.3.3 Modelling Adverbs as Action Modifiers

While actions exist without adverbs, adverbs are by definition tied to the action, and only gain visual representation when attached to one. Although adverbs have a similar effect on different actions, the visual representation is highly dependent on the action. Therefore, following prior work from [123] on object-attribute pairs, adverbs are modelled as learned transformations in the video-text embedding space E (Section 6.3.1). As these transformations modify the embedding of the action they are named **action modifiers**. One action modifier O_m is learnt for each adverb $m \in M$, such that

$$O_m(z) = W_m z \tag{6.3}$$

where z is any point in the embedding space E and $O_m : E \to E$ is a learned linear transform with weight matrix W_m. In Section 6.4, other geometric transformations: fixed translation, learned translation and non-linear transformation are tested. Each transformation O_m can be applied to any text representation $O_m(g(a))$ or video representation $O_m(f(x))$ in E to add the effect of the adverb m. Since the function is invertible, O_m^{-1} could also be applied to any text or video representation in E to remove the effect of the adverb m.

A video $x \in X$, labelled with action-adverb pair (a, m), contains a visual representation of the adverb-modified action. Thus, the aim is to embed $f(x)$ close to $O_m(g(a))$, while ensuring different videos x' displaying different actions a' or adverbs m' are embedded far away. Note that this is equivalent to embedding the inverse of the transformation $O_m^{-1}(f(x))$ near the action $g(a)$. The embedding function $f(\cdot)$ is learnt jointly alongside action modifiers O_m $\forall m \in M$ using two triplet losses. One focuses on the action:

$$\mathcal{L}_{act} = \max(0, d(f(x), O_m(g(a))) - d(f(x), O_m(g(a'))) + \beta) \; s.t. \; a' \neq a \tag{6.4}$$

where a' is a different action and d and β are the distance function and margin as in

Section 6.3.1. The second loss function focuses on the adverb, such that:

$$\mathcal{L}_{adv} = \max(0, d(f(x), O_m(g(a))) - d(f(x), O_{\overline{m}}(g(a))) + \beta) \tag{6.5}$$

where $\overline{m}$ is the antonym of the labelled adverb m, *e.g.* when m = *quickly*, the antonym $\overline{m}$ = *slowly*. The negative in $\mathcal{L}_{adv}$ is restricted to only the antonym because adverbs are not mutually exclusive. Furthermore, the adverb annotations are not exhaustive as the instructor in the video only notes the key adverbs relevant to succeeding in the task. For example, a video labelled as *slice quickly* does not preclude the slicing from also being done *finely*. However, it surely has not been done *slowly*. The effect of this choice is demonstrated in Section 6.4.

The overall loss is then the sum of the action and adverb focussed losses:

$$\mathcal{L} = \mathcal{L}_{act} + \mathcal{L}_{adv} \tag{6.6}$$

6.3.4 Weakly-Supervised Embedding

All prior works which learn attributes of objects from images [23, 66, 115, 123, 124] utilise fully annotated datasets, where the object the attributes relate to is the principal object of interest in the image. In contrast, this chapter aims to learn action modifiers from video in a weakly-supervised manner. The input is untrimmed videos containing multiple consecutive actions. To learn adverbs, the visual representation needs to exclusively be from video parts relevant to the action which the adverb applies to (*e.g.* *roll* in the example in Figure 6.7). This chapter proposes using scaled dot-product attention [185], where the embedded action of interest acts a 'query' to identify relevant video parts.

The scaled dot-product attention uses keys, values and queries. The motivation for this comes from retrieval systems. If you wanted to search for some video on YouTube you would type in a **query**. The search engine would then map the query against a set of **keys** (*e.g.* video title, description) associated with the videos (**values**) in the database. For the weakly-supervised embedding in this chapter, the **keys** are a set of video segments from a video which can be **queried** with the action of interest. The result of this is the video segments (also the **values**) containing the relevant action.

For each video x, a temporal window of size T is used, containing video segments $\{x_1, x_2, ..., x_T\}$. This is centred around the timestamp of the narrated action-adverb pair. Starting from the visual representation of all segments $f(x) = \{f(x_1), f(x_2), ..., f(x_T)\}$, where $f(\cdot)$ is an I3D network, the aim is to learn to embed the visual features relevant

to the action a. This embedding is performed by the function $f'(x, a)$. First, $f(x)$ is projected into lower dimensional keys K and values V using linear projections:

$$K = W^K f(x); \qquad V = W^V f(x) \tag{6.7}$$

The query $Q = W^Q g(a)$ is set to be a projection of the action embedding, to weight video segments by their relevance to that action. The attention weights are obtained from the dot product of the keys K and the action query Q. These then pool the values V to obtain a single feature vector per video. Specifically:

$$H(x, a) = \text{softmax}\left(\frac{(W^Q g(a))^\top W^K f(x)}{\sqrt{T}}\right) W^V f(x) \tag{6.8}$$

where $H(x, a)$ is a single attention head. Mapping videos $f(x)$ into separate keys K and values V means information necessary to identify the relevant segments can be present in K without being in V, which is used to produce the final embedding used for adverb retrieval.

As explained in Chapter 4, a single attention head will tend to focus on a single aspect of the videos. Therefore, multiple attention heads are trained to give the attention multiple representation subspaces and allow it to jointly attend to the multiple pertinent parts of an action in a video. The final video embedding function is:

$$f'(x, a) = W^H[H_1(x, a), ..., H_h(x, a)] \tag{6.9}$$

where W^H projects the concatenation of the multiple attention heads into the embedding space. Each of the h attention heads $H_i(x, a)$ learns its own separate weights: W_i^Q, W_i^K, W_i^V. It is important to highlight that these weights are trained jointly with the embedding space E so that $f'(x, a)$ is used instead of $f(x)$ in Equations 6.4 and 6.5.

6.3.5 Inference of Adverbs

Once trained, the model can be used to evaluate cross-modal retrieval of videos and adverbs. For video-to-adverb retrieval, from a video query x and the associated narration action a, the goal is to estimate the most relevant adverb(s) m. For example, from a video with the narration "slice the cucumber..." the aim is to determine the manner in which the action *slice* was performed. The learned function $f'(x, a)$ is used to embed the parts of the video x relevant to the action a in E. Adverbs are then ranked by the distance from this embedding to all modified actions $\forall m : O_m(g(a))$.

For adverb-to-video retrieval, the query is an action-adverb pair (a, m), *e.g. slice finely*, embedded as $O_m(g(a))$. The distance from this text representation to all relevant embedded videos $\forall x : f'(x, a)$ is calculated to find the nearest videos. For both cases, this allows the action a to be used as the query in the weakly-supervised embedding so as to attend to the relevant video parts.

6.4 Experiments and Results

This section performs experiments on the method proposed in Section 6.3 on the data collected in Section 6.2. First, the implementation details of the method are described in Section 6.4.1, followed by the metrics used for evaluation in Section 6.4.2. Quantitative and qualitative results are then presented in Sections 6.4.3 and 6.4.4 respectively and the contribution of the different model components is evaluated in Section 6.4.5.

6.4.1 Implementation Details

Video Features. For consistency in frame rate, videos are first re-sampled to 25 frames per second. Both motion and appearance features are extracted from an I3D network [18] before the output layer. Networks for both modalities were first pre-trained on ImageNet[32] and then on Kinetics-400 [79]. The appearance network takes RGB images and the flow network uses grey-scale images from the TV-L^1 optical flow algorithm [211]. Both networks use 16 frame segments with frames re-scaled to a height of 256 pixels and centre cropped to obtain 224×224 pixel images. The feature extraction results in a 1024 dimensional vector for both modalities, these are then concatenated to create a 2048 dimensional vector. One feature vector $(f(x_i))$ is extracted per second as in [218], for $T = 20$ seconds around the weak timestamp for both training and test videos.

Text Features. Action embeddings are initialised with a 300 dimensional GloVe [140] vector corresponding to the action's verb. The GloVe model used was pre-trained on the Wikipedia and Gigaword corpora.

Architecture Details. In all experiments, the embedding space E is 300 dimensional, the same as GloVe word vectors. The weights W_m of the action modifiers O_m (Equation 6.3) are initialised with the identity matrix so the transformation begins with no effect. For the scaled dot-product attention (Equation 6.8), Q is of size 75×1 and K and V are of size $75 \times T$. Unless otherwise stated, 4 attention heads are used in $f'(x, a)$.

Training Details. All models are trained with the Adam optimiser [80], with a batch size of 512 and learning rate of 10^{-4}, for 1000 epochs. To aid with disentangling the actions and adverbs, the model first uses $\mathcal{L}_{triplet}$ (Equation 6.1) for 200 epochs to learn a good action embedding space before the action modifier transformations O_m are introduced. Once introduced the weights of the action modifiers, W_m (Equation 6.3) are learned at a slower rate of 10^{-5}.

6.4.2 Evaluation Metrics

Mean Average Precision (mAP) is reported for video-to-adverb and adverb-to-retrieval. For **video-to-adverb** retrieval, given a video and the narrated action, the relevance of the 6 adverbs are ranked. There are two settings for this metric: **Antonym** and **All**. In **All** the rank of all 6 adverbs are considered, this indicates which adverbs the model predicts as being relevant and present in the narrated action of the instructional video. In **Antonym** the retrieval is restricted to the only ground-truth adverb and its antonym. This metric better represents the available labels as the adverbs in the narrations are not exhaustive. To clarify, a video may be narrated with *cut coarsely*, meaning the *cut* was performed *coarsely* rather than *finely*. However, the narration gives no indication of whether the cut was performed *quickly* or *slowly*. Therefore, it is particularly important that the correct adverb be retrieved before its antonym. In video-to-adverb retrieval with the 'Antonym' setting there are only two possible adverbs to retrieve, therefore Precision@1 (P@1) is reported, which is the same as binary classification accuracy.

For **adverb-to-video**, given an adverb query (*slowly*) the method should rank the videos based on how much they display this adverb. To achieve this, videos are embedded by $f'(x, a)$ with their narrated action a (*e.g. put*). These videos are then ranked by the distance of their embedding to the embedding of the queried adverb with the ground-truth narrated action (*e.g. put slowly*). With the **All** setting all videos are ranked. In the **Antonym** setting only videos with the labelled with the query adverb or its ground-truth are ranked. In this case a perfect average precision score of 100 would indicate all videos labelled *slowly* are retrieved before all videos labelled *quickly*. This method of ranking allows for an overall rank per adverb, rather than a rank per action-adverb pair. While a rank per action-adverb pair would also be interesting, a rank per adverb allows the model to be compared to baselines which do not have knowledge of the action and prevents models from trivially succeeding in the case where certain actions are never narrated with the antonym.

Video-to-adverb with the 'Antonym' setting is the most relevant metric to skill determination as it could assess whether an action has been done as instructed. However, the

'All' setting is also useful; a ranked list of relevant adverbs could be used to indicate how an action should be done from visual demonstration when verbal or written instructions are not available. Adverb-to-video retrieval is not directly related to assessing skill, but it could be used to retrieve demonstrations for instruction or feedback.

6.4.3 Comparative Results

Since the method described in Section 6.3 is the first to learn from adverbs in instructional videos, there are no existing baselines. The most similar existing works are those that learn attributes of objects in images, therefore a selection of these works are adapted for comparison alongside several naive baselines. In this adaptation, actions replace objects and adverbs replace attributes. The following approaches are compared to:

- **Chance:** Returns a random order of items for a given query. This is included as a lower bound.
- **Classifier-SVM:** A Linear Support Vector Machine (SVM) which trains a binary one-vs-all classifier per adverb. This is analogous to the Visual Product baseline used in [115, 123]. In video-to-adverb, adverbs are ranked by classifiers' confidence scores. In adverb-to-video the confidence of the corresponding classifier is used to rank videos.
- **Classifier-MLP:** A 6-way Multi-Layer Perceptron (MLP) consisting of two fully connected layers. For video-to-adverb, adverbs are ranked by the confidence score of the corresponding output. For adverb-to-video videos are ranked by each video's score from the output corresponding to the query adverb.
- **RedWine [115]:** This method first learns a linear SVM classifier per action and adverb and then learns a transformation network to compose the component classifiers into a classifier for the combination of action and adverb. This is done by learning a projection from the component classifier weights to the feature space of the extracted visual features.
- **LabelEmbed:** This is a baseline from [115]. It is similar to RedWine, but composes word vector representations instead of SVM classifier weights. For the word vector representations, the pre-trained GloVe embeddings described in Section 6.3.2 are used.
- **AttributeOp [123]:** This method projects actions into an embedding space and learns attributes (adverbs) as linear operators on the action embeddings as described in Section 6.3.3. The authors also propose several regularisers for the

embedding space inspired by linguistic properties of adjectives. These linguistic properties can also apply to adverbs. Results for this method are reported with the auxiliary and commutative regularisers, as this is the best performing combination of regularisers. The auxiliary regulariser adds an classifier for actions and adverbs to ensure neither is lost in the composition, while the commutative regulariser ensures the order adverbs applied in can be swapped (*i.e. cut quickly and finely* is the same as *cut finely and quickly*).

Since RedWine, LabelEmbed and AttribtueOp were designed to retrieve attributes in images, they have no temporal aspect nor do any of these methods have any form of spatial attention. Therefore, the image representation is replaced by a uniformly weighted visual representation of the video segments. Similar to the method presented in this chapter, results for these methods are also reported with the action given in testing. This is referred to as the 'oracle' evaluation in [123]. Furthermore, for a fair comparison, only the antonym is used as the negative in each method's loss, as done in Equation 6.5.

Comparative results are presented in Table 6.5. The method presented in this chapter outperforms all baseline methods for video-to-adverb retrieval, both when comparing against all adverbs and when restricting the evaluation to antonym pairs. It can be seen that AttributeOp is the best baseline method, generally performing better or comparably to RedWine and LabelEmbed. The two latter methods work on a fixed visual feature space, so while these methods are less prone to over-fitting, they are prone to errors when the features are non-separable in that space. As will be shown in Section 6.4.4, it is not easy to distinguish between adverbs from the initial I3D feature space, therefore these methods struggle. Table 6.5 also shows that LabelEmbed performs better than RedWine across all metrics, demonstrating that GloVe features are better representations for actions and adverbs than SVM classifier weights.

The only metric for which the method presented in this chapter does not offer a significant improvement over baselines is adverb-to-video retrieval with the 'All' setting. AttributeOp is able to better retrieve videos containing the most likely action for an adverb. For instance, with the query *finely* AttributeOp retrieves the majority of videos which contain a *cut* action before any other action. The method presented in this chapter is able to isolate the most important video segments to better separate adverbs from their antonyms, *i.e.* retrieve more *cut finely* videos before *cut coarsely*. However, by not performing this separation Attributes as Operators is more successful at learning correlations between videos and likely adverbs.

Method	video-to-adverb		adverb-to-video	
	Antonym	All	Antonym	All
Chance	50.0	40.8	51.1	17.0
Classifier-SVM	60.5	53.2	56.3	26.4
Classifier-MLP	68.5	60.2	60.3	30.4
RedWine [115]	69.3	59.4	59.5	29.0
LabelEmbed [115]	71.7	<u>62.1</u>	<u>61.8</u>	29.7
AttributeOp [123]	<u>72.8</u>	61.2	59.7	**35.0**
This method	**80.8**	**71.9**	**65.7**	<u>32.9</u>

Table 6.5: *Comparative Evaluation. Best performance in **bold** and second best <u>underlined</u>. Results are reported for both video-to-adverb and adverb-to-video retrieval when results are restricted to the adverb and its antonym (Antonym) and when unrestricted (All).*

Learning one modifier per adverb targets generalisability, therefore it is useful to examine how the proposed model does for action-adverb pairs with few occurrences in training. The proposed model achieves 64 P@1, compared to 57 P@1 for AttributeOp, on action-adverb pairs with less than 10 training samples. While the proposed method does improve few-shot performance there is still a large gap between few and many-shot pairs, which obtain 84 P@1.

6.4.4 Qualitative Results

This section first looks at qualitative video-to-adverb retrieval results with a demonstration of the temporal attention. Then the learnt embedding space is examined.

Video-to-Adverb Retrieval. Figure 6.8 shows attention weights for several action queries alongside the predicted actions and adverb (closest in the embedding space) when using the ground-truth action as the query in the scaled dot-product attention (see Section 6.3.4). From Figure 6.8, it is evident that the scaled dot-product attention is able to successfully attend to key segments relevant to each action query. This is true even for relatively rare actions in the dataset, such as hold or move. It is also clear that the model attends to different sets of segments unique to each action query and that all queries ignore segments which do not contain any action. With the ground-truth narrated action as the query, the model is able to successfully recognise the action from the embedding of the video (as shown by the 'predicted actions') and is therefore able

Figure 6.8: *Temporal attention values from several action queries. The intensity of the colour indicates the attention value. Recall that the narrated action is used to weight the relevance of the video segments. Using this, the top-5 predicted actions, as well as the predicted adverb, are displayed for all cases.*

to predict the correct adverb.

The fifth example demonstrates the two main outcomes when an action is not present in the video. In many cases the distribution of attention values will be close to uniform, as with *bend*. Alternatively, the method will attend to the action segments which are most similar to the queried action. For instance, with *dip*, the method attends to segments in the latter part of the video as the scissors *dip* into the kimchi to cut it.

However, the results in Figure 6.8 also highlight some limitations of the method. In some cases the scaled dot-product attention is unable to locate the segments relevant to the narration. This occurs in both the fifth and sixth examples. In the fifth example, the method mainly attends to the cutting of the pork, rather than the kimchi. In the sixth

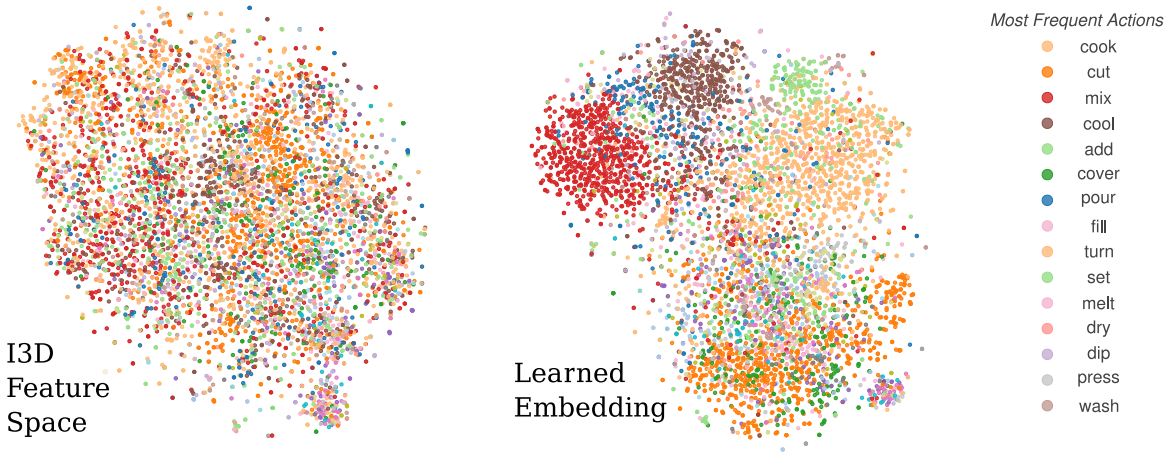

Figure 6.9: *Visualisation of the feature space for all actions before and after training. Colours indicate the different actions.*

video, the method detects the removal of lug nuts and taking of the tire as *take* actions, but not the removal of the jack in the penultimate segment. Even if the method picked up on these segments it could still be problematic as different adverbs may apply to each instance of the queried action. This highlights the need for the inclusion of further context in future work. In some cases, multiple actions occur simultaneously (*e.g.* *mix* and *add* in the second example). If different adverbs apply to these co-occurring actions it could make it difficult for the model to identify the correct adverb, even if it correctly attends to the relevant video segments.

Embedding Space. As Section 6.3 argues, it is necessary to learn a good action embedding in order to learn the action modifiers which represent how an adverb effects an action. Figure 6.9 shows t-SNE projections of the full embedding space before (*i.e.* from I3D features) and after training. Before training, the feature space does not separate either actions or adverbs well. After training, the overall feature space shows good separation of the actions. This is expected as the actions are dominant in the visual signal; *slice coarsely* and *slice finely* actions will look more visually similar than *slice coarsely* and *spread coarsely*.

To examine whether the embedding space has a good separation of adverbs, two sample actions are visualised separately. Figure 6.10 shows the embedding of all videos narrated with the action *cook* before and after training. Figure 6.10(a) highlights the adverb-antonym pair *completely-partially*, while Figure 6.10(b) highlights *quickly-slowly*. Points corresponding to other adverbs are faded out for ease of viewing. In each case, the embedding space shows the training successfully separates the adverb and its antonym. This is

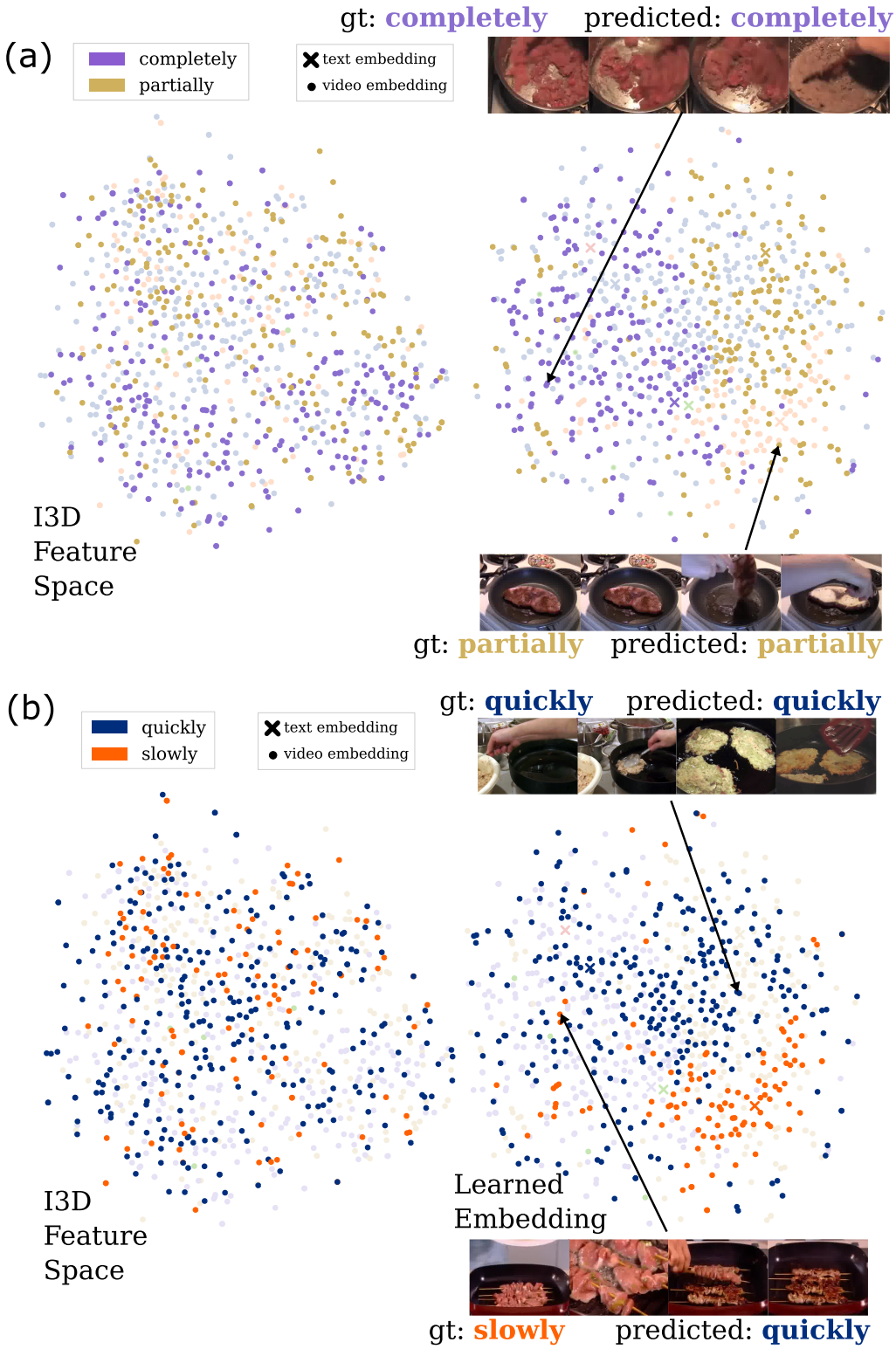

Figure 6.10: *Comparison between the feature space for the action 'cook' before and after training highlighting antonym pairs. In (a) the 'completely'/'partially' pair is highlighted with the other adverbs faded out. In (b) 'quickly' and 'slowly' are highlighted.*

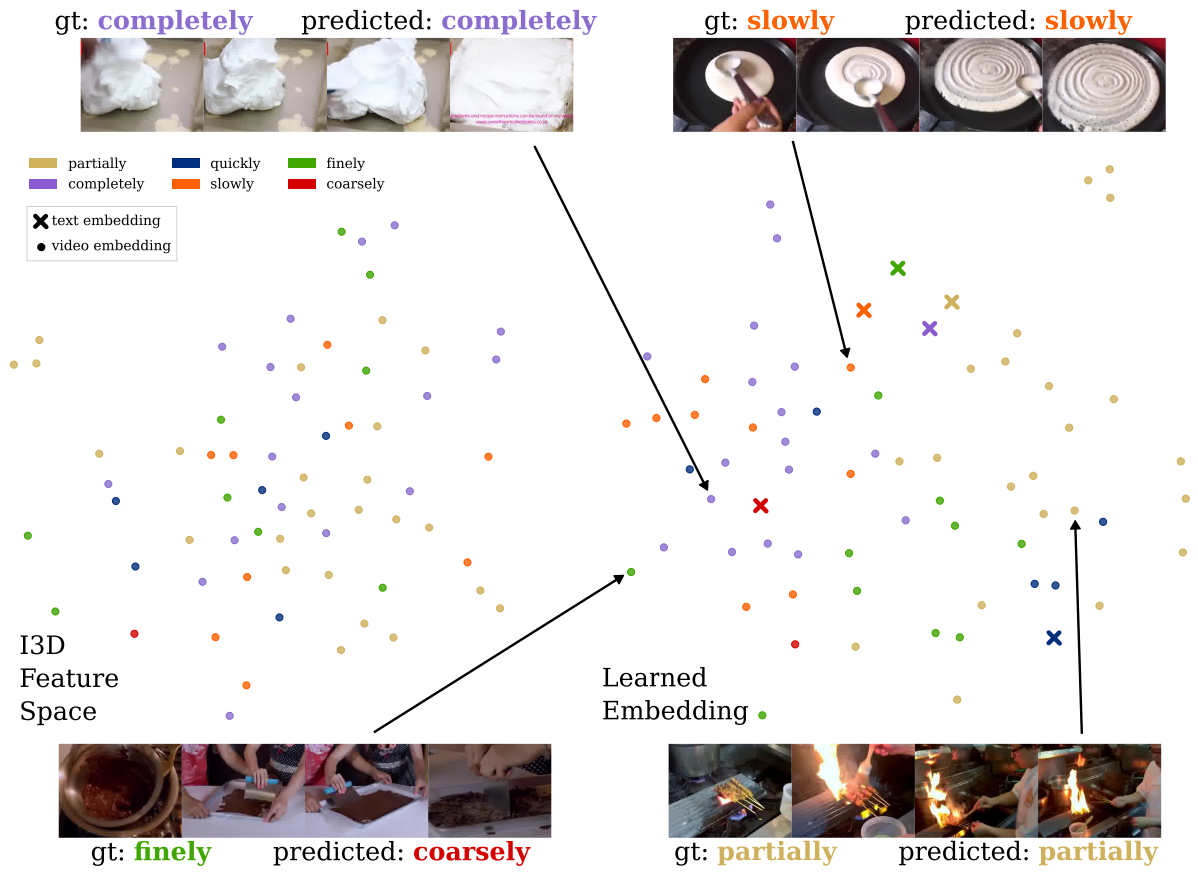

Figure 6.11: *Comparison of the features spaces before and after training for the action 'spread'. All adverbs are shown.*

also true for less frequent actions; Figure 6.11 demonstrates the embedding space for the action *spread*. Examples of embedded segments are shown alongside the embedding in both figures, with six videos embedded correctly within the corresponding ground-truth and two incorrect predictions: *cook slowly→quickly* and *spread finely→coarsely*.

With closer inspection of Figures 6.10 and 6.11, it can be seen that while adverb-antonym pairs are separated, there is still a significant amount of overlap between other adverbs. This is because the adverbs are not mutually exclusive and multiple adverbs can be applicable to a single action. For instance, in Figure 6.10 the top-left example demonstrates the *spread* being done *coarsely* as well as *completely*, hence it is embedded close the text embedding of *coarsely*.

6.4.5 Ablation Study

Having presented results on all metrics in comparison to baselines in Section 6.4.3, this section studies the impact of different model components. These ablation experiments focus on video-to-adverb retrieval using the Antonym P@1 metric, as this allows ques-

tions like "was the *cut* performed *quickly* or *slowly*" to be answered and is thus the metric most relevant for skill determination. The ablation studies focus on various aspects of the method including the temporal attention, the action modifier transformation $O_m(\cdot)$, the contribution of the loss functions, the length of the video (T), the contribution of the different modalities and the choice of query Q in the attention.

Temporal Attention. In Table 6.6, the proposed multi-head scaled dot-product attention approach (Section 6.3.4) is compared with alternative approaches to temporal aggregation and attention. Since the proposed scaled dot-product attention is reliant on a good action embedding, action retrieval results (mAP) are reported alongside video-to-action P@1. That is, given the embedding of the video $f'(x, a)$ queried by the ground-truth action, all actions in the embedding $\forall a : g(a)$ are ranked by their distances to the visual query and the rank of the correct action is evaluated.

It is important to note that the proposed method does not aim for action retrieval as it assumes knowledge of the ground-truth action and the embedding of that ground-truth action is used to calculate the importance of video segments. However, this metric does give an indication of the quality of the weakly-supervised embedding space. Results are compared to:

- **Single**: uses only a one-second clip at the timestamp.
- **Average**: uniformly weights the T features.
- **Class-agnostic Attention**: is a widely used form of attention [101, 125], similar to the attention filters used in Chapter 4. It calculates attention values per segment with two fully connected layers, *i.e.* $f'(x, a) = \text{softmax}(w_1 \tanh(W_2 f(x)))W_3 f(x)$.
- **Class-specific Attention**: a version of the above with one attention filter per action class.
- **This method w/o two-stage optimization**: the proposed attention without the first 200-epoch stage of learning only actions.
- **This method**: the proposed attention as described in Section 6.3.4.

Table 6.6 demonstrates superior performance of the proposed method for the learning of action embeddings and, as a consequence, better learning of action modifier transformations. This demonstrates that adverbs are highly contextual and require a good grounding and knowledge of the relevant action in order to determine how the action is performed. These results also demonstrate the challenge of weak-supervision, with video-to-action retrieval only performing at 24.6 mAP when considering the single second

Method	Action	Adverb
Single	24.6	70.5
Average	25.7	71.6
Class-agnostic attention	23.5	70.8
Class-specific Attention	40.1	72.8
This method w/o two-stage optimization	58.6	77.4
This method	**69.2**	**80.8**

Table 6.6: *Comparison of temporal attention methods. Video-to-action retrieval mAP and video-to-adverb retrieval P@1 are reported.*

surrounding the narrated action. This improves to 69.2 with the proposed method.

There is a small improvement when including more video segments (Average) than the one centred around the weak-timestamp (+1.1 P@1). However, it is clear how important it is to attend to the relevant segments within the temporal window of size T as these results are still below the final performance.

The class agnostic attention performs poorly, with results comparable to 'Single' and 'Average'. The appearance of each action is significantly different. While a class agnostic attention may be able to distinguish between no action occurring and an action taking place, it is not capable of attending to the desired action when faced with multiple confounding actions in the video clip. A significant improvement in the action embedding can be seen with the 'Class-specific attention' (+16.6 mAP) which also reflects in the adverb results (+2.0 P@1), however this over-fits for rare actions.

The initial step of only optimising the action embedding for the first 200 epochs demonstrates how important learning a good action representation is to learning adverbs. This sole change improves the final result by 3.4 P@1. The error of the adverb is inherently tied to the error of the action. Without being able to correctly identify the relevant action it is difficult to determine the adverbs which pertain to this action, as the visual appearance of an adverb depends so heavily on the action it applies to. By learning a better action embedding initially, the proposed attention is able to better identify video segments relevant to the queried action and thus relevant to the adverb.

Action Modifier Representation. Table 6.7 examines different representations for the action modifiers $O_m(\cdot)$ (Equation 6.3). A fixed translation from the GloVe representation of the adverb (m), which is not learned, is compared to three learned representations. First, a learnt translation vector b_m initialised from the GloVe embedding

$O_m(z)$	Dimension	Learned	P@1
$z + \text{GloVe}(m)$	300D		73.5
$z + b_m$	300D	✓	74.9
$W_m z$	300×300D	✓	**80.8**
$W_{m_2}\text{ReLU}(W_{m_1} z + b_{m_1})$	$2 \times 300 \times 300$D	✓	74.2

Table 6.7: *Comparison of action modifier representation $O_m(\cdot)$. The linear transformation choice ($W_m z$) clearly improves results.*

is used. Second, is the representation from Equation 6.3 used in the results above: a 2D linear transformation with matrix W_m. Third, a non-linear transformation is learnt, implemented as two fully connected layers, the first with a ReLU activation.

Results show the linear transformation clearly outperforms a vector translation and the non-linear transformation. While learning an adverb translation does improve results over the static GloVe representation (+1.4 P@1), the translation vector does not have enough capacity to represent the complexity of the adverb. On the other hand, the non-linear transform is prone to over-fitting, particularly with action-adverb pairs which have a small number of examples.

To understand the success of the linear transformation it is also important to test the impact of the initialisation. The weights of the transformation (W_m) are initialised with the identity matrix, so the action modifiers initially have no effect in the embedding space. With a more standard initialisation of Kaiming Uniform [57], as used by fully connected layers in Pytorch [137], the performance drops to 80.2 P@1. This is a relatively small difference, indicating the initialisation method is not crucial to success, but initialising the adverbs such that they begin with no effect it is marginally helpful.

These results test the type of transform the action modifier performs on points in the video-text embedding space. However, this assumes having a transform per adverb which is shared between actions is beneficial. Table 6.8 tests this hypothesis by investigating how learning an independent embedding per action-adverb pair performs. This is still performed with the same scaled dot-product attention and initially learning the action embedding for 200 epochs to keep the conditions as close as possible. The adverbs are not learned with translation matrices for this experiment as it would be unreasonable to learn a 300×300 transformation per action-adverb pair and would most likely over-fit. Thus, learning a translation vector b_m per adverb m is tested against learning an a translation vector $b_{a,m}$ per action-adverb pair (a, m).

Initialisation	$O_m(z)$	P@1
Zero	z + $b_{a,m}$	57.0
GloVE	z + $b_{a,m}$	71.3
GloVE	$z + b_m$	74.9

Table 6.8: *Comparison of a learnt action modifier representation when shared across actions (b_m) and when learnt separately for which action-adverb pair ($b_{a,m}$)*

Table 6.8 shows that by learning a separate adverb transformation per action-adverb pair causes a significant drop in performance (3.6 P@1), even when initialising all instances of an adverb translation from a common GloVE embedding. In many cases the adverbs stay close to the GloVE embedding the translation vector is initialised with, meaning the adverbs are still mostly shared between different actions. Without the GloVe initialisation the performance drops further to 57.0 P@1.

Loss Functions. In this section the need for the two separate loss functions $\mathcal{L}_{act}$ (Equation 6.4) and $\mathcal{L}_{adv}$ (Equation 6.5) is evaluated, as well as the impact of the negatives used. As an alternative approach, a single loss function is used where the negative contains a different action and/or the antonym adverb. Specifically:

$$\mathcal{L}_{comb} = \max(0, d(f(x), O_m(g(a))) - d(f(x), O_{m'}(g(a'))) + \beta) \quad (6.10)$$
$$s.t.(a' \neq a \text{ and } m' = m) \text{ or } (a' = a \text{ and } m' = \overline{m}) \text{ or } (a' \neq a \text{ and } m' = \overline{m})$$

where $\overline{m}$ is still the antonym of the narrated adverb m.

As shown by Table 6.9, this performs worse by 3.7 P@1. The worse performance is due to the action dominating in the sampling of negative triplets. The same improvement could probably be achieved by weighting the sampling of negative action-adverb pairs, however two separate loss functions is a more intuitive solution. An alternative would be to have only the case where both the action and adverb are negative, however this means the model could effectively ignore the adverb when making the modified action text embedding closer than the modified text embedding of another action.

Using any different adverb ($m' \in M \setminus m$) as opposed to only the antonym $\overline{m}$ in $\mathcal{L}_{adv}$ (Equation 6.5) also results in worse performance. As many of the adverbs are not mutually exclusive, this version of the loss causes the model to over-fit to the narrated adverbs thereby rejecting any other correct but unlabelled adverbs. With this version of the loss there is no longer overlap between different adverbs in the embedding space as shown in Figures 6.10 and 6.11.

Loss	m'	P@1
$\mathcal{L}_{comb}$	$m' = \overline{m}$	77.1
$\mathcal{L}_{act} + \mathcal{L}_{adv}$	$m' \in M \setminus m$	78.2
$\mathcal{L}_{act} + \mathcal{L}_{adv}$	$m' = \overline{m}$	80.8

Table 6.9: *Ablation of the loss function. This demonstrates need for two separate loss functions, one focussing on the action the other on the adverb, over a single combined loss function. It also shows using the antonym $\overline{m}$ is the most effective choice for the negative adverb m'.*

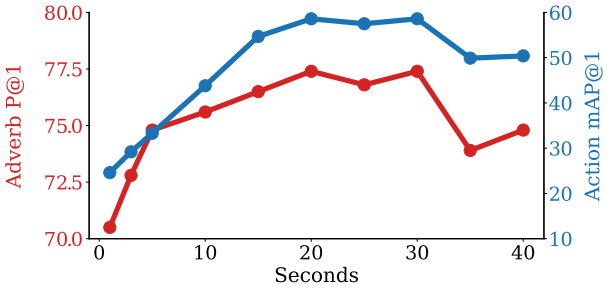

Figure 6.12: *Performance as T increases. Blue (axis and plot) shows video-to-action retrieval mAP while red shows video-to-adverb retrieval with Antonym P@1.*

Effect of T. The experiments thus far have been tested with a video snippets extracted from 10 seconds before the narration timestamp to 10 seconds after, *i.e.* $T = 20$. When evaluating the proposed method, particularly the scaled dot-product attention, it is important to verify that this length of video is necessary. The temporal attention ablation showed that this was a better alternative than using only a single second around the narration timestamp; here intermediate and succeeding values are investigated.

Figure 6.12 shows the video-to-adverb 'Antonym' performance alongside video-to-action mAP for $T = \{1, 3, 5, 15, 20, 25, 30, 35, 40\}$. As shown in Section 6.2, many actions are long and there are many cases where the action is not well aligned with the narration. Therefore, despite a higher proportion of confounding actions with larger T, videos are also more likely to contain the full length of the relevant action. The embedding function $f'(x, a)$ is able to ignore other actions in the video, to a point, and successfully learn to attend to the relevant parts given the query action, resulting in better performance with $T = \{20, 25, 30\}$. The performance does degrade for $T > 30$. After this point is it likely the additional segments only contain confounding actions, as indicated by the decrease in video-to-action mAP performance.

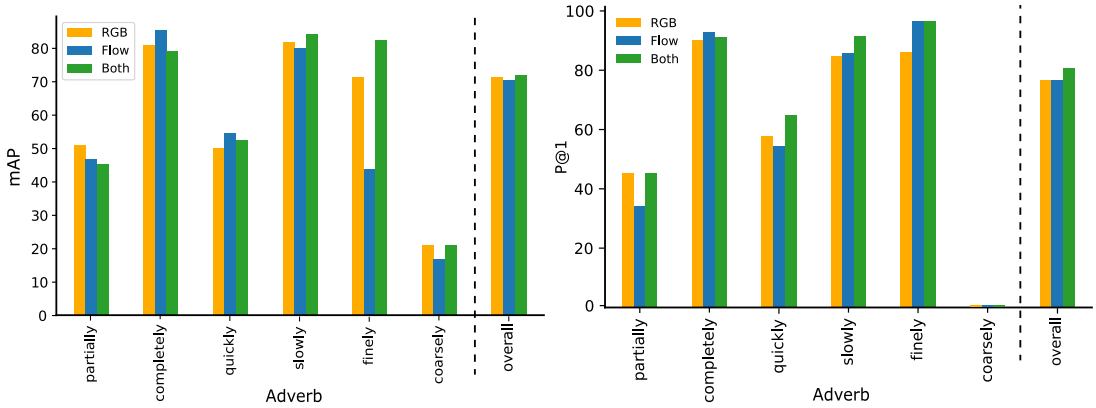

Figure 6.13: *Video-to-adverb retrieval mAP per adverb with different modalities. Left are results with the 'All' setting, right display results with the 'Antonym' setting.*

Effect of Different Modalities. Figure 6.13 shows the effect of different modalities on the results per adverb for 'All' (left) and 'Antonym' (right). Firstly, it can be observed that overall the inclusion of both RGB and Flow is better than either modality separately (+0.7 mAP and +1.6 mAP respectively with 'All'). However, modalities perform differently across individual adverbs. *Finely* is retrieved significantly more successfully with RGB than with Flow in the 'All' setting, although the addition of Flow in 'Both' does improve the performance of this result significantly (+11.2 mAP). Unsurprisingly, *quickly* and *slowly* also benefit from the inclusion of Flow features alongside RGB (+2.6 and +2.4 mAP respectively).

The results do highlight a drawback of the proposed method in that it is more likely to predict the adverbs with better support in training.

Choice of Q. As explained in Section 6.3.4, Q is the query in the scaled dot-product attention, which is used to attend to the relevant parts of the video for weakly-supervised embedding. The attention is calculated by the compatibility of this query Q with the keys K (a linear projection of the video segment features), therefore the choice of Q is integral to the method. This query is set to a linear projection of the embedding of the action, $W^Q g(a)$ (Equation 6.8) under the intuition that the adverb is highly dependent on the action, therefore it is necessary to attend to action-relevant segments in order to be able to retrieve the correct adverb(s). Table 6.10 tests several alternatives to this choice, including incorporating the adverb into the query. Note that for this ablation the two-stage optimisation of only learning actions in the first 200 epochs is not used, as this is specific to using only the action and not the adverb in the query Q. Thus, the performance of $g(a)$ matches that of 77.4 in Table 6.6.

	$Q =$	P@1
Action	$W^Q g(a)$	**77.4**
	$W^Q \text{OneHot}(a)$	73.6
Adverb	$W^Q \text{GloVe}(m)$	70.2
	$W^Q \text{Vec}(W_m)$	73.1
Both	$W^Q O_m(g(a))$	72.8

Table 6.10: *Comparison of the choice of query, Q, in the proposed scaled dot-product attention. The first column denotes whether the choice of Q relates to the action a, the adverb m or both.*

First, an alternative representation of the action is compared: a one-hot vector representing the action (OneHot(a)). From Table 6.10 it is evident the action embedding $g(a)$ performs better as a query.

Second, querying with the types of adverb instead of the action is tested. A single adverb from each antonym pair is used (*e.g.* 'slowly'/'quickly'→'quickly'). This offers the attention an understanding of the type of adverb to be retrieved, so as to pick video segments relevant to this action manner. Two representations for the adverb m are tested: the GloVe representation GloVe(m) and a flattened representation of the learned action modifier weights Vec(W_m). Again, while this can allow the method to focus on segments relevant to the type of action manner, using either action representation performs better than solely querying with the adverb for both representations.

Although it can be concluded that the adverb on its own in not useful, it is possible for it to be useful in combination with the action. Therefore, the final test is the full action-adverb embedding $O_m(g(a))$, again with a single adverb from each antonym pair (*e.g. slowly/quickly→quickly*). This shows a drop in performance compared to using only the action's embedding alone. This is likely related to the adverbs not being mutually exclusive.

Number of Attention Heads. Thus far the proposed scaled dot-product attention has used $h = 4$ attention heads (Equation 6.9). Multiple heads are used to allow the attention to jointly attend to the different pertinent parts of a video. This section investigates how crucial multiple attention heads are and how many heads are needed.

Figure 6.14 shows video-to-adverb P@1 alongside video-to-action mAP for different numbers of attention heads h. There is a clear improvement from incorporating multiple

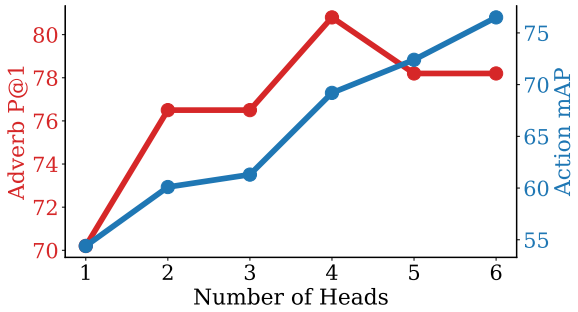

Figure 6.14: *Comparing the number of attention heads. Blue shows video-to-action retrieval mAP, while red shows Antonym P@1 video-to-adverb retrieval.*

heads in the scaled dot-product attention: 4 heads improves the result by 10.8 P@1 over a single attention head. Despite the video-to-action mAP increasing with more than 4 attention heads, the video-to-adverb result plateaus. It is likely with 4 heads the method can attend to enough action relevant segments to determine the adverb and further segments which benefit action retrieval do not aid retrieval of the correct adverb.

Comparison with Action Localisation. This chapter has explained how a weakly-supervised embedding containing action modifiers can be learnt by attending to action relevant segments. So far it has been established that the proposed scaled dot-product attention is better than alternative temporal attention strategies (Table 6.6) and that knowledge of the action is necessary to attend to relevant video segments in the weakly-supervised embedding (Tables 6.6 and 6.10). Since the success of the attention hinges on identifying key segments relevant to the action, this section tests whether off-the-shelf weakly-supervised action localisation methods can be used instead of the proposed attention, to locate key segments before learning action modifiers.

One disadvantage of the proposed method is that no component deals with the noise in the data. As mentioned in Section 6.2 in around 44% of cases the narrated action-adverb pair is not visually present in the T second window around the weak timestamp. Our method uses all data in training, including these noisy clips. With action localisation methods there is potential for these noisy clips to be ignored if there are no action proposals for the narrated action in the clip.

Published code is used for two state-of-the-art weakly-supervised action localisation methods:

- **W-TALC** [138]: uses Multiple Instance Learning (MIL) in combination with a Co-Activity Similarity loss to temporally localise actions given video level labels. MIL models video clips as bags, where a positive bag (*i.e.* video clip labelled with

the action) has one or more positive instances (*i.e.* segments containing the labelled action) and negative bags have no positive instances. The Co-Activity Similarity loss identifies correlations between videos containing the same action.

- **CMCS** [96]: aims to model the completeness of actions and to separate action instances from their surrounding context. The network contains multiple branches which are forced to focus on distinct video segments to model completeness. To separate action instances from surrounding context, negative training samples which contain little motion are used.

For both of these methods the same I3D features and the same $T = 20$ second temporal window used. Each method is fine-tuned on training set. Action proposals are extracted from each video in the training and test set if and only if the action proposal is of the same class as the narrated action. For the training set, only proposals with a confidence score greater than 0.5 are used. For the test set, the highest scoring action proposal of the narrated action is taken as the test set only contains videos where the narrated action has been confirmed to be present.

First, the output of these methods with an adverb classifier (Classifier-MLP as in Section 6.4.3) is tested, thereby completely replacing the proposed method with off-the-shelf components. For this experiment, the features contained within extracted action proposals are average pooled before being used to train the MLP adverb classifier. Results are presented in Table 6.11 (Attention='Avg' and Adverb Rep='Classifier MLP'). From these results it is clear that the weakly-supervised localisation methods in combination with the adverb classifier perform poorly.

These weakly-supervised action localisation methods are also tested in combination with the proposed action modifier transformations. As before, the features in the extracted action proposals are averaged (Avg). This operation replaces $f'(x, a)$ in the proposed method. While the action modifier representation does improve the results for the action proposals from W-TALC (+3.4 P@1), it still underperforms in comparison to the proposed method (-6.9 P@1). However, this result does indicate W-TALC is able to successfully filter out some irrelevant segments it is better than simply averaging all features (73.9 P@1 versus 71.6 from Table 6.6).

Although CMCS is a newer method, it underperforms in comparison to W-TALC. This is due to the modelling of 'background' video segments in this method: video segments with little motion are assumed to not have any action taking place and are used as negative examples. Since the actions in instructional videos are dense and many actions

Method	Attention	Adverb Rep	P@1
W-TALC [138]	Avg	Classifier-MLP	70.5
	Avg	Action Modifiers	73.9
	SDP	Action Modifiers	76.8
CMCS [96]	Avg	Classifier-MLP	69.6
	Avg	Action Modifiers	69.9
	SDP	Action Modifiers	70.5
This method	SDP	Action Modifiers	**80.8**

Table 6.11: *Comparison to weakly-supervised action localisation methods, with and without the proposed scaled dot-product (SDP) attention and action modifier representations.*

can be subtle[3], many of the negative examples detected in CMCS are actually examples of actions with relevant adverbs.

Finally, relevant segments detected by the weakly-supervised action localisation methods are combined with the scaled dot-product attention (SDP). This tests whether the two approaches are complementary. From Table 6.11 it can be concluded that using the output of a weakly-supervised action localisation method is insufficient. The proposed scaled dot-product attention is able to improve the result for W-TALC, however joint optimisation on the full video clips still performs best.

6.5 Conclusion

This chapter has presented work on understanding how different actions are performed by learning from adverbs in instructional videos in a weakly-supervised manner. This expands the scope of previous chapters which have focussed on how-well a task is performed overall. Instead, this chapter uses narrations from instructional videos as a guide to indicate which steps within a task require a particular performance and learns a model which is capable of determining whether actions are performed in this manner. It is possible to imagine future work which extends this method to provide feedback to people following either instructional videos or written instructions.

The proposed method learns to obtain and embed the relevant parts of the video with

[3]particularly in combination with an adverb such as *slowly*

scaled dot-product attention, using the narrated action as a query. Adverbs are then represented as action modifiers: linear transformations in the embedding space, which are shared between actions. This method is evaluated for the tasks of video-to-adverb and adverb-to-video retrieval and successfully outperforms existing works used for recognition of object attributes. The ablation study demonstrates the importance of the action query and that learning a good action embedding from the weak supervision is key to being able to disentangle the adverb's representation from the action to which it applies.

Future Work

The method used to attend to relevant action segments could be improved. As seen in Section 6.4, when there are multiple instances of the same action within the temporal window, the proposed method is unable to distinguish between them. Therefore, further context is needed. The most obvious way to incorporate this is by using the noun of the relevant object as a query in the attention in addition to the action's verb. In many cases the object is not explicitly mentioned, instead referred to as 'this' or 'it', however co-reference resolution or visual grounding [63] could by explored as a method to determine what 'this' and 'it' refer to. While the noun is the most obvious addition, further context from the rest of the narration could also be explored. Another potential avenue is using the action query for spatial attention in addition to temporal attention, as often there are multiple actions occurring at the same time.

Additionally, the work is limited by the number of adverbs it can represent. The method successfully models how the same adverbs can be applied to multiple different actions by examining six adverbs: *completely*/*partially*, *quickly*/*slowly* and *finely*/*coarsely*. While there is nothing in the method which inherently limits the number of adverbs which can be represented by action modifiers, the per adverb results (Figure 6.13) demonstrated that the method struggles with the imbalanced of *coarsely* versus *finely*. Therefore, either a method which tackles this imbalance is needed, or a way to eliminate the noisy video clips where the mentioned action is not present. This would allow adverbs to be mined from further instructional videos so the training distribution could be balanced. It would also be interesting to examine few-shot or zero-shot scenarios where the instances of an adverb, or the number of actions to which the adverb applies, is limited in training.

Chapter 7

Conclusion

This thesis has explored 'general' skill determination from video, where the aim is to be able to automatically assess skill in a variety of tasks. This was examined predominantly in the context of daily-living tasks, for instance preparing scrambled eggs or making origami, where videos are typically longer and more complex than previous tasks studied, such as diving. Therefore, a large focus of this thesis has been on extracting relevant information from long videos and dealing with sparse or weak annotations.

Previous approaches for skill determination from video focused on specific domains such as surgery or sports, where existing scoring metrics and prior knowledge about the task can be utilised. The thesis began by exploring skill determination without task-specific prior knowledge and found a general method could be used to learn to rank skill from videos in a variety of different tasks. However, skill is not apparent from all parts of a video, so it was then found that focusing on specific video parts indicative or higher or low skill is important to be able to determine skill accurately.

This thesis then explored sharing information about skill between tasks. It established that there are common features which can be used to determine skill, even in seemingly unrelated tasks. However, current methods struggle to learn generalisable features and to identify features shared between tasks using video-level supervision. Instead, the last chapter explored weak-supervision from the narrations of instructional videos to learn a representation for adverbs. These representations indicate how an action is performed can be shared across different actions and tasks.

7.1 Findings and Limitations

This section briefly summarises the findings and limitations of each chapter within this thesis.

Chapter 3 studied skill determination in a fully-supervised setting using a ranking approach. It proposed a pairwise ranking method, which can be used to assess skill in a variety of different tasks using pairs of videos with an obvious difference in skill as well as pairs which are similarly ranked. This was the first work to assess skill in daily tasks and was tested on 4 distinct types of tasks, with videos ranging from 30 seconds to 3 minutes in length. However, the method assumed all video parts to be equally important to determining skill and was therefore unable to cope with variability in the skill shown throughout a video. Another drawback is the need to train on tasks separately. While the method is generally applicable to any task and does not use prior, task-specific knowledge is does learn features unique to a particular task.

Chapter 4 explored the issue of some video parts being more important to determining skill. The proposed approach trains temporal attention modules, learnt with only video-level supervision using a novel rank-aware loss function. In addition, to attending to task-relevant video parts, the proposed loss jointly trained two attention modules to separately attend to video parts which are indicative of higher and lower skill. This chapter also described the collection of the skill determination dataset BEST, which was larger and contained longer tasks than existing datasets. The results showed the benefit of the rank-aware approach, particularly for the longer and more complex tasks in BEST, such as scrambling eggs. While this method does better focus on specific, skill-relevant parts of a video, it doesn't take temporal dependency into account, thus it cannot explicitly account for the order of actions being relevant to determining skill. Additionally, this method, like the one presented in Chapter 3, learns task-specific features for individual tasks, thus trained models cannot be easily deployed to new skill tasks.

Chapter 5 looked at multi-task and transfer learning in the context of skill determination. It found that multi-task learning was helpful for highly related tasks, but was otherwise challenging to learn and does not provide an experimental benefit. Performing zero-shot transfers between tasks highlighted some surprising relationships, for instance a model trained on people using chopsticks has reasonable performance when ranking videos of people rolling pizza dough and vice-versa. This chapter demonstrated that there is potential for skill-relevant features to be shared between tasks, however this remains an open and challenging problem, even for seemingly related tasks.

Finally, Chapter 6 shifted the focus to the performance of individual actions within a

task and looked at *how* specific actions were performed, rather than *how-well* the overall task was performed. It identified adverbs in instructional videos as a form of weak supervision for this and used these narrations to learn a representation for six adverbs which transform an action's embedding in a latent space. The representations of these adverbs generalise well across different actions and tasks, demonstrating this to be a promising future direction for determining skill in accordance with instructions given for a task. As well as the obvious limitation of the number of adverbs, this approach also fails to take into account other key sources of information in instructional videos. For instance, the state of particular objects can be key to a successful performance of a task.

7.2 Directions for Future Works

This thesis explored skill determination from videos of daily-living tasks. The work in this thesis is an initial exploration in this area, as such there are many interesting directions for future work in skill determination. Three of these avenues are outlined below.

Transferring Knowledge Between Tasks

The most limiting aspect of the skill determination methods presented in Chapters 3 and 4 is the need to train a separate model per task and therefore the need for a sizeable amount of annotated data per task. Chapter 6 demonstrated that representations shared between tasks can be learnt, however this is at the level of individual actions and requires an external source of knowledge which links the manner of an action to skill.

Chapter 5 examined why the transfer of information between tasks was difficult, although it did indicate that similar tasks, and some less related tasks, share features. This demonstrates that transferring and sharing information between skill tasks should be possible, although this exploration only considered the nine tasks within EPIC-Skills and BEST. A further exploration would require more tasks to better explore the relationships between them. It seems that jointly training all tasks would not be a helpful approach to learning generalisable and adaptable features for skill determination. Instead, a successful method may predict which previously learnt skill tasks, or features within those tasks, would be useful to determining skill in a new task.

Greater Understanding of Individual Tasks

This thesis moved away from prior works in skill determination which focused on predict-

ing skill in specific tasks, to propose 'general' methods applicable to determining skill in a wide variety of daily-living tasks. The proposed methods do obtain good performance across the variety of tasks examined in this thesis, however a method hand-crafted for a particular task is likely to have much better performance in the specific task.

The work in Chapters 4 and 6 has moved towards identifying key parts of a video relevant to skill in individual tasks, however models which learn a deeper understanding of global context in individual tasks would likely be more successful. For instance, when a subject performs an action in relation to other actions may be a key factor to determining skill in some tasks. One way to better learn to model individual tasks without requiring much more labelled data is to continue on the path of Chapter 6 and use narrations of instructional videos, or text-based instructions. Further information could be gleaned from these instructions, such as the properties of objects at particular stages of the task, which steps are optional or which steps are temporally dependent on one another.

Feedback and Explainability

Skill determination can be used to identify whether a person needs guidance in a particular task. However, it is not yet capable of providing that guidance. To achieve this, skill determination needs to be applied to other tasks such as retrieving the best videos to demonstrate (part of) a task or identifying which aspects the subject needs to improve in order to become more skilled. Chapter 4 did take a step towards this, by identifying video parts indicative of high or low-skill. Chapter 6 is also linked to this direction, as it used language to describe how individual actions were performed. However, this is far from explaining many aspects of skill in a task. Generating text feedback to explain what a person could do better or retrieving the most informative video clips to demonstrate the difference with a skillful performance of the task would both be promising avenues for research in this area.

7.3 Beyond Skill Determination

The main challenges in skill determination from long videos have been extracting relevant information from videos and doing so with limited supervision. These challenges are not unique to skill determination are also present in other video understanding tasks. As explained in Chapter 2, a large amount of work in video understanding has focussed on understanding short, temporally-trimmed video snippets. It is necessary to be able to understand these types of videos, however only focussing on short videos would be limiting. A deeper understanding of videos is be needed to move from identifying *what* is

happening in a video to answering other questions such as *how* and *why* it is happening. In such problems, picking out key information from long videos will be paramount.

Chapters 3 and 4 both proposed methods for fine-grained ranking in video, with Chapter 4 focussing on identifying which parts of the video were key to the final ranking. Particularly as these methods target 'general' skill determination, they are adaptable and can learn relevant features for a given task. This also means that these methods could be applied to different video ranking problems outside of skill. For instance, instead of ranking how-well as person completes a task, the videos could be ranked by their usefulness to instruct another person in the task. Online videos could also be ranked by their expected interestingness to a user or their virality. Another possibility is extending the relative attributes problem, studied in images [131, 173], into the video domain. These problems would all require ranking of videos where only certain video parts may be useful, thus the rank-aware attention method from Chapter 4 would allow these tasks to be explored.

The work from Chapter 6 could also be applied to other video understanding problems. The proposed method uses weak supervision from the narrations of instructional videos to learn a representation for adverbs which is shared across actions and tasks. A key component of the method is learning a weakly-supervised embedding, where the embedded action acts as a query to identify relevant video parts. This attention could also be applied for other problems which require close inspection of a particular action, such as identifying the object used in the action or the state changes of that object. With additional constraints the method could also be applied to localisation of actions or steps in instructional videos. Beyond actions, a similar approach could be taken for identifying attributes of objects in video, which instead learns a weakly-supervised object embedding.

The adverbs representations learnt in Chapter 6 are not only applicable to determining skill and could be useful in other areas of video understanding. For instance, the ability to represent adverbs effectively could lead to better video captioning methods, which are able to describe fine-grained differences between videos. Adverbs could also be used in fine-grained action recognition or retrieval. For example, the action 'scrub' is defined as 'to rub something hard in order to clean it'[1], thus the adverb 'hard' is a key factor in distinguishing between the actions 'scrub' and 'rub'. These are only a couple of examples of tasks where a deeper understanding of language through adverbs could be useful for video understanding.

[1] from the Cambridge English Dictionary

References

[1] Wikihow. https://www.wikihow.come. 19, 133

[2] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4575–4583, 2016. 18, 20, 130

[3] J.-B. Alayrac, I. Laptev, J. Sivic, and S. Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2127–2136, 2017. 21

[4] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5803–5812, 2017. 33

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016. 29

[6] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *Internation Conference on Learning Representations (ICLR)*, 2015. 13

[7] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision (ECCV)*, pages 584–599, 2014. 29

[8] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006. 6

[9] G. Bertasius, H. Soo Park, S. X. Yu, and J. Shi. Am i a baller? basketball performance assessment from first-person videos. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 36, 37

[10] V. Bettadapura, G. Schindler, T. Plötz, and I. Essa. Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2619–2626, 2013. 1, 40, 41, 44

[11] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 223–232, 2013. 45

[12] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, pages 25–36, 2004. 8

[13] M. Brysbaert, A. B. Warriner, and V. Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46 (3):904–911, 2014. 136

[14] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine learning (ICML)*, pages 89–96, 2005. 24

[15] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. 15

[16] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 129–136, 2007. 25

[17] Z. Cao, M. Long, J. Wang, and M. I. Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2724–2732, 2018. 127

[18] 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 (CVPR)*, pages 6299–6308, 2017. 10, 12, 44, 93, 145

[19] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 10

[20] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 10

[21] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. 110

[22] C.-Y. Chang, D.-A. Huang, D. Xu, E. Adeli, L. Fei-Fei, and J. C. Niebles. Procedure planning in instructional videos. *European Conference on Computer Vision (ECCV)*, 2019. 21

[23] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–207, 2014. 45, 143

[24] M.-H. Chen, Z. Kira, G. AlRegib, J. Yoo, R. Chen, and J. Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 75

[25] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 641–647, 2002. 23

[26] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 1, pages 1–2, 2004. 6

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005. 6

[28] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision (ECCV)*, pages 428–441, 2006. 6

[29] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *British Machine Vision Conference (BMVC)*, volume 2, page 3, 2014. 49

[30] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 13, 79, 82, 136

[31] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database. *Robotics Institute*, page 135, 2008. 49

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 7, 31, 66, 71, 93, 116, 145

[33] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997. 34

[34] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3725–3734, 2017. 12

[35] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1338–1345. IEEE, 2012. 127

[36] O. Duchenne, I. Laptev, J. Sivic, F. R. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 1491–1498, 2009. 18

[37] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy"–automatic naming of characters in tv video. In *British Machine Vision Conference (BMVC)*, volume 2, page 6, 2006. 18

[38] M. J. Fard, S. Ameri, R. B. Chinnam, A. K. Pandya, M. D. Klein, and R. D. Ellis. Machine learning approach for skill evaluation in robotic-assisted surgery. *arXiv preprint arXiv:1611.05136*, 2016. 36, 40

[39] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785. IEEE, 2009. 45

[40] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision (ECCV)*, pages 15–29, 2010. 30

[41] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision (ECCV)*, pages 314–327, 2012. 49

[42] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016. 63

[43] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 11

[44] C. Finn and S. Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *International Conference on Learning Representations (ICLR)*, 2018. 121

[45] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 120, 121

[46] G. Forestier, F. Petitjean, P. Senin, F. Despinoy, and P. Jannin. Discovering Discriminative and Interpretable Patterns for Surgical Motion Analysis. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 136–145, 2017. 36, 40

[47] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2121–2129, 2013. 31

[48] Y. Gao, S. Vedula, C. Reiley, N. Ahmidi, B. Varadarajan, H. Lin, L. Tao, L. Zappella, B. Béjar, D. Yuh, et al. The JHU-ISI gesture and skill assessment dataset (JIGSAWS): A surgical activity working set for human motion modeling. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2014. 48, 51, 55, 113

[49] L. Ge, J. Gao, H. Ngo, K. Li, and A. Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):254–271, 2014. 127

[50] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional Sequence to Sequence Learning. *International Conference on Machine Learning (ICML)*, 2017. 12

[51] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2014. 31

[52] A. S. Gordon. Automated Video Assessment of Human Performance. In *Proceedings of AI-ED*, pages 16–19, 1995. 36, 37, 40

[53] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 142

[54] G. S. Guthart and J. K. Salisbury. The intuitive/sup tm/telesurgery system: overview and application. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 618–621. IEEE, 2000. 42, 55

[55] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 34

[56] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006. 29

[57] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 156

[58] F. Heidarivincheh, M. Mirmehdi, and D. Damen. Beyond action recognition: Action completion in rgb-d data. In *British Machine Vision Conference (BMVC)*, volume 1, page 4, 2016. 44

[59] F. Heidarivincheh, M. Mirmehdi, and D. Damen. Action completion: A temporal model for moment detection. *British Machine Vision Conference (BMVC)*, 2018. 44, 49

[60] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. *International Conference on Artificial Neural Networks (ICANN)*, 1999. 24

[61] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153, 2016. 18, 20

[62] D.-A. Huang, J. J. Lim, L. Fei-Fei, and J. Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2183–2192, 2017. 21

[63] D.-A. Huang, S. Buch, L. Dery, A. Garg, L. Fei-Fei, and J. Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 21, 22, 164

[64] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report*, (07-49), 2007. 24

[65] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015. 10, 66

[66] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2015. 45, 143

[67] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 27

[68] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision (ECCV)*, pages 304–317, 2008. 29

[69] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311. IEEE, 2010. 29

[70] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):221–231, 2012. 7, 10, 11, 44

[71] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2000–2009, 2019. 11

[72] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of ACM International Conference on Multimedia Retrieval (ICMR), oral session*, 2011. 14

[73] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 15

[74] A. Jin, S. Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and L. Fei-Fei. Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 691–699, 2018. 40, 41

[75] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002. 24, 71

[76] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006. 71

[77] M. Jug, J. Perš, B. Dežman, and S. Kovačič. Trajectory based assessment of coordinated human activity. In *International Conference on Computer Vision Systems*, pages 534–543, 2003. 37

[78] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014. 7, 11, 71

[79] 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. 10, 93, 145

[80] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 93, 146

[81] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 44

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012. 7, 66, 71

[83] O. Kuchaiev and B. Ginsburg. Factorization tricks for lstm networks. *International Conference on Learning Representations Workshops*, 2017. 12

[84] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2556–2563. IEEE, 2011. 10

[85] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–787, 2014. 49

[86] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35 (12):2891–2903, 2013. 30

[87] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958. IEEE, 2009. 45

[88] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 6, 18

[89] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[90] P. Li, Q. Wu, and C. J. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 897–904, 2008. 23

[91] S. Li, G. Kulkarni, T. Berg, A. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, 2011. 30

[92] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. VideoLSTM Convolves, Attends and Flows for Action Recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 12, 84

[93] H. Lin and G. Hager. User-independent models of manipulation using video contextual cues. In *International Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)-Workshop*, 2009. 36, 40, 43

[94] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 11

[95] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A Structured Self-Attentive Sentence Embedding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 91

[96] D. Liu, T. Jiang, and Y. Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1298–1307, 2019. 5, 17, 28, 84, 162, 163

[97] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1656, 2017. 12

[98] T.-Y. Liu. *Learning to rank for information retrieval*. Springer Science & Business Media, 2011. 25

[99] X. Liu, J. van de Weijer, and A. D. Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1040–1049, 2017. 26, 35

[100] M. Long and J. Wang. Learning multiple tasks with deep relationship networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1594–1603, 2017. 115, 128

[101] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7834–7843, 2018. 13, 14, 91, 154

[102] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. 6

[103] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5334–5343, 2017. 115, 128

[104] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018. 12

[105] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015. 18, 20, 130

[106] A. Malpani, S. S. Vedula, C. C. G. Chen, and G. D. Hager. Pairwise comparison-based objective score for automated skill assessment of segments in a surgical task. In *International Conference on Information Processing in Computer-Assisted Interventions*, pages 138–147, 2014. 36, 40, 41, 44, 47

[107] A. Malpani, S. S. Vedula, C. C. G. Chen, and G. D. Hager. A study of crowdsourced segment-level surgical skill assessment using pairwise rankings. *International Journal of Computer Assisted Radiology and Surgery*, 10(9):1435–1447, 2015. 36, 40, 41, 44, 52

[108] J. Martin, G. Regehr, R. Reznick, H. Macrae, J. Murnaghan, C. Hutchison, and M. Brown. Objective structured assessment of technical skill (osats) for surgical residents. *British Journal of Surgery*, 84(2):273–278, 1997. 41, 43, 56, 113

[109] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 141

[110] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 18, 20, 34, 133, 138

[111] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9879–9889, 2020. 34

[112] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119, 2013. 136, 142

[113] G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 136

[114] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016. 115, 128

[115] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1792–1801, 2017. 45, 143, 147, 149

[116] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the ACM on International Conference on Multimedia Retrieval*, pages 19–27, 2018. 33

[117] N. C. Mithun, S. Paul, and A. K. Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11592–11601, 2019. 34

[118] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2204–2212, 2014. 13

[119] D. Moltisanti, S. Fidler, and D. Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9915–9924, 2019. 13

[120] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5715–5725, 2017. 75

[121] T. Munkhdalai and H. Yu. Meta networks. *Proceedings of Machine Learning Research*, 70:2554, 2017. 120

[122] J. Munro and D. Damen. Multi-modal Domain Adaptation for Fine-grained Action Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 75

[123] T. Nagarajan and K. Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018. 45, 46, 142, 143, 147, 148, 149

[124] Z. Nan, Y. Liu, N. Zheng, and S.-C. Zhu. Recognizing unseen attribute-object pair with generative model. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019. 45, 143

[125] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6752–6761, 2018. 5, 15, 16, 28, 84, 85, 91, 92, 93, 94, 95, 154

[126] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, 2005. 7

[127] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168, 2006. 29

[128] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016. 30

[129] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Learning joint representations of videos and sentences with web image search. In *European Conference on Computer Vision*, pages 651–667, 2016. 32

[130] B. Pang, K. Zha, and C. Lu. Human action adverb recognition: Adha dataset and a three-stream hybrid model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2325–2334, 2018. 44

[131] D. Parikh and K. Grauman. Relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 503–510. IEEE, 2011. 24, 45, 169

[132] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC)*, pages 1–12, 2015. 30

[133] P. Parmar and B. Morris. Action quality assessment across multiple actions. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1468–1476. IEEE, 2019. 36, 38, 39, 118, 119, 122, 123, 124, 125

[134] P. Parmar and B. T. Morris. Learning to Score Olympic Events. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 76–84. IEEE, 2017. 1, 36, 38, 39, 85, 122, 125

[135] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–313, 2019. 1, 36, 38, 116

[136] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1310–1318, 2013. 12

[137] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. 2019. 49, 156

[138] S. Paul, S. Roy, and A. K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018. 5, 16, 17, 28, 84, 161, 163

[139] W. Pei, T. Baltrusaitis, D. M. Tax, and L.-P. Morency. Temporal attention-gated model for robust sequence classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 14, 44, 84

[140] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 46, 141, 145

[141] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391, 2010. 29

[142] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156, 2010. 6

[143] M. Perše, M. Kristan, J. Perš, and S. Kovačič. Automatic evaluation of organized basketball activity using bayesian networks. In *Computer Vision Winter Workshop*. Citeseer, 2007. 37

[144] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 29

[145] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 29

[146] A. Piergiovanni, C. Fan, and M. S. Ryoo. Learning Latent Sub-events in Activity Videos Using Temporal Attention Filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 12, 13, 28, 44, 84

[147] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In *European Conference on Computer Vision (ECCV)*, pages 556–571, 2014. 1, 36, 37, 38, 47, 48

[148] W. Price. Two stream action CNN analysis - code. https://github.com/willprice/two-stream-action-cnn-analysis/, 2017. 74

[149] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations (ICLR)*, 2016. 120

[150] C. E. Reiley and G. D. Hager. Decomposition of robotic surgical tasks: an analysis of subtasks and their correlation to skill. In *M2CAI workshop. MICCAI, London*, 2009. 36, 40, 43

[151] A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5996, 2018. 18, 20

[152] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3202–3212, 2015. 32

[153] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, pages 1–28, 2015. 49

[154] J. Rosen, B. Hannaford, C. G. Richards, and M. N. Sinanan. Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. *IEEE Transactions on Biomedical Engineering*, 48(5):579–591, 2001. 36, 40, 43

[155] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–904, 2015. 12

[156] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1842–1850, 2016. 120

[157] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 30

[158] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36, 2004. 7

[159] D. Sculley. Large scale learning to rank. *NeurIPS 2009 Workshop on Advances in Ranking*, 2009. 24

[160] F. Sener and A. Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 862–871, 2019. 18

[161] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4480–4488, 2015. 18, 20, 130

[162] S. Sharma, R. Kiros, and R. Salakhutdinov. Action Recognition using Visual Attention. *Neural Information Processing Systems: Time Series Workshop*, 2015. 84

[163] Y. Sharma, V. Bettadapura, T. Plötz, N. Hammerla, S. Mellor, R. McNaney, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa. Video based assessment of osats using sequential motion textures. In *International workshop on modeling and monitoring of computer assisted interventions (M2CAI)-workshop*, 2014. 36, 40, 42, 44, 48

[164] Y. Sharma, T. Plötz, N. Hammerld, S. Mellor, R. McNaney, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa. Automated surgical osats prediction from videos.

In *IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 461–464, 2014. 36, 42

[165] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations (ICLR)*, 2017. 12

[166] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems (NeurIPS)*, pages 568–576, 2014. 8, 10, 44, 63

[167] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014. 13

[168] K. K. Singh and Y. J. Lee. End-to-End Localization and Ranking for Relative Attributes. In *European Conference on Computer Vision (ECCV)*, pages 753–769, 2016. 27

[169] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4077–4087, 2017. 120

[170] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 966–973. IEEE, 2010. 33

[171] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, volume 1, page 7, 2017. 12

[172] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 10, 11

[173] Y. Souri, E. Noury, and E. Adeli. Deep relative attributes. In *Asian Conference on Computer Vision (ACCV)*, pages 118–133, 2016. 24, 25, 26, 27, 169

[174] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1199–1208, 2018. 120

[175] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1057–1063, 2000. 121

[176] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 18

[177] G. W. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2729–2736. IEEE, 2011. 29

[178] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 77–86, 2008. 25

[179] B. Thompson. Canonical correlation analysis. *Encyclopedia of Statistics in Behavioral Science*, 2005. 31

[180] O. Tilk and T. Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*, 2016. 135

[181] A. Torabi, N. Tandon, and L. Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 32

[182] 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 (ICCV)*, pages 4489–4497, 2015. 10, 11, 13, 38, 44, 71, 122

[183] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5552–5561, 2019. 11

[184] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017. 75

[185] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 12, 143

[186] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3630–3638, 2016. 120

[187] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558, 2013. 6

[188] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC)*, 2009. 6

[189] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011. 6, 13, 44

[190] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1393, 2014. 29, 30, 31, 63

[191] K. Wang, X. Gao, Y. Zhao, X. Li, D. Dou, and C.-Z. Xu. Pay attention to features, transfer learn faster cnns. In *International Conference on Learning Representations (ICLR)*, 2020. 128

[192] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314, 2015. 13

[193] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5005–5013, 2016. 31

[194] 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 *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016. 9, 10, 14, 44, 61, 62, 63, 65, 66, 68, 70

[195] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4334, 2017. 5, 15, 16

[196] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(11):2740–2755, 2018. 9

[197] X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2120–2127, 2013. 45

[198] Z. Wang and A. M. Fey. Deep Learning with Convolutional Neural Network for Objective Skill Evaluation in Robot-assisted Surgery. *International Journal of Computer Assisted Radiology and Surgery*, 13(12):1959–1970, 2018. 36, 40

[199] Z. Wang, A. C. Bovik, and L. Lu. Why is image quality assessment so difficult? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–3313. IEEE, 2002. 35

[200] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2):121–132, 2004. 35

[201] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11293–11302, 2019. 127

[202] M. Wray, D. Larlus, G. Csurka, and D. Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 33, 141

[203] X. Xiang, Y. Tian, A. Reiter, G. D. Hager, and T. D. Tran. S3d: Stacking segmental p3d for action quality assessment. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 928–932. IEEE, 2018. 36, 38

[204] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. 21, 30, 33

[205] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 32, 141

[206] H. Xuan, A. Stylianou, and R. Pless. Improved embeddings with easy positive triplet mining. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2474–2482, 2020. 30

[207] Y. Yang, C. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 444–454, 2011. 30

[208] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 982–990, 2016. 25, 27, 63, 71, 85

[209] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014. 27

[210] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, 2015. 11, 71

[211] 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, 2007. 8, 66, 145

[212] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3712–3722, 2018. 110, 117, 127

[213] K.-H. Zeng, T.-H. Chen, J. C. Niebles, and M. Sun. Generation for user generated videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 609–625, 2016. 44

[214] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision (ECCV)*, 2016. 74

[215] H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba. Slac: A sparsely labeled dataset for action classification and localization. *arXiv preprint arXiv:1712.09374*, 2017. 82

[216] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 11

[217] L. Zhou, C. Xu, and J. J. Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 18, 21

[218] D. Zhukov, J.-B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3537–3545, 2019. 18, 19, 20, 21, 130, 133, 145

[219] A. Zia and I. Essa. Automated surgical skill assessment in rmis training. *International Journal of Computer Assisted Radiology and Surgery*, 13(5):731–739, 2018. 36, 48, 189

[220] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, M. A. Clements, and I. Essa. Automated assessment of surgical skills using frequency analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 430–438, 2015. 36, 40, 42, 44, 48

[221] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, and I. Essa. Video and Accelerometer-Based Motion Analysis for Automated Surgical Skills Assessment. *International Journal of Computer Assisted Radiology and Surgery*, 13(3):443–455, 2018. 1, 36, 40, 42, 44, 189

Appendix A

Method	Knot-Tying	Needle-Passing	Suturing
Zia *et al.* [219]	0.62	-0.19	0.37
This method	0.43	0.19	0.27

Table 1: *Comparison between the proposed method from Chapter 3 and a method using kinematic data specifically for surgical tasks. Methods are compared using Spearman's rank correlation where values range between -1 and 1.*

Although this thesis aims for a general method to determine skill, not specific to any task, its interesting to examine how close the proposed generic method comes to previous task-specific methods. The results on each of the Surgery tasks using the method from Chapter 3 are compared against the motion features proposed by Zia *et al.* [219]. Zia *et al.* use kinematic data from the surgical arms to extract several features: sequential motion textures, discrete Fourier transform, discrete cosine transform and approximate entropy. These features are then used to estimate the OSATS scores. The success of these features for predicting skill in surgical tasks is reliant on the repetitive and predictable nature of the tasks [219]. While it is possible to predict these features from video data [221], there are no published results on predicting the OSATS scores on the tasks within the JIGSAWS dataset.

To compare to this approach, 8-fold leave-one-user-out cross validation is used and Spearman's rank correlation (ρ) is employed to measure the correlation between predicted and ground-truth scores of the left out user. Results are available in Table 1. Although the proposed method outperforms the result from Zia *et al.* in Needle-Passing, it falls short in both Knot Tying and Suturing. From these results it can be concluded that there is a long way to go for a general skill determination method to reach the level of task-specific method performance. While the proposed method is suitable for skill determination in daily-tasks, task-specific approaches are more successful where available.