Towards Interaction-level Video Action Understanding

YANG BAI

Towards Interaction-level Video Action Understanding

Yang Bai

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
August 2023

Abstract

A huge amount of videos have been created, spread, and viewed daily. Among these massive videos, the actions and activities of humans account for a large part. We desire machines to understand human actions in videos as this is essential to various applications, including but not limited to autonomous driving cars, security systems, human-robot interactions and healthcare. Towards real intelligent system that is able to interact with humans, video understanding must go beyond simply answering "what is the action in the video", but be more aware of what those actions mean to humans and be more in line with human thinking, which we call interactive-level action understanding. This thesis identifies three main challenges to approaching interactive-level video action understanding: 1) understanding actions given human consensus; 2) understanding actions based on specific human rules; 3) directly understanding actions in videos via human natural language. For the first challenge, we select *video summary* as a representative task that aims to select informative frames to retain high-level information based on human annotators' experience. Through self-attention architecture and meta-learning, which jointly process dual representations of visual and sequential information for video summarization, the proposed model is capable of understanding video from human consensus (e.g., how humans think which parts of an action sequence are essential). For the second challenge, our works on *action quality assessment* utilize transformer decoders to parse the input action into several sub-actions and assess the more fine-grained qualities of the given action, yielding the capability of action understanding given specific human rules. (e.g., how well a diving action performs, how well a robot performs surgery) The third key idea explored in this thesis is to use graph neural networks in an adversarial fashion to understand actions through natural language. We demonstrate the utility of this technique for the *video captioning* task, which takes an action video as input, outputs natural language, and yields state-of-the-art performance. It can be concluded that the research directions and methods introduced in this thesis provide fundamental components toward interactive-level action understanding.

Declaration

The work contained within this thesis has been previously published in the following peer-review publications by the author and is used in the chapters as indicated below, no part of this thesis has been submitted elsewhere for any other degree or qualification

- **Chapter 3: "Query Twice: Dual Mixture Attention Meta Learning for Video Summarization."**
  Junyan Wang, **Yang Bai**, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, Xiaolin Wei. In *ACM International Conference on Multimedia*, 2020.

- **Chapter 4: "Action Quality Assessment with Temporal Parsing Transformer."**
  Yang Bai, Desen Zhou, Songyang Zhang, Jian Wang, Errui Ding, Yu Guan, Yang Long, Jingdong Wang. In *European Conference on Computer Vision (ECCV)* 2022.

- **Chapter 5: "Towards Cycle-counterfactual Action Quality Assessment."**
  Yang Bai, Haoran Duan, Zixian Gao, Junyan Wang, Yang Song, Hao Liu, Yang Long. Under review in *IEEE Transactions on Circuits and Systems for Video Technology.*

- **Chapter 6: "Discriminative Latent Semantic Graph for Video Captioning."**

  Yang Bai, Junyan Wang, Yang Long, Bingzhang Hu, Yang Song, Maurice Pagnucco, Yu Guan. In *ACM International Conference on Multimedia*, 2021.

Acknowledgements

I am deeply grateful to the individuals who have played significant roles in the completion of my PhD study and the development of this thesis. Their contributions and support have been invaluable, and I would like to express my sincere appreciation to each of them.

First and foremost, I extend my sincerest gratitude to my supervisor, Dr. Yang Long. His guidance and expertise in the field of computer vision have been instrumental in shaping my research. Dr. Long's insightful suggestions and encouragement have provided me with the necessary inspiration and direction to explore the intricacies of this field. I am truly grateful for the freedom he has given me to delve into the mysteries of computer vision.

I would also like to express my heartfelt thanks to my family for their unwavering support throughout my academic journey. Your constant encouragement and love have been a source of strength and motivation. I am truly grateful for everything you have done for me.

Furthermore, I would like to acknowledge Dr. Bingzhang Hu for his invaluable advice and inspiration regarding the application of deep learning techniques in the industry. His insights have broadened my perspective and enriched my research.

I am also indebted to my dear colleagues, Junyan Wang, Haoran Duan, and Peng Zhang, for their unwavering support and inspiration. They have provided invaluable assistance not only in my research endeavors but also in my personal life. Your encouragement during challenging times remains etched in my memory as a cherished and precious part of my life.

To all those who have contributed to my academic journey and the successful completion of this thesis, I am truly grateful. Thank you for your unwavering support, guidance, and inspiration.

Contents

List of Figures

List of Tables

Dedication

I dedicate this thesis to my beloved parents, my lovely wife and my daughter.

CHAPTER 1

Introduction

For hundreds of millions of years, humans have observed, interacted with, and made decisions about their environment. This ability to understand intricate, continuous actions and events has been critical throughout our evolution. From the precision timing required for hunting in ancient times, to the rules governing modern-day sports and music, our observations and interactions have been governed by a complex tapestry of understanding.

Computers have long perceived the world through vision, automatically extracting, analyzing, and comprehending information from digital images [2–5] [6–9] [10–13] [14–17]. In the realm of video, action recognition has been a focus for a long time [18–24]. However, only knowing what actions appeared in a short period of time is far away from human-level action understanding, where a specific object or motion appeared within a complex series of actions and events is only basic elements; what those motions and objects mean to humans is more important.

1.1 Motivation

The motivation driving this thesis is rooted in the pressing need for machines to attain a higher level of understanding when it comes to actions in videos, ultimately enabling them with enough perception ability for interacting seamlessly with humans. This imperative extends to a wide array of applications, including but not limited to autonomous driving, security systems, human-robot interactions, and healthcare.

To realize genuinely intelligent systems capable of interacting with humans, action understanding must transcend the mere ability to answer the question, "What is happening in the video?" Instead, it should align more closely with human cognition, a concept we refer to as "Interaction-Level" action understanding. In this context, "Interaction-Level" signifies a degree of comprehension that empowers machines to perceive and analyze complex movements and events on par with human capabilities, thereby unlocking the potential for machines to interact meaningfully with humans.

Interaction-level action understanding is a holistic approach that aims to enable machines to comprehend human actions in a way that mimics human-level understanding, this thesis identifies three key challenges: 1) understanding action based on human consensus, 2) understanding action based on specific human rules, and 3) understanding action via natural language, to achieve this level of understanding. Here's why tackling these challenges can contribute to achieving interaction-level action understanding:

Understanding Action via Human Consensus involves modeling human perception to align machine interpretations of actions with human consensus. By capturing the shared understanding of actions among multiple human annotators, we aim to account for the subjectivity and nuance that can exist in action interpretation. Additionally, we address the challenge of handling variability in individual interpretations, enhancing the robustness of machine models. By addressing this challenge, machine models can become more robust to individual variations in perception, thereby enhancing their overall comprehension. Finally, understanding ac-

tions based on human consensus enables the extraction of contextual information, going beyond mere action recognition to consider their relevance and significance within specific scenarios.

Understanding Action Based on Specific Human Rules. Many domains require actions to adhere to specific rules or criteria, and this approach focuses on enabling machines to understand actions in accordance with these rules. By doing so, machines can assess the quality, completeness, or compliance of actions with a level of precision akin to human experts. Fine-grained understanding is emphasized, allowing machines to differentiate between subtle variations and evaluate actions at a granular level. Also, many real-world applications, such as sports judging, medical procedures, or manufacturing processes, require machines to evaluate actions based on domain-specific rules. Addressing this challenge enables machines to excel in these specialized contexts.

Understanding Action via Natural Language bridges the gap between visual perception and human communication by allowing machines to understand actions through natural language. It facilitates the generation of action descriptions that align semantically with human descriptions, enhancing the interpretative capabilities of machines. Moreover, this approach enables machines to interpret actions within complex scenarios, conveying essential elements effectively. Effective communication and interaction between machines and humans are crucial aspects of this approach, as machines express their understanding of actions in a manner comprehensible to humans.

By addressing these challenges, machines can move beyond basic action recognition and achieve interaction-level understanding, where they not only recognize actions but also grasp their context, significance, and implications. This level of understanding is essential for machines to interact intelligently with humans in a wide range of applications, from autonomous systems to healthcare and natural language communication.

For the first challenge, this thesis chooses *video summary* as a representative task that aims to select informative frames to retain high-level information based on

human annotators' experience or consensus. It means that the algorithm needs to achieve a certain level of understanding about human consensus for a given video. For the second challenge, we explore action understanding via specific human rules with *action quality assessment*, which aims to assess the quality of actions in a given video with particular rules, such as assessing the quality of a diving action with the Olympic diving rule, determine the completeness of a robot operating a surgery with expert judgements. For the third challenge, this thesis investigates understanding action via human natural language by exploring the video captioning task, which aims to generate descriptions of actions and events based on the input video. However, there exist problems with modelling the above tasks and reaching better performance, and we summarise the challenges of the above tasks as follows:

- **Video Summary:** Video summary aims to select representative frames given an input action video sequence based on human consensus or experience, each video segment from the whole input sequence is associated with an importance score provided by human annotators. However, the importance scores are very subjective and highly related to human perception. We need to model and understand human consensus while being less influenced by individual variation. Besides, the annotations are more expensive to obtain than image-based tasks due to the increased temporal dimension. Also, each video is provided with multiple annotations from different annotators, making the annotation cost more expensive. Thus, the model should be able to cope with limited labelled data while retaining high generalization.

- **Action Quality Assessment:** Video Action Quality Assessment (AQA) aims to quantify the performance of actions given a specific rule. In contrast to the conventional action recognition tasks [25,26], AQA poses unique challenges due to the subtle visual differences. Since the videos to be evaluated usually are from the same coarse action category (e.g., diving) in AQA, it's crucial to capture *fine-grained intra-class variation* to estimate more accurate quality scores. To model intra-class variation, we need to represent the video in a set of atomic action patterns. However, it is difficult to obtain temporal part-level

annotations in practice. Take diving as an example; although a diving action consists of several sub-actions such as take-off and approach, the judges provide only the quality of the whole diving movement. Hence, the first challenge would be parsing the action into sub-actions without annotations. Second, the subsequent challenge would be extracting the quality-related representations of each parsed sub-action or atomic action pattern without knowing the sub-actions' quality ground truth.

- **Video Captioning:** To generate high-quality captions based on the given videos, we need to explicitly extract the object-level interactions and frame-level information from complex spatio-temporal image signals into high-level semantic information. Since there are many redundancies from adjacent frames, we need to keep objects that occasionally appear in the video while removing overlapping objects. Also, after extracting high-level representations from the image modal, we need to align those features with tokenized human natural language information, which is quite difficult under existing encoder-decoder based video captioning models.

Motivated by the above concerns and challenges, this thesis dedicates to exploring Interaction-Level action understanding via different video understanding tasks, namely video summary, action quality assessment and video captioning.

1.2 Contributions

This thesis makes several significant contributions towards Interaction-Level action understanding this section summarize the contribution in order of importance in terms of technical novelty as follows.

Chapter 6, Discriminative Latent Semantic Graph for Video Captioning unravels a captivating journey into semantic alignment in video captioning. The chapter presents a novel framework that astutely extracts spatio-temporal contexts from videos, embedding this information into object entities with learnable tokens. Additionally, it proposes methods to distill visual knowledge and condense redundant proposals into succinct visual words. In ensuring the cohesion of the output,

the chapter introduces validation techniques that assess the fidelity and readability of the generated captions, promising an unwavering semantic alignment across modalities. The idea of validation bright together Graph Neural Network and Generative Adversarial networks, which preserve the semantic meaning from input video to the generated captions.

In **Chapter 3, Query twice: Dual Mixture Attention Meta Learning for Video Summarization**, the focus is on modeling the subjective human perception. The introduced Dual Mixture Attention model (DMASum) seamlessly merges both visual and sequential information using a state-of-the-art self-attention architecture. Moreover, the chapter innovates a high-level semantic understanding via a meta learning module, enhancing the efficiency of the training data and bolstering model generalization.

Chapter 4, Action Quality Assessment with Temporal Parsing Transformer brings to light the essence of fine-grained action understanding. Here, the temporal parsing transformer is introduced, which adeptly decomposes holistic features into temporal part-level representations using a set of learnable queries. Further enriching this approach, the chapter offers two novel loss functions tailored for the decoder's cross attention responses: one ensures temporal order, and the other encourages discriminative part representations.

The innovations continue in **Chapter 5, Towards Cycle-Counterfactual Action Quality Assessment**. This chapter delves into the realm of quality scores estimation, proposing a generative cycle counterfactual framework. By meticulously disentangling sub-action features into quality and context attributes, and generating counterfactual samples that tweak a sub-action's quality attribute, this approach paves the way for a cycle framework. Such a framework insists on a model's use of sub-action quality, in turn, enriching its learning space.

1.3 Thesis Outline and Summary

The rest of this thesis is organized as follows. Chapter 2 offers a literature review on fundamental deep learning techniques. Emphasis is placed on state-of-the-art

deep learning architectures for video backbone networks. The focus of Chapter 3 is on the challenge of understanding action via human consensus, where the challenge is explored with video summary task. Chapters 4 and 5 are trying to explore the research question brought by Action Understanding based on Specific Human Rules. In Chapter 4, the action quality assessment task is introduced along with the temporal parsing transformer. Chapter 5 presents the generative cycle counterfactual framework, which aims to better estimate quality scores using variations of sub-actions. Chapter 6 explores the video captioning task. A new framework consisting of three sub-tasks is introduced to extract and align high-level object and motion semantics from videos with natural language descriptions. Chapter 7 concludes by summarizing its contributions and suggesting directions for future research.

To summarize, this PhD thesis revolves around the central pursuit of achieving advanced Interaction-Level action understanding in machines. The primary research question addressed is: How can machines be taught to understand and interpret actions in videos at a depth similar to human cognition? To answer this pivotal question, the study breaks down the challenge into three integral parts, each representing a dimension of human understanding:

Video Summarization: This mirrors the human ability to derive consensus. By focusing on selecting representative frames from video sequences based on human consensus, it aims to make machines grasp the essence of a video just as humans would perceive and interpret the most crucial moments.

Action Quality Assessment: Humans inherently evaluate actions based on set criteria or rules. To imbue machines with this capability, this part evaluates the quality, compliance, or completeness of actions. It's not just about recognizing an action, but also assessing its quality according to certain standards or rules, akin to how a human expert might evaluate a performance.

Video Captioning: Language is a quintessential human trait, used to describe and communicate complex scenarios. By enabling machines to generate descriptions of video content in natural language, this part bridges the gap between visual perception and human language. This would allow machines to not only understand actions but also convey this understanding in terms humans can relate to.

Together, these three components weave a comprehensive fabric of action understanding. While each focuses on a distinct facet of human perception consensus, rule-based evaluation, and linguistic description they collectively strive to achieve the overarching goal of the research: endowing machines with a rich, nuanced understanding of actions in videos, parallel to human interaction levels.

CHAPTER 2

Background

2.1 Introduction

With the development of deep learning techniques and computational power, in recent years, deep learning has achieved great success in various domains such as computer vision [2, 4, 5, 27], natural language processing [28–30], speech recognition [31–33], sensor-based applications [34–37] and robotics [38, 39]. The fundamental architectures of applications from these domains also play essential roles in the video domain. In the rest of this chapter, Section 2.2 reviews various of fundamental deep learning architectures and highlights their applications in the video domain. Section 2.3 reviews papers proposed to solve the video action recognition task, where the task is the foundation of the video domain as it is typically adopted for studying and evaluating video backbone networks. These video action recognition models can be adapted to extract spatio-temporal features in various video-based computer vision tasks such as temporal action localization [40–42], action detection [43–46] and tasks explored in this thesis. Section 2.4 reviews research area of 3D video understanding. Section 2.5 reviews state-of-the-art methods related to Interaction-level action understanding.

2.2 Deep Learning

2.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the most important deep learning architectures, capable of capturing position-invariant patterns in images, thereby reducing the number of parameters and improving generalization compared with Multilayer Perceptron(MLP). Recently, CNNs have made groundbreaking results in many fundamental computer vision tasks such as image classification [2, 4, 5, 27], object detection [6–9], instance segmentation [14–17], image inpainting [47–49], etc. In 1998, LeCun *et al.* proposed LeNet to solve the handwritten digit recognition task with the proposed MINIST dataset [27]. LeNet first clearly define the basic components in CNN, such as convolution, pooling, fully-connected etc. To solve more challenging vision tasks, in 2012, Alex *et al.* [5] first showed the effectiveness of the CNN-based AlexNet on the challenging ImageNet dataset [50], where AlexNet had deeper network architecture consisting of five convolutional layers and three fully-connected layers. Then, Simonyan *et al.* [4] introduced VggNet in 2014 with extended network depth and better performance compared with AlexNet. Same year in 2014, GoogleNet was proposed to further improve the performance on Imagenet dataset, where the inception architecture was introduced to extend the width and depth of networks without consuming more computational power [3]. To further increase network depth, He *et al.* [2] proposed ResNet with the residual connection that perverse deep network from degradation as the network depth increased, making networks with more than a hundred layers possible.

CNNs have also achieved great success in the video domain. Karpathy *et al.* [18] tried to solve large-scale action recognition tasks by extracting video frame-level features with 2D CNNs. Tran *et al.* [24, 51] proposed to extract spatio-temporal features via 3D convolutional neural networks.

2.2.2 Recurrent Neural Networks

Recurrent Neural Networks(RNN) is a class of neural networks for sequential modelling, as RNN maintain the internal states to process elements of the input sequence

one after another [52–54]. Since RNN suffer from gradient exploding and gradient vanishing, the Long Short Term Memory(LSTM) [29] was proposed with multiple internal gates that preserve flow of information, so to elimnate gradient exploding and vanishing. The gates mechanisms also provide LSTM with abiliy of handling long-term dependencies, which makes LSTM popular in Natural Language Processing tasks such as machine translation, automatic summarization, text classification, question answering, etc. To reduce the number of parameters and speed up training, Gated Recurrent Unit (GRU) [55], a variant of LSTM with fewer gates, was proposed and achieved good performance in NLP tasks.

In the video domain, since the input is composed of a series of images, it is more natural and appropriate to apply RNN models such as LSTM for image sequence modelling in varies video tasks such as action recognition [19, 20], video action detection [56, 57], video summary [58], video captioning [59–61], etc.

2.2.3 Adversarial Learning

The Generative Adversarial Networks (GANs) was first introduced by Goodfellow *et al.* [62] for image generation. GANs transform the problem of unsupervised generation into a supervised problem by employing two sub-models: a generator, which generates new data, and a discriminator, which aims to distinguish real samples (from training data) from fake samples (generated data). Two sub-models with independent parameters are alternately trained in a zero-sum game adversarially until the generator is capable of generating fake samples to fool the discriminator. The effectiveness of the GANs encouraged many works to experiment with adopting the adversarial training fashion for various generation tasks such as super-resolution [63–65], text-to-image generation [66–68], image captioning [69], image-to-image translation [70–72], image inpainting [73, 74] and etc.

For video-based tasks, many works experimented to adopting GANs for video content generation considering the sptatio-temporal information such as video prediction [75–77], video frame interpolation [78–80], video inpainting [81–83] and etc. Another branch of video-based generation is generating natural language based on video content, such as video captioning [69,84,85], video question answering [86], etc.

Some works employed GANs to keep the generated languages fluent and realistic. Chapter 6 also adopt GANs as a language validation for better caption generation.

2.3 Video Action Recognition

2.3.1 Conventional Approaches

Although deep learning methods have become mainstream for video-based action recognition, several conventional hand-crafted methods proposed for learning the pattern of motion and appearance in videos are still applied in some real-world applications nowadays. In 2005, the Space-Time Interest Points (STIP) was introduced by Laptev for extracting spatio-temporal patterns into compact and abstract representations in videos [87]. Wang et al. [88] proposed the Dense Trajectory Feature (DTF), which is contracted based on displacement information from dense optical flow, resulting from dense points sampled from each frame. Besides, a descriptor to encode the trajectory information based on Motion Boundary Histograms (MBH) was introduced to further improve the performance. To improve the inaccurate information brought by DTF, Wang *et al.* [88] took camera motion into account and proposed the improved Dense Trajectory Feature (iDTF). They also integrated iDTF with Fisher Vector (FV) encoding [89, 90].

2.3.2 Deep Learning Methods

Different from directly applying 2D CNN networks to extract image features, as illustrated in the previous section, the video modality has an extra temporal dimension compared with image modality, which leads to more complex feature extraction and correspondingly more types of models. Figure 2.1 shows typical video feature extraction models, from native solutions to state-of-the-art I3d [26] networks.

2D ConvNets with Average Pooling. A straightforward yet efficient approach to encoding video frames for action recognition is directly applying 2D CNNs proposed for image classification to extract features for individual video frames to 1D vectors [18]. Then based on the sequence of 1D vectors, action class labels can be

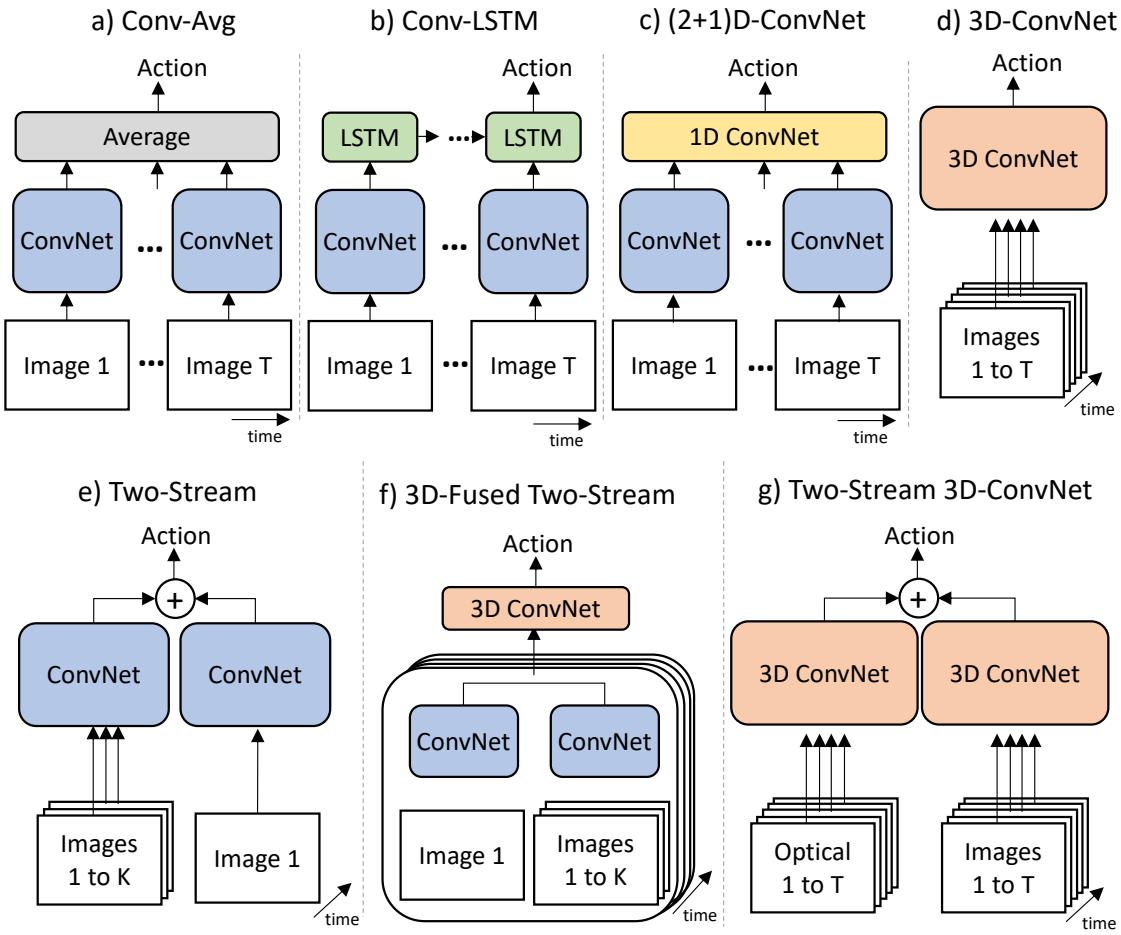


Figure 2.1: An illustration of a series of representative video action recognition backbones that considered in this chapter. T indicates the total number of video frames, while K represents the number of frames within a neighbouring subset. Optical indicates Optical flow.

derived by pooling their predictions across the whole sequence along the temporal dimension, as shown in Figure 2.1 a). Although this method is simple and efficient in practice, the temporal information is entirely ignored, leading to inaccurate prediction for order-sensitive actions such as opening and closing a window.

2D ConvNets with RNNs. To model temporal information based on the extracted sequence of frame features, a more reasonable approach is to adopt recurrent layers, such as LSTMs [19,20], which is capable of capturing temporal structure and keeping long-range dependencies as shown in Figure 2.1 b). LSTMs on extracted features from the last layers of CNNs can capture high-level temporal information,

but the backpropagation-through-time based on multiple frames for optimizing the LSTMs makes it hard and expensive to train. Besides, compared with CNN based networks such as ResNet with more than a hundred layers for processing information from low-level texture to high-level semantic, shallow LSTMs limit the richness of processed features. In Chapter 3, this type of network for time-seris feature extraction is adopted, for effecient spatio-temporal information extraction, which can provide information-rich features and maintain low computational consumption.

(2+1)D ConvNets. To model the temporal feature while reducing training difficulty, Tran *et al.* [21] proposed (2+1)D Convolution, which replaced LSTMs with 1D temporal convolutions on top of 2D spatial convolutions (shown in Figure 2.1 c)), yields signicantly better performance than previous backbone models. Although adopting temporal models such as LSTMs and 1D ConvNets on features extracted from 2D spatial convolutions can encode temporal information, such type of models are challenging to capture fine-grained low-level motion, which is essential in considerable circumstances.

3D ConvNets. 3D ConvNets seem like a practical path to extract spatio-temporal information from videos, where the networks are similar to 2D convolutional architectures but with spatio-temporal convolutional filters. Hence, 3D ConvNet basedmethods have been well explored [22–24], and the typical structure of 3D ConvNets video backbone is shown in Figure 2.1 d). However, these models have many more parameters than previously introduced methods due to the additional filter dimension, which leads to harder training processes. The models proposed in this thesis avoid the use of 3D CNNs because for interactive-level action understanding scenarios, the number of frames is much higher than for action recognition tasks, and we will incur more computational consumption.

Two-Stream Networks. Simonyan *et al.* [91] proposed another branch of solution for modelling detailed motion information while keep the number of parameters relatively low. The method introduced the optical flow modality to jointly predict action labels with image modality, where the optical flow contains detailed motion information. To be specific, the model derives predictions by averaging the output labels from a single image frame and a set of neighbouring pre-computed optical

flow frames processed by two individual 2D CNNs, as illustrated in Figure 2.1 e). During test time, the model samples multiple snapshots from a given video, and the predicts action labels are then averaged. Later, [92] proposed to fuse the image and optical flow streams at the last convolutional layers, proving some performance improvement. Based on this architecture, [26] proposed the Two-Stream Inflated 3D ConvNets(I3D) network that convert 2D ConvNets into 3D ConvNets by endowing all the filters and kernels with an additional temporal dimension. The I3D architecture has been proven the capability of capturing spatio-temporal information in many video-based tasks such as temporal action localization [40–42], video captioning [59–61], etc. The models proposed in Chapter 5, Chapter 6 and Chapter 7 also adopted I3D as the spatio-temporal feature extraction backbone.

2.4 3D Video Understanding

Understanding videos in 3D space is an important realm of computer vision, promising richer interactions and deeper comprehension of action sequences. The integration of 3D video understanding offers a potential alternative for interaction-level action comprehension.

2.4.1 3D Video Object Detection

LiDAR-based techniques for 3D video object detection [93,94] commonly align point clouds from sequential frames by compensating for ego-motion, accumulating these clouds to address their inherent sparsity. Recently, object-level approaches [95, 96] that handle multi-frame point clouds of tracked objects have gained traction. Despite its potential, 3D object detection from monocular videos remains relatively under-explored. Kinematic3D [97] stands out as a pioneering effort that disentangles kinematic information into ego-motion and object-targeted motion. Notably, it leverages a 3D Kalman Filter [98] for kinematic modeling, focusing on short-term temporal associations (limited to 4 frames). BEVFormer [99] introduces an attention-based transformer technique to assess spatial and temporal relationships from a birds-eye perspective. Concurrently, DfM [100], drawing from Multi-view

Geometry, regards two frames in stereo, using the cost volume in the stereo setup to determine depth. Nevertheless, addressing moving objects within this framework remains an unresolved challenge.

2.4.2 Geometry in Videos

3D geometry in videos offers a robust tool for scene reconstruction and camera pose estimation, standing as a cornerstone in computer vision. Structure from Motion (SfM) [101] and Multi-view Stereo (MVS) [102] represent two approaches for estimating sparse and dense depths from multiple views, respectively. Within robotics, 3D geometry principles underpin Simultaneous Localization and Mapping (SLAM) applications [103]. For the global optimization of 3D feature point positioning and consistent camera poses, the bundle adjustment algorithm [104] has become a popular choice. However, these methodologies often struggle with dynamic scene elements. In the contemporary deep learning landscape, the rise of object detection has ushered in the era of object-level semantic SLAM [105,106], focusing on object reconstruction rather than entire scenes. This shift allows for dynamic scene management and enhanced object localization within videos. Additionally, the domain of feature correspondence learning [107] has seen significant advancements, with deep learning reshaping the feature matching process. Innovations like BANet [108] have rendered the entire 3D geometry system end-to-end trainable. Our work diverges from these approaches; we emphasize the 3D object's representation and incorporate feature correspondence learning into 3D object detection. By leveraging learned temporal feature correspondences, our proposed BA-Det refines the object pose for a tracklet in each frame.

2.5 Towards Interaction-level Video Action Understanding

Delving into video actions at an interaction-level yields nuanced insights into intricate scenes. This thesis decomposes the primary research question into three distinct

challenges, addressing them through the lenses of video summarization, action quality assessment, and video captioning. The subsequent subsections elaborate on how advancements in these areas collectively bolster our understanding of interaction-level video actions.

2.5.1 Video Summarization

Video, as a media containing complex spatio-temporal relationship of visual contents, has a wide range of applications [6, 109–113]. However, because of its huge volume, video summarization is to compress such huge volume data into its light version while preserving its information. Early works have presented various solutions to this problem, including storyboards [114, 114–116] and objects [117–119]. LSTM-based deep learning approaches are proposed for both supervised and unsupervised video summarization in recent years. Zhang *et al.* [120] proposed a bidirectional LSTM model to predict the importance score of each frame directly, and this model is also extended with determinantal point process [121]. Mahasseni *et al.* [122] specified a generative adversarial framework that consists of the summarizer and discriminator for unsupervised video summarization. The summarizer is an auto-encoder LSTM network for reconstructing the input video, and the discriminator is another LSTM network for distinguishing between the original video and its reconstruction. Based on reinforcement learning, Zhou *et al.* [123] proposed a deep summarization network which applies diversity and representativeness jointly for generated summaries. Besides, auxiliary resources have been employed in the summarization process recently. Zhang *et al.* [124] proposed a sequence learning model with an additional "retrospective encoder" which employed a pre-trained single-layer LSTM for shot-boundary detection with another disjoint dataset. For example, Wei *et al.* [125] developed a semantic attended video summarization network that employed the information of human-annotated text of the original video. Meanwhile, based on the observation of Otani *et al.* [126], they propose another evaluation approach as well as a visualization of correlation between the estimated scoring and human annotations.

Many prevailing methods adopted on TVSum and SumMe as their primary eval-

uation datasets. Despite this, the results garnered often fall short of human-level performance. Two major challenges persist in this domain. The first pertains to effectively modeling the inherent subjectivity among different annotators. The second challenge revolves around achieving robust generalization when working with limited data, a constraint imposed by the high costs associated with annotation.

In contrast to previous research approaches, Chapter 3 of this thesis postulates that attention models are inherently more suited for the video summarization task. Further, within Chapter 3, we elucidate the issues stemming from the softmax bottleneck and advocate for the integration of meta-learning to bolster model generalization. Notably, our work represents a pioneering effort in amalgamating Meta Learning strategies within the video summarization domain.

2.5.2 Action quality assessment

In the past years, the field of action quality assessment (AQA) has been repaid developed with a broad range of applications such as health care [127], instructional video analysis [128,129], sports video analysis [130,131], and many others [132,133]. Existing AQA methods can be categorized into two types: regression based methods and ranking based methods.

Regression based methods Mainstream AQA methods formulate the AQA task as a regression task based on reliable score labels, such as scores given by expert judges of sports events. For example, Pirsiavash et al. [134] took the first steps towards applying the learning method to the AQA task and trained a linear SVR model to regress the scores of videos based on handcrafted features. Gordan et al. [132] proposed in their pioneer work the use of skeleton trajectories to solve the problem of quality assessment of gymnastic vaulting movements. Parmar et al. [135] showed that spatiotemporal features from C3D [136] can better encode the temporal representation of videos and significantly improve AQA performance. They also propose a large-scale AQA dataset and explore all-action models to better evaluate the effectiveness of models proposed by the AQA community. Xu et al. [137] proposed learning multi-scale video features by stacked LSTMs followed [135]. Pan et al. [138] proposed using spatial and temporal graphs to model the interactions be-

tween joints. Furthermore, they also propose to use I3D [26] as a stronger backbone network to extract spatiotemporal features. Parmar et al. [131] introduced the idea of multi-task learning to improve the model capacity of AQA, and collected AQA datasets with more annotations to support multi-task learning. To diminish the subjectiveness of the action score from human judges, Tang et al. [139] proposed an uncertainty-aware score distribution learning (USDL) framework Recently. Wang et al. [140] introduced the TSA-Net, which incorporates a single object tracker for AQA, leading to the development of the Tube Self-Attention Module (TSA). This module is adept at efficiently generating rich spatio-temporal contextual information through sparse feature interactions. However, it's worth noting that the TSA-Net requires an external dataset to train the object tracker. If other methods were to leverage similar datasets, they might also experience a boost in performance.

Though the above methods reached relatively good performance, the video's final score can only provide weak supervision concerning action quality. Because two videos with different low-quality parts are likely to share similar final scores, which means the score couldn't provide discriminative information.

Ranking based methods Another branch formulates AQA task as a ranking problem. Doughty et al. [128] proposed a novel loss function that learns discriminative features when a pair of videos exhibit variance in skill and learns shared features when a pair of videos show comparable skill levels. Doughty et al. [129] used a novel rank-aware loss function to attend to skill-relevant parts of a given video. However, they mainly focus on longer, more ambiguous tasks and only predict overall rankings, limiting AQA to applications requiring some quantitative comparisons. Recently, Yu et al. [1] proposed the Contrastive Regression (CoRe) framework to learn the relative scores by pair-wise comparison, highlighting the differences between videos and guiding the models to learn the key hints for assessment. But the CoRe framework still works on holistic features, which igores the fine-grained details. Different from the above methods, frameworks proposed in Chapter 4 and 5 more focus on fine-grained features of sub-actions as well as the quality of sub-actions without any finer level supervision.

2.5.3 Video Captioning.

Video captioning, a confluence of Computer Vision (CV) and Natural Language Processing (NLP), has magnetized burgeoning research scrutiny. Early endeavors in this realm predominantly hinged on template-based language models [141, 142].

The deep learning revolution ushered in a paradigm shift, orienting methods towards an encoder-decoder structure for sequence learning [143–145]. Notably, some considered video captioning analogously to a machine translation task [143]. Yao *et al.* 's seminal contribution [144] leveraged a temporal attention mechanism, crafting a dynamic visual features summarization per generated word.

In the current research landscape, object-level intricacies have emerged as a focal point [59–61]. For instance, [60] incorporated dual LSTM layers, sculpting temporal structures at both the frame and object echelons. [61] ventured into visual reasoning across spatial-temporal continuums, while Zhang *et al.* 's work [59] brought forth an object-aware aggregation via the bidirectional temporal graph (OA-BTG), capturing granular temporal dynamics of salient video objects.

To encapsulate, traditional approaches emphasized global information or the temporal architecture of conspicuous objects. We posit that our D-LSG framework ignites three pivotal research inquiries in video captioning. The debut of interaction manipulation amongst diverse objects through graph modeling hints at a novel video captioning trajectory. Our trailblazing conditional graph operation uniquely amalgamates heterogeneous features from an array of base models. Furthermore, the extraction of visual knowledge from bolstered object proposals resonates with the age-old Bag-of-Visual-Words (BoVW) paradigm. Augmenting this, our dynamic graph's integration into end-to-end training enhances efficiency. Lastly, we underscore the necessity for nuanced supervision in sentence validation within the discriminative model, ensuring the preservation of both fidelity and structural integrity.

CHAPTER 3

Query Twice: Dual Mixture Attention Meta Learning for Video Summarization

Comprehending Actions through Human Consensus entails aligning machine interpretations of actions with collective human perspectives. To address this challenge, this chapter designates video summarization as the emblematic task. Its objective is to select informative frames that encapsulate high-level information, mirroring the insights and consensus of human annotators.

The video summarization problem addressed in this thesis pertains to the selection and extraction of salient and representative frames or segments from a video to create a condensed version, ensuring that the primary content and context of the original video are retained. Specifically, our focus lies in fully-supervised video summarization, where the objective is to align the summarized content with human annotations or consensus, ensuring that machine-generated summaries closely resonate with human perception and expectations. The aim is not just compression, but also enhancing the interpretability and relevance of the content based on human-centric criteria.

Video summarization is usually solved by predicting the segment-wise importance score via a softmax function. However, softmax function suffers in retaining

high-rank representations for complex visual or sequential information, which is known as the *Softmax Bottleneck* problem. In this chapter, we propose a novel framework named Dual Mixture Attention (DMASum) model with Meta Learning for video summarization that tackles the softmax bottleneck problem, where the Mixture of Attention layer (MoA) effectively increases the model capacity by employing twice self-query attention that can capture the second-order changes in addition to the initial query-key attention, and a novel Single Frame Meta Learning rule is then introduced to achieve more generalization to small datasets with limited training sources. Furthermore, the DMASum significantly exploits both visual and sequential attention that connects local key-frame and global attention in an accumulative way. We adopt the new evaluation protocol on two public datasets, SumMe, and TVSum. Both qualitative and quantitative experiments manifest significant improvements over the state-of-the-art methods.

3.1 Introduction

With the tremendous growth of video materials uploaded to various online video platforms like YouTube, automatic video summarization has received increasing attention in recent years. The summarized video can be used in many scenarios such as fast indexing and human-computer interaction in a light and convenient fashion. The main objective of video summarization is to shorten a whole video into summarized frames while preserving crucial plots. One of the mainstream directions focuses on key-frames summarization [146] is illustrated in Fig. 3.1 A video is first divided into 15-second segments, and the problem is modeled as an importance score prediction task to select the most informative segments.

The nature of video summarization task encourages a line of research [122, 125, 147, 148] focusing on unsupervised learning methods. Besides, [123] applied deep reinforcement learning with a diversity-representativeness reward function for the generated summary; Currently, the most popular benchmarks are SumMe [146] and TVSum [149]. Otani *et al.* [126] proposed to evaluate the methods by using the rank-order correlation between predicted and human-annotated importance scores. These

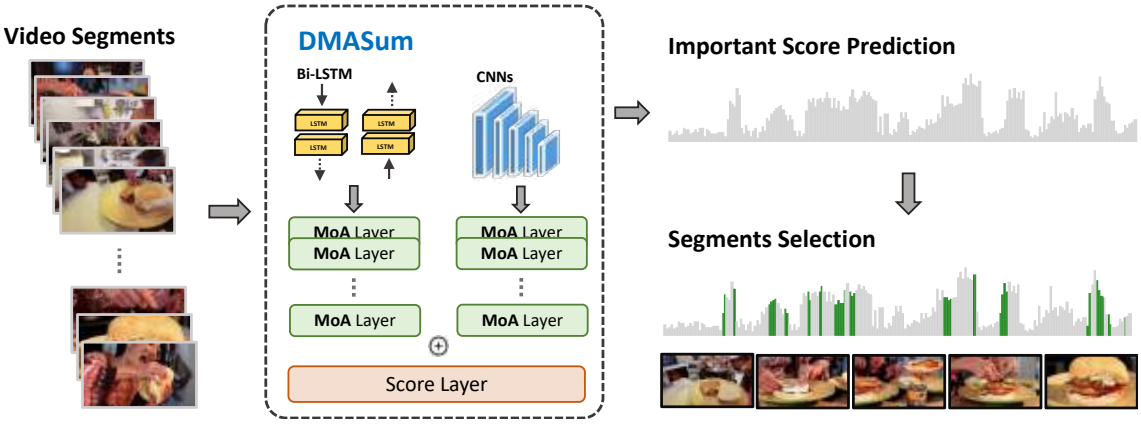


Figure 3.1: An illustration of the video summarization task using our proposed DMASum. Each gray bar represents the predicted important score of a segment and green bars denote the key-segments in the summarized video. Highlights of DMASum include Visual-sequential Dual Channels, Stacked MoA modules.

key evaluation matrices measure agreements between generated summaries and reference summaries. Therefore, supervised methods [115, 120, 125, 150] are still very important for investigating essential technical questions because they can directly compare against human-annotated scores as ground truth. One of the mainstream directions focuses on key-frames summarization [146] is illustrated in Fig. 3.1.

The challenges for supervised key-frames summarization are two-fold. First, the importance scores are very subjective and highly related to human perception. Second, the annotations are expensive to be obtained; thus, the model should be able to cope with limited labeled data while retaining high generalization. These are not only unsolved questions for video summarization but also essential for many other research domains. To this end, this chapter proposes a new framework, namely the Dual Mixture Attention model (DMASum) that aims to achieve **1)** human-like attention by adopting cutting-edge self-attention architecture and takes both visual and sequential information into a unified process; and **2)** high-level semantic understanding of the whole content by incorporating a novel meta learning module to maximally exploit the training data and improve the model generalization.

The proposed framework manifested promising results in our early experiments. However, the early implementation reflected two major technical challenges. The first is known as the *Softmax Bottleneck* problem associated with the self-attention architecture. In video summarization, there is a need to process long and complex

videos, each potentially consisting of numerous and diverse frames. The traditional softmax function, when used to represent attention distribution across these many frames, struggles to capture the high-rank representations efficiently. This is because softmax tends to accentuate larger values and suppress smaller ones, leading to loss of nuanced details essential for summarizing intricate videos. Both theoretical and empirical evidences in this chapter show that traditional softmax function does not have the sufficient capability to retain high-rank representation for long and complex videos. To this end, we propose a *Query Twice* module by adding self-query attention to query-key attention. The Mixture of Attention layer can then compare the two attentions to capture the second-order changes and increase the model capability. The second problem is that the most common meta learning strategy does not naturally fit the video summarization task. Because meta-learning, in its standard form, is optimized for few-shot learning settings. This means it is designed for scenarios where there's a limited number of examples to learn from, divided into a query set and a support set. In contrast, video summarization does not naturally lend itself to such divisions; it primarily deals with a singular training dataset. Given that video summarization lacks the structured divide of query and support sets, traditional meta-learning can't be straightforwardly applied. This divergence from the usual meta-learning setup can lead to challenges in training and model optimization. To mitigate this, the framework proposes a Single-video Meta Learning rule to refrain the learner tasks so as to purify the meta learner updating processes. To summarize our contributions:

- To our best knowledge, this is the first paper that successfully introduces self-attention architecture and meta learning to jointly process dual representations of visual and sequential information for video summarization.

- We provide in-depth theoretical and empirical analyses of the Softmax Bottleneck problem when applying attention model to video summarization task. And a novel self-query module with Mixture-of-Attention is provided as the solution to overcome the problem effectively.

- We explore the meta learning strategy, and a Single-Video Meta Learning rule

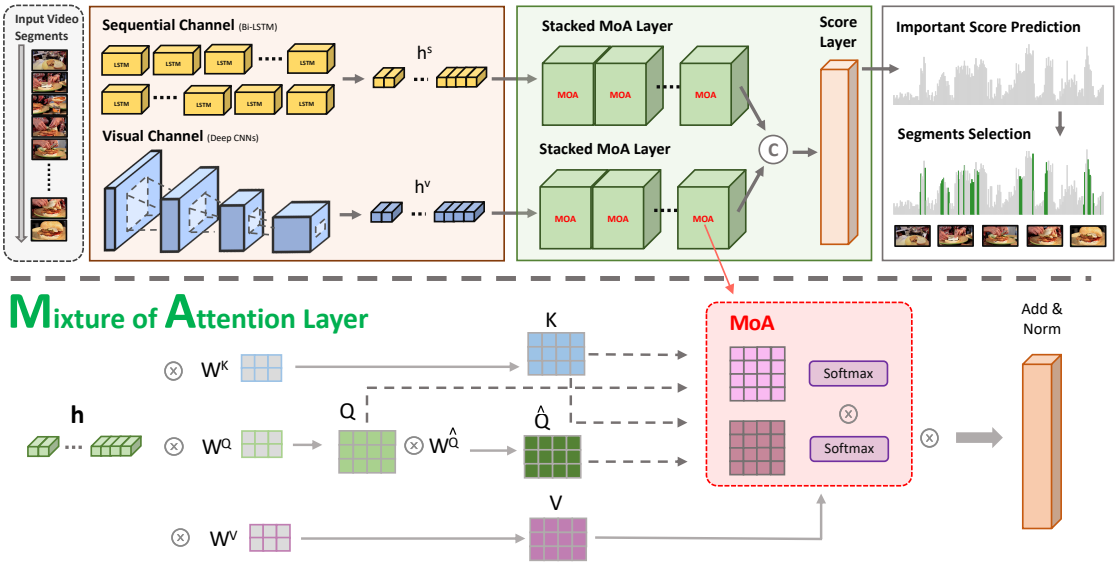


Figure 3.2: The overall architecture of our DMASum is shown as the top figure, which consists of a sequential channel and a visual channel and stacked MoA layers. The bottom part shows the structure of the Mixture of Attention layer.

is particularly designed for video summarization tasks.

- Quantitatively and qualitatively experiments on two datasets: SumMe [146] and TVSum [149] demonstrate our superior performance over the state-of-the-art methods. More impressively, our model achieves human annotator level performance under new protocols of Kendall's τ correlation coefficients and Spearman's ρ correlation coefficients. The groundbreaking results suggested that our DMASum has effectively modeled human-like attention.

3.2 Related Work

Attention-based Models. The attention mechanism was born to help memorize long source sentences in neural machine translation [151]. Rather than building a single context vector out of the translation encoder, the attention method is to create shortcuts between the context vector and the entire input sentence, then customize the weights of these shortcut connections for each element. The Transformer [30], without a doubt, is one of the most impressive works in the machine translation task. The model is mainly built on self-attention layers, also known as intra-attention, and the self-attention network is relating different positions of the

same input sequence. Many recent works have applied self-attention to a wide range of video-related applications, such as video question answer [152] and video captioning [153]. Particularly for the video summarization task, Ji *et al.* [154] proposed an attention-based encoder-decoder network for selecting the key shots. He *et al.* [155] proposed an unsupervised video summarization method with attentive conditional Generative Adversarial Networks.

However, a common limitation observed in these methods is their reliance on the softmax function. Notably, the softmax function often struggles to maintain high-rank representations, especially when dealing with intricate visual or sequential data. This phenomenon, termed the 'Softmax Bottleneck' problem, suggests that attention mechanisms, as currently implemented, may not always capture sufficient information, especially in complex scenarios."

Meta Learning. Meta learning, also known as learning to learn, aims to design a model that can be learned rapidly with fewer training examples. Meta learning usually used in few-shot learning [156, 157] and transfer learning [158]. Finn *et al.* [156] propose a Model Agnostic Meta Learning (MAML) which is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. Like MAML, the work of Nichol *et al.* [157] proposed a strategy which repeatedly sampling and training a single task, then moving the initialization towards the trained weights on that task. Recently, meta learning methods have been applied in a few video analysis tasks. Especially in video summarization, Li *et al.* [159] proposed a meta learning method that explores the video summarization mechanism among summarizing processes on different videos. However, when applying MAML-like meta learning modules into our framework, we discover a unique challenge for video summarization. In learning subtasks, the combined batch sampling tends to confuse the learner due to different video contents. Therefore, we propose key technical changes to impose a Single-Video Meta Learning rule to make subtasks of learning of meta learning more efficient.

3.3 The Proposed Approach

Video summarization is modeled as a sequence labeling (or sequence to sequence mapping) problem. Given a sequence of video frames, the task is to assign each frame an importance score based on which key-frames can be selected. Existing sequence labelling approaches include deep sequential models such as LSTM [120,124], attention model [154]. However, the key difficulty is to learn the frame dependencies within the video and capturing the internal contextual information of the video. Considering video is a highly context-dependent source that shares many similar properties in sentences. As the outstanding performance of the Transformer [30], we introduce the self-attention structure that has been widely used in natural language processing (NLP) as our architecture basis. Both visual and sequential representations are considered in order to model complex human-like attention and better match the subjective annotations. Also, the motivation of meta learning aims to improve the model generalization when training sources are insufficient due to expensive human annotations. An overview of the proposed video summarization architecture and the details of the Mixture of Attention layer that are illustrated in figure 3.2.

3.3.1 Architecture Design

Dual-representation Learning: For the video summarization task, we introduce both visual and sequential channels as the input. The visual channel (deep CNNs) extracts visual features $\boldsymbol{H}_v = \{\boldsymbol{h}_t^v\}_{t=1}^T$ from each video frame image. Based on the extracted visual features, the sequential features $\boldsymbol{H}_s = \{\boldsymbol{h}_t^s\}_{t=1}^T$ is obtained by the sequential channel (bidirectional LSTM network) and consists of the dual-channel feature $\boldsymbol{H} \in \{\boldsymbol{H}_v, \boldsymbol{H}_s\}$. The dual representation is critical to model complex human-like attention and can link frame-wise attention to the overall story line.

The Attention Module: Taking a feature sequence $\boldsymbol{H} = \{\boldsymbol{h}_t\}_{t=1}^T \in \mathbb{R}^{D\times T}$ extracted from the video as input, the attention network can re-express each $\boldsymbol{h}_t^*$ within input $\boldsymbol{H}$ by utilizing weighted combination of the entire neighborhood from $\boldsymbol{h}_1$ to $\boldsymbol{h}_T$, where D is the feature dimension and T is number of frames within

a video. In concreteness, the attention network first linearly transforms $\boldsymbol{H}$ into $\boldsymbol{Q} = \boldsymbol{W}_Q\boldsymbol{H}^*$, $\boldsymbol{K} = \boldsymbol{W}_K\boldsymbol{H}^*$ and $\boldsymbol{V} = \boldsymbol{W}_V\boldsymbol{H}^*$, where $\boldsymbol{Q} = \{\boldsymbol{Q}_t\}_{t=1}^T \in \mathbb{R}^{D_a \times T}$, $\boldsymbol{K} = \{\boldsymbol{K}_t\}_{t=1}^T \in \mathbb{R}^{D_a \times T}$ and $\boldsymbol{V} = \{\boldsymbol{V}_t\}_{t=1}^T \in \mathbb{R}^{D_a \times T}$ are known as Queries, Keys and Values vectors, respectively and D_a represents the attention feature size, and $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{D_a \times D}$ are the corresponding learnable parameters. $\boldsymbol{K}$ is employed to learn the distribution of attention matrix on condition of the query matrix $\mathbf{Q}$, and $\boldsymbol{V}$ is used to exploit information representation. Thus the scaled dot-product attention $\boldsymbol{A}$ is defined as:

$$\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q}) = \frac{\boldsymbol{K}^T\boldsymbol{Q}}{\sqrt{D_a}} \ , \tag{3.1}$$

$$\boldsymbol{A} = \mathcal{F}_{Softmax}(\boldsymbol{K}, \boldsymbol{Q}) = \frac{\exp(\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q}))}{\sum_{t=1}^T \exp\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q})} \ , \tag{3.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{T \times T}$ and we consider $\boldsymbol{A}$ as the distribution of attention matrix on condition of the query matrix $\mathbf{Q}$. In Eq.3.1, due to the large degree of high dimensional $\boldsymbol{K}^T\boldsymbol{Q}$, scaling factor $\frac{1}{\sqrt{D_a}}$ is used to prevent the potential small gradient suffered by softmax. The output of attention network is:

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{A}. \tag{3.3}$$

After applying the attention module to both channels, We concatenate their outputs and feed into a score layer, which consists of multiple fully-connected layers ended with a sigmoid function. The score layer predicts the importance score $\hat{s}$ is sampled as:

$$\hat{\boldsymbol{S}} = \mathcal{F}_{Score}(\mathcal{F}_{Concat}(\boldsymbol{Z}_v, \boldsymbol{Z}_s)), \tag{3.4}$$

where $\mathcal{F}_{Score}$ denotes the score layer and $\mathcal{F}_{Concat}$ in this chapter means concatenation operation on different channels.

Overall Objective Function. We intend to treat the outputs as the importance scores of the whole video frames in this work. Thus, we simply employ the mean square loss $\mathcal{L}$ between the ground truth importance scores and the predicted

importance scores.

$$\mathcal{L} = \frac{1}{T}\sum_{i=1}^{T}(s_i - \hat{s}_i)^2, \tag{3.5}$$

3.3.2 Architecture Design Justification

The design of our architecture for video summarization capitalizes on the intrinsic dual nature of video data: visual content and sequential ordering.

Dual-representation Learning: In the realm of video summarization, capturing intricate visual patterns and the temporal dynamics of content is paramount. Our *visual channel*, empowered by Deep Convolutional Neural Networks (CNNs), is adept at extracting complex patterns and high-level features from visual data. By leveraging deep CNNs, we ensure that the salient visual intricacies of each frame are meticulously represented by $\boldsymbol{H}_v$. Concurrently, the *sequential channel*, implemented using a Bidirectional LSTM network, captures the inherent temporal dynamics of video content. The bidirectional nature of the LSTM ensures a rich representation informed by both preceding and succeeding frames, encapsulated by $\boldsymbol{H}_s$. The fusion of these representations into a dual-channel feature is a strategic move to bridge frame-wise attention to the overarching narrative of the video. This synergistic approach adeptly simulates human attention, marrying visual cues with the unfolding storyline.

The Attention Module: The revolutionizing potential of attention mechanisms in emphasizing specific segments of input data is harnessed in our architecture to underscore the pivotal frames or sequences. By refining each feature in the video, the network reinterprets it using a weighted combination of its entire neighborhood, mirroring the human propensity to derive meaning contextually. Central to the attention dynamics are the Queries, Keys, and Values. These elements play a pivotal role in the dynamic weighting of frames. The transformation of the initial feature set to $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, coupled with the subsequent attention computations, affords the model the discernment to identify focal points in videos. Culminating the attention process is the generation of the network output, $\boldsymbol{Z}$, which delineates the weighted importance of frames. This output is then channeled into a score layer, translating the attention outcomes into quantifiable metrics.

3.3.3 The Softmax Bottleneck

Almost all existing attention models follow the original pipeline from NLP tasks using the softmax function Eq. equation 3.2 to compute the attention. However, this section identifies the key limitation of softmax function for video summarization. It can be considered that the attention distribution is a finite set of pairs of a context and its conditional distribution $\mathcal{V} = \{(c_1, P^*(X|c_1)), \ldots, (c_1, P^*(X|c_T))\}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denotes T compatible keys in the video $\mathcal{V}$ and $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ denotes the contexts. It is assumed $P^* > 0$ and $\boldsymbol{A}^*$ represents the true attention distribution. Thus the true log distribution of attention in equation 3.2 can be re-formulated as:

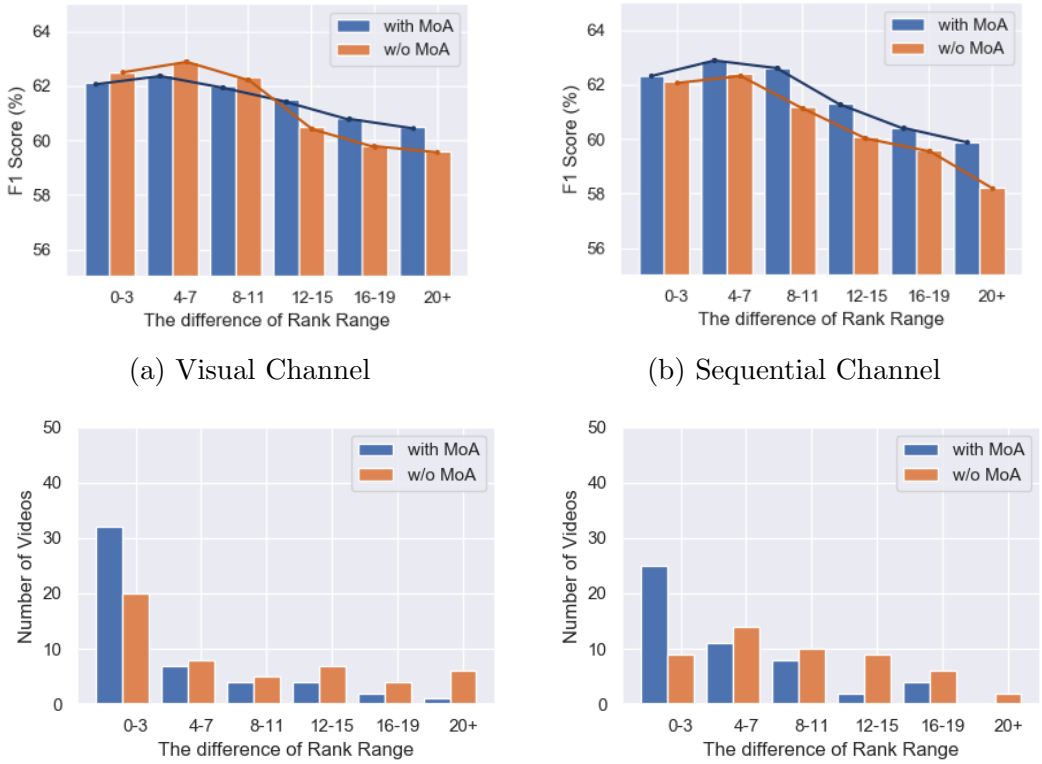
$$\boldsymbol{A}^* = \begin{bmatrix} logP^*(x_1|c_1) & logP^*(x_2|c_1) & \cdots & logP^*(x_T|c_1) \\ logP^*(x_1|c_2) & logP^*(x_2|c_2) & \cdots & logP^*(x_T|c_2) \\ \vdots & \vdots & \ddots & \vdots \\ logP^*(x_1|c_T) & logP^*(x_2|c_T) & \cdots & logP^*(x_T|c_T). \end{bmatrix} \tag{3.6}$$

The objective of attention model is to learn the conditional attention distribution $P_\theta(\mathcal{X}|\mathcal{C})$ parameterized by θ to match the true attention distribution $P^*(\mathcal{X}|\mathcal{C})$. It can be seen that the attention distribution problem is now turned into a **matrix factorization problem**. Since $\boldsymbol{A}$ is a matrix with size $N \times N$, the rank of learned attention distribution $\boldsymbol{A}$ is upper bounded by the embedding size d. If $d < rank(\boldsymbol{A}^*) - 1$, for any model parameter θ, there exists a context c in $\mathcal{V}$ such that $P_\theta(\mathcal{X}|\mathcal{C}) \neq P^*(\mathcal{X}|\mathcal{C})$. This is so called ***Softmax Bottleneck*** [160] which reflects the circumstance when softmax function does not have the capacity to express the true attention distribution when d is smaller than $rank(\boldsymbol{A}^*) - 1$.

The softmax bottleneck problem arises when trying to encapsulate intricate patterns or representations, especially in scenarios where the data, like video sequences, exhibits high complexity. To unpack this in the realm of video summarization: Consider a video, which is essentially a sequence of frames that encapsulate various visual elements and changes. When these visual contents are intricate and transitions between consecutive frames are marked by notable differences, the resulting

log probability matrix, denoted as $\boldsymbol{A}$, assumes a high rank. This high rank indicates a rich, diverse set of features and relationships between them. Take the example of a video depicting a cooking process compared to another that simply showcases someone eating. The act of cooking, with its myriad steps, ingredients, and repetitive actions, invariably contains more complex visual data than the straightforward act of eating. Yet, from a human perspective, one might regard both actions cooking and eating as equally important in the context of a broader narrative, like a meal preparation. However, the computational representation doesn't always reflect this intuitive human judgement. Due to the intrinsic richness of the cooking action, its representation matrix can have a much higher rank than that of the eating action. This poses a challenge when using the softmax function for attention mechanisms in video summarization. The softmax function, by its design, normalizes and assigns probabilities to these actions. When confronted with such high-rank matrices, the softmax can inadvertently diminish certain features from the richer content (like cooking) in an attempt to maintain a consistent representation across the video. This phenomenon, termed the 'softmax bottleneck', suggests that despite the utility of the softmax function, it can sometimes struggle to capture the nuances of complex visual sequences, potentially leading to a loss of critical information or misrepresentation in the summarized output.

In figure 3.3 we empirically verify such a Softmax Bottleneck problem can degrade the performance severely. We choose the TVSum dataset and calculate the difference $\mathcal{D} = T - rank(\boldsymbol{A})$, where T denotes the video length. This is because video lengths are not consistent so we only consider the difference between the actual rank and the full rank T. Lower difference values indicate the attention layer, after softmax, can retain high rank with minimum redundancy. On the other hand, Higher difference values mean the attention matrix of the whole video is low-rank. It can be due to the input video is not complex, e.g. no movement and the background is monotonous. But for most of the cases, the low-rank attention matrix is often resulted by key information missing due to long videos with high complexities. The statistics are collected from attention matrices of both visual and sequential channels. Our key observations are summarised as follows.



(a) Visual Channel

(b) Sequential Channel

(c) Visual Channel

(d) Sequential Channel

Figure 3.3: Averaged F1-score (%) and Number of videos with respect to the rank difference $\mathcal{D}$ in TVSum dataset.Blue and Orange bars compare our MoA against traditional softmax.

1. From figure 3.3 (a) and (b), higher rank representations tend to achieve higher F1 score. But due to the softmax capacity, significant performance drops can be seen in visual (after range 8-11) and sequential channels (after range 4-7), which confirms the existence of bottleneck. In other words, the softmax function cannot retain high-rank information for long complex videos.

2. From the distribution of video numbers in figure 3.3 (c) and (d), many video representations fall out of high-rank range (0-7) after softmax. According to the last observation, these videos are prone to getting lower performances.

3. The softmax bottleneck problem is more severe on sequential attention, which indicates the changes between frames are the key missing information that results in the lower rank.

Motivated by the above insights and inspired by the work of Yang *et al.* [160],

we come up with a **Mixture of Attention layer** (MoA) to alleviate the softmax bottleneck issue. We propose the Associated Query $\hat{\boldsymbol{Q}} = tanh(\boldsymbol{W}^{\hat{Q}}\boldsymbol{Q})$, where $\boldsymbol{W}^{\hat{Q}}$ is the Associated Query parameter. The idea is to capture the second-order changes between queries so that both complex and simple contents can be represented in a more smoothed attention representation. The conditional attention distribution is defined as:

$$\begin{aligned} P(x|c) = & \sum_{t=1}^{T} \frac{\exp(\mathcal{F}_{Scale}(\boldsymbol{K}_{c,t}, \boldsymbol{Q}_{c,t}))}{\sum_{t=1}^{T} \exp \mathcal{F}_{Scale}(\boldsymbol{K}_{c,t}, \boldsymbol{Q}_{c,t})} \hat{\boldsymbol{A}}_{c,t} \ , \\ & s.t. \sum_{t=1}^{T} \hat{\boldsymbol{A}}_{c,t} = 1 \ , \end{aligned} \tag{3.7}$$

$$where \ \hat{\boldsymbol{A}} = \mathcal{F}_{Softmax}(\boldsymbol{K}, \hat{\boldsymbol{Q}}) \ , \tag{3.8}$$

In Eq.3.8, $\hat{\boldsymbol{A}} \in T \times T$ is the associated attention distribution. Thus, MoA formulates the conditional attention distribution as:

$$\boldsymbol{A}_{moa} = \boldsymbol{A}\hat{\boldsymbol{A}}^{T} \ , \tag{3.9}$$

where $\boldsymbol{A}_{moa} \in \mathbb{R}^{T \times T}$. In Eq.3.1, due to the large degree of high dimensional $\boldsymbol{K}^{T}\boldsymbol{Q}$, scaling factor $\frac{1}{\sqrt{D_a}}$ is used to prevent the potential small gradient suffered by softmax. As $\boldsymbol{A}_{moa}$ is a non-linear function of the attention distribution, $\boldsymbol{A}_{moa}$ can be arbitrarily higher rank than standard self-attention structure $\boldsymbol{A}$. Thus the output of the mixture of attention network $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{A}_{moa}$ now can break the bottleneck problem. In figure 3.3, after applying the MoA, we can see a large proportion of videos fall into the 0-3 high rank range compared that of traditional softmax. Also, videos especially with lower ranks ($\mathcal{D} > 11$) can be predicted with higher F1 scores. The performance of the sequential channel is boosted, which indicates that all of the previous softmax representations missed high rank information. The smoothed performance drop and increased number of high rank videos serve as strong evidence to manifest the Softmax Bottleneck has been resolved by proposed MoA.

Besides, the DMASum utilizes stacked mixture of attention networks, and in

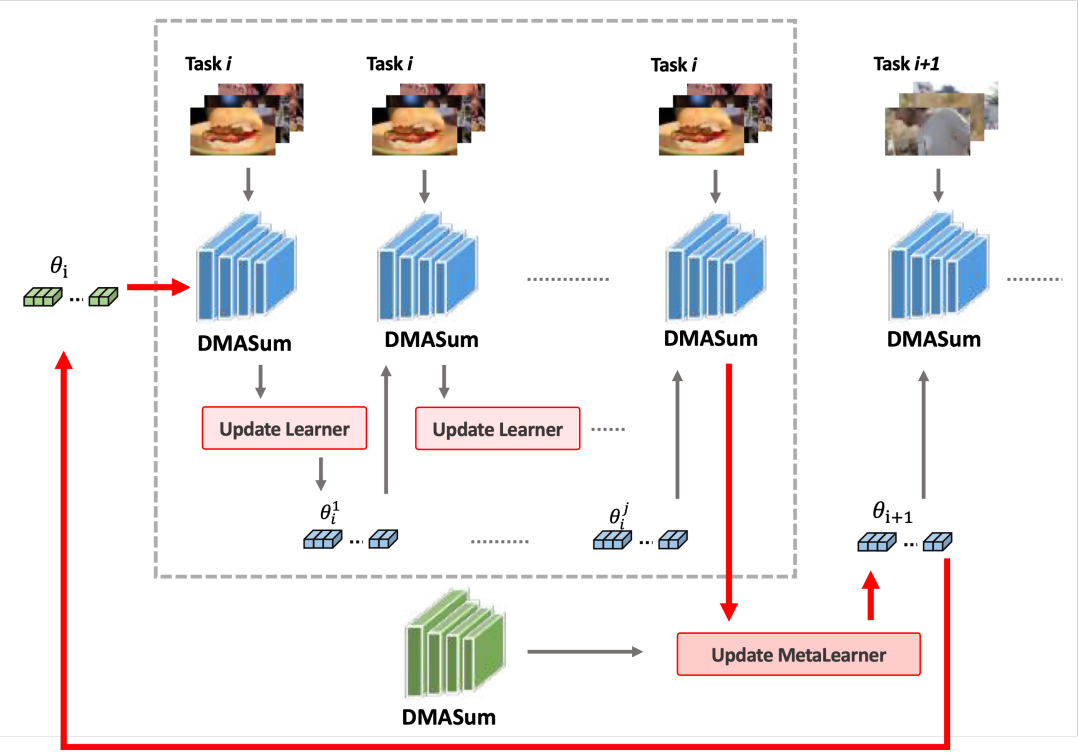


Figure 3.4: Overview of the i^{th} iteration for update θ_i to θ_{i+1}. There are two stages in this update process. The middle part shows the stage about how the Learner updates θ_i to θ_i^m by iterating m times. The outside parts shows the stage about how the Meta Learner updates θ_i to θ_{i+1}. The red line indicates mete learner parameters update process.

each layer we employ residual dropout connection [2] for allowing gradients to flow through a network directly and layer normalizaion [161] for normalizing the inputs across the features. Overall, the n^{th} layer output can be defined as:

$$\boldsymbol{Z}_n = \mathcal{F}_{Normalize}(\mathcal{F}_{Attention}(\boldsymbol{Z}_{n-1}) \oplus \boldsymbol{Z}_{n-1}), \tag{3.10}$$

where $\mathcal{F}_{Normalize}$ denotes as layer normalization, $\mathcal{F}_{attention}$ represents the attention layer and $\oplus$ represents the residual connection.

3.3.4 Single-Video Meta Learning

The key motivation to introduce Meta Learning is to improve the model generalization when the dataset of video summarization is small. Different from gradient descent, the *MetaLearner* is updated by weighted parameters of *Learner* in sub-

tasks, which can be formularized as:

$$Learner^* = MetaLearner(Learner(\tau_i)) \tag{3.11}$$

where τ_i denotes i^{th} video, *Learner* and $MetaLearner$ means the DMASum model in meta learning. We first employ the Model Agnostic Meta Learning (MAML) [156] due to its flexibility and superior performance but did not achieve expected results. Our observation is that in the video summarization context each video has its own latent mechanism that is not shared by different videos. Therefore, we propose a *Single-Video Meta Learning* rule to refrain the learner by only one video at each task. The process is shown as figure 3.4.

There are two stages of each epoch in this meta learning strategy. Firstly, to train the task τ_i, the *Learner* updates the parameter θ_i by traditional gradient descent. And, the *Learner* trains the task in a set number m recurrently to explore its latent summarizing context. The equation of updating parameter θ is:

$$\theta_i^j = \theta_i^{j-1} - \alpha \nabla \mathcal{L}_i^j(\mathcal{F}_{\theta_i^{j-1}}), \quad where j = 1 \dots m \tag{3.12}$$

where α denotes learning rate and ∇ denoted as the gradient, and $\mathcal{F}_\theta$ is the loss function on i^{th} task. After j^{th} iteration, the $MetaLearner$ updates the parameter θ_{i+1} by using the parameter θ_i^m of the *Learner* by:

$$\theta_i = \theta_{i-1} - \beta \nabla \mathcal{L}_i(\mathcal{F}_{\theta_i^m}), \tag{3.13}$$

where β is the learning rate of the *Learner*. θ_i updated state of *Learner* after the j^{th} iteration in $MetaLearner$. Overall, our meta learning is summarized in Algorithm 1. Note that in the last step of the algorithm, we treat θ_i^m - θ_i as a gradient and plug it into Adam instead of simply updating θ_i in the direction θ_i^m - θ_i.

Algorithm 1: Meta learning in DMASum

```
/* θ : Parameter of Learner;                          */
/* α : Learning rate in Learner;                      */
/* β : Learning rate in MetaLearner;                  */
/* n : The number of videos;                          */
/* m : Recurrent training Learner number;             */
/* F : the DMASum model;                              */
Initialize: θ
for k = 1 to epoch number do
    for i = 1 to n do
        Sample video i as task τ_i
        for j = 1 to m do
            θ_i^j = θ_i^{j-1} − α∇L_i^j(F_{θ_i^{j-1}})
        Update θ_{i+1} ← θ_i + β(θ_i^m - θ_i)
```

3.4 Experiments

3.4.1 Experiment Setup

Datasets. We evaluate our model on two datasets: SumMe [146] and TVSum [149]. SumMe consists of 25 videos covering a variety of events, such as sports and cooking. The duration of each video varies from 1 to 6.5 minutes. TVSum contains 50 videos downloaded from Youtube, which are selected from 10 categories. The video length varies from 1 to 10 minutes. Both datasets include ego-centric and third-person camera views, and the annotations were labeled by 25 human annotators. We also exploit two auxiliary datasets to augment the training data, where Open Video Project[1] (OVP) contains 50 videos and Youtube [162] contains 39 videos.

Rationale for Dataset Choice: The choice of SumMe and TVSum as our primary evaluation datasets is anchored in benchmarking considerations. These datasets have emerged as the de facto standards in the video summarization domain [114,114–116,120–122]. By using them, we ensure that our findings and advancements can be juxtaposed against a wide array of previous works, ensuring both comprehensibility and relevance of our results to the research community. Such a common ground paves the way for a more direct and transparent comparison. While SumMe and

[1]Open video project: https://open-video.org

TVSum are employed in many research studies for video summarisation, they are notably smaller datasets compared to others used in broader video understanding tasks, such as MSR-VTT [163]. Given their limited size, there's a legitimate concern regarding potential model overfitting on such constrained datasets.

Acknowledgment of Other Datasets: While SumMe and TVSum are our primary choices, the landscape of video summarization datasets is rich and diverse. Notable among them is the MED Summaries dataset [164], crafted explicitly for the evaluation of dynamic video summaries. This dataset encompasses annotations for 160 videos, bifurcated into a validation set of 60 videos and a test set of 100 videos, with the latter spanning 10 event categories. Another pertinent dataset is the Univ. of Texas at Austin Egocentric (UT Ego) Dataset [114], containing 4 videos garnered from head-mounted cameras. These videos, which stretch between 3-5 hours, provide authentic, unfiltered footage from natural settings. Such datasets, though not central to our current analysis, exemplify the ever-evolving resources in video summarization research and could serve as potential grounds for future evaluations.

Evaluation Metrics. We follow the commonly used protocol from [120] and converted the importance scores to shot-based summaries for both datasets, and the user annotations are changed from frame-level scores to key-shots scores using the kernel temporal segmentation (KTS) [165] method, which can temporally segment a video into disjoint intervals. We then compute the harmonic mean F-score as the evaluation metric. In addition, according to the recent evaluation protocol [126], we apply Kendall's τ [166] and Spearman's ρ [167] correlation coefficients for comparing the ordinal association between generated summaries and the ground truth (i.e. the relationship between rankings). Also, they provided correlation curves to visualize the predicted importance score ranking with respect to the reference annotations, i.e., when the predicted importance scores are perfectly concordant with averaged human-annotated scores, the curve lies on the upper bound of the light-blue area. Otherwise, the curve coincides with the lower bound of the area when the ranking of the scores is in reverse order of the reference.

Evaluation Settings. Following [120], we conducted the experiments under three settings. (1) Canonical (C): we used the standard 5-fold cross-validation (5FCV)

for SumMe and TVSum datasets. (2) Augmented (A): we used OVP and YouTube datasets to augment the training data in each fold under the 5FCV setting. (3) Transfer (T): we set a target testing dataset, e.g., SumMe or TVSum, and used the other three as the training data.

Implementation details. To be consistent with existing methods, the 1024 dimensional visual features extracted from the *pool*5 layer of the GoogLeNet [3] are used for training. To extract the temporal features, we design a Bi-LSTM model in the proposed network, as a two-layer LSTM with 512 hidden units per layer. For each attention layer, we set the attention dimension as 1024. We stack four attention layers for visual feature attention pipeline, and two layers for the sequential feature attention pipeline. The score layer consists of two fully-connected layers with 1024 hidden units. For Single-video Meta Learning, we set the learning rate of *Learner* as 3×10^{-5} and the learning rate of $MetaLearner$ as 6×10^{-5}. Moreover, the recurrent training Learner number is set as 3 and 5 in SumMe and TVSum datasets respectively. During the test, we follow the strategy of prior work [120, 122, 123] to generate the summary. In addition, we employ the ADAM optimizer to train our network and the hyperparameters are optimized via cross-validation.

3.4.2 Quantitative Evaluation

We first compare our method with state-of-the-art supervised approaches in three evaluation settings. Then, we re-implement the VS-LSTM, SUM-GAN, and DR-DSN models, and quote results for other methods from [125, 126, 147, 148, 154, 155, 159]. An in-depth ablation study is then provided to better understand of our DMASum.

Comparison with State-of-the-art Methods. Table Table 3.1 offers a comprehensive juxtaposition of our method, DMASum, against prevailing techniques in the video summarization arena. The evaluated methodologies predominantly align with categories like LSTM, GAN, Attention, and meta-learning models. Among these, M-AVS [154] and ACGAN [155] are rooted in attention models, while MetaL-TDVS [159] revolves around meta-learning. DMASum's standout performance across both datasets is immediately apparent from the results. Significantly, the enhanced F1-

Table 3.1: F1-score (%) of DMASum with state-of-the-art approaches on both SumMe and TVSum dataset.

Method	SumMe	TVSum
DPP-LSTM [120]	38.6	54.7
SASUM [125]	45.3	58.2
SUM-GAN [122]	41.7	54.3
Cycle-SUM [147]	41.9	57.6
DR-DSN [123]	42.1	58.1
MetaL-TDVS [159]	44.1	58.2
ACGAN [155]	46.0	58.5
CSNet [148]	51.3	58.8
M-AVS [154]	44.4	61.0
DMASum	**54.3**	**61.4**

Table 3.2: Rank-order correlation coefficients computed between predicted importance scores by different models and human-annotated scores on both SumMe and TVSum datasets using Kendall's τ and Spearman's ρ correlation coefficients.

Method	SumMe		TVSum	
	τ	ρ	τ	ρ
Random	0.000	0.000	0.000	0.000
DPP-LSTM [120]	-	-	0.042	0.055
SUM-GAN [122]	0.049	0.066	0.024	0.031
DR-DSN [123]	0.028	-0.027	0.020	0.026
Human	**0.227**	**0.239**	0.178	0.205
DMASum	0.063	0.089	**0.203**	**0.267**

score results assert that our unique approach, which addresses the softmax bottlenecka challenge that other attention-based models often overlookby synergizing attention mechanisms with meta-learning, yields more nuanced and precise importance score predictions.

To delve deeper into DMASum's effectiveness, we employed the innovative rank-order statistics [126]. This metric offers a more granulated evaluation, considering not just the significance of frames in isolation, but also their interdependencies and the consensus of annotators. The limitations of the F1-score become evident when faced with varied segment lengths, as is the case in scenarios like two-peak, KTS, and randomized KTS. In contrast, correlation coefficients, specifically Kendall's τ and Spearman's ρ, proficiently measure the alignment between the importance scores produced by machine annotation and those derived from human judgment.

As observed in Table 3.2, DMASum's coefficients not only outpace those of other premier models but also, in the context of the TVSum dataset (0.233 and 0.267), surpass scores from human annotators (0.205 and 0.267). This exemplary performance can be traced back to our dual-channel attention mechanism, which adeptly replicates human perceptual processes, capturing both visual and sequential content nuances. Moreover, our integration of meta-learning empowers DMASum to discern and adapt to the intrinsic narrative structures within videos. It's essential to recognize the inherent variability among human annotators; when presented with identical video content, different annotators may emphasize disparate elements. However, DMASum, by synthesizing feedback from a myriad of annotators, produces a more balanced, consistent, and thus superior attention-based model for video summarization.

For a visual corroboration of these insights, figure 3.5 showcases two representative correlation coefficients. We can see that the considerable variance among human annotators underscores the inherent subjectiveness in their annotations, which also indicates the resson why the scores derived from human annotators of TVSum is lower than our proposed framework. Curves that ascend above the black dashed line, indicative of random importance scores, signal higher consistency. Here again, DMASum doesn't merely parallel the mean human annotator performance but no-

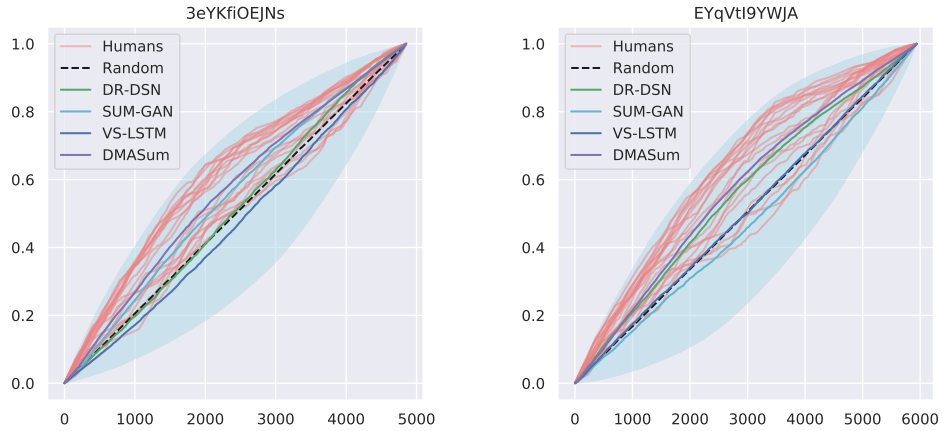


Figure 3.5: Example correlation curves produced for two videos from the TVSum dataset (3eYKfiOEJNs and EYqVtl9YWJA are video ids). The red lines represent correlation curves for 25 human annotators and the black dashed line is the expectation for a random importance score. The magenta curve shows the corresponding result.

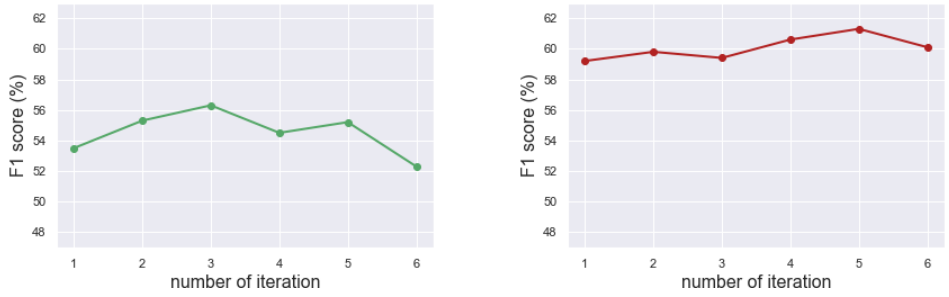
tably outstrips most of its contemporaries.

3.4.3 Ablation study.

The success of our DMASum ascribes to both the framework design and technical improvement in each module. To analyze the effect of each component in DMASum, we conduct six ablation study models including DMASum without meta learning (DMASum$_{wom}$), DMASum with standard softmax function in self-attention network

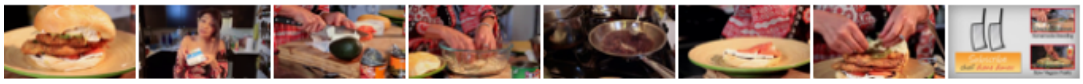
Table 3.3: F1-score (%) of ablation study on SumMe and TVSum datasets. There are five ablation models: DMASum$_{wom}$ (without meta learning strategy), DMASum$_{softmax}$ (with standard softmax function in self-attention network), DMASum$_{v}$ (without sequential channel), DMASum$_{s}$ (without visual channel), DMASum$_{b}$ (with multiple videos in a batch), and DMASum$_{maml}$ (with MAML)

Method	SumMe	TVSum
DMASum$_{wom}$	51.6	60.6
DMASum$_{softmax}$	50.6	60.1
DMASum$_{v}$	53.2	60.5
DMASum$_{s}$	53.3	61.0
DMASum$_{b}$	51.3	60.0
DMASum$_{maml}$	49.3	59.2
DMASum	**54.3**	**61.4**

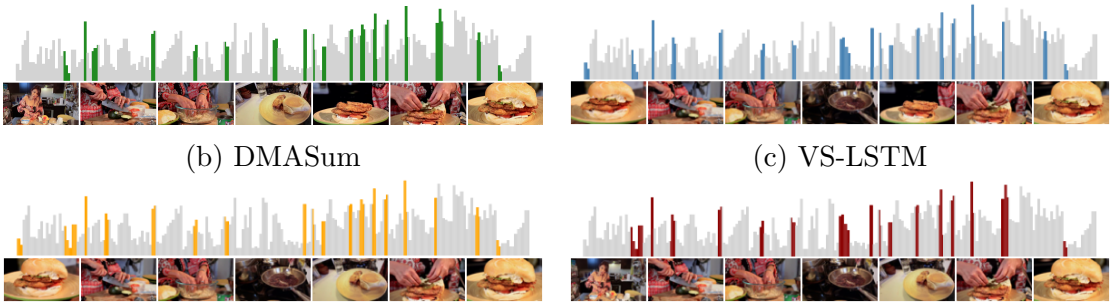


(a) SumMe Dataset (b) TVSum Dataset

Figure 3.6: Different recurrent training Learner number with respect to the F1-score (%) in DMASum on both SumMe and TVSum datasets.



(a) Example frames from video 37 in TVSum (indexed by [149])



(b) DMASum (c) VS-LSTM

(d) SUM-GAN (e) DR-DSN

Figure 3.7: Quantitative results of different approaches for video 16 in TVSum. In (b) to (e), the light-gray bars represent the ground truth importance scores, and the colored bars correspond to the selected frames by different methods.

(DMASum$_{softmax}$), DMASum without sequential channel (DMASum$_v$), DMASum without visual channel (DMASum$_s$), DMASum$_b$ is developed with the batch version of Reptile, and DMASum is designed with MAML (DMASum$_{maml}$). Results are summarised in Table 3.3, from which we can understand the following questions.

The Effectiveness of Self-attention Architecture. The inception of self-attention architecture has ushered in a paradigm shift in video summarization, presenting a tangible enhancement in performance. By scrutinizing the efficacy of attention mechanisms, we unveil their pivotal role in discerning intricate temporal and spatial interdependencies within video frames, a challenge often encountered in video summarization tasks. To isolate the impact of our self-attention approach, we tem-

porarily set aside our meta-learning module, facilitating a direct comparative analysis against benchmarks such as DPP-LSTM [120] and M-AVS [154]. The latter models employ either no attention or conventional attention frameworks. In this face-off, our self-attention model underscores its prowess, registering an average performance surge ranging between 5% to 10%. However, it's imperative to acknowledge that M-AVS holds a marginal advantage over our method on the TVSum dataset, a distinction attributed to its integrated autoencoder architecture. This exercise underscores a key insight: attention-based models, particularly self-attention, are adept at capturing and representing the dynamic evolution of visual content in videos. Such models can decipher intricate scene transitions, varying actions, and nuanced content shifts, which are fundamental aspects of video summarization. By giving weightage to salient frames and sequences, attention mechanisms pave the way for summaries that resonate more with the human perception of video narratives.

The Softmax Bottleneck problem results in severe performance gaps. By replace the MoA back to traditional softmax function, the performance drops 3.7% and 1.3% respectively on the two datasets. A more detailed analysis has been discussed in Section 3, from where we can see the problem is more critical when video contents are long and complex, involving rich sequential information.

Visual vs Sequential Representation. By comparing the performance of DMASum$_v$ or DMASum$_s$, we can observe that: 1) In TVSum dataset, the DMASum$_v$ gained a slightly better performance than DMASum$_s$. 2) The performance in SumMe dataset benefits more from the sequential channel. The self-attention network can effectively connect visual features from frames and the sequential information for the whole story line and thus our combined DMASum achieves better results.

The Necessity of Meta Learning. Removing the meta learning can heavily affect the performance by 2.7% on SumMe dataset. The key reason is that SumMe is a relatively small dataset. This observation serves as strong evidence to validate the motivation and necessity of our meta learning module.

MAML, Batch, and Single Video Meta Learning. The Single Video rule is the key finding that distinguish it from meta learning in other applications, e.g. few-

shot learning. This is due to a video itself is rich and complex. By increasing each meta learning task from one video to three in a batch, the performance of DMASum$_b$ drops 3% and 1.4% with the clearly slowed training process. In addition, we can see that the performance of our proposed meta learning strategy is better than the batch version of the Reptile strategy, and the batch version of the Reptile strategy is time-consuming during the training process. The efficiency of Single-Video rule is also validated by comparing it to DMASum$_{maml}$.

Number of Recurrent Learning. In a controlled experiment, we observe that when the recurrent training Learner number is 3 for SumMe Dataset and 5 for the TVSum dataset, the F-score reaches the highest shown from figure 3.6. Which means, the Learner might not learn the summarizing mechanism when the number is too low, and when the number is too high, the Learner might overfit the current video. In this chapter, the number of recurrent training is automatically chosen by using the standard 5-fold cross validation.

Comparison under Different Settings. Another approach to examining the model generalization is to investigate its performance under different task settings. Table 3.4 shows the experimental results of the comparison between the DMASum and cited results of state-of-the-art approaches in canonical, augmented and transfer settings. Note that even though the performance of our model in augmented and transfer settings are partially better than the best results. We observe that the given importance scores in Youtube and OVP datasets are either 0 or 1. However, the DMASum is learning by the importance scores within the range of zero to one from SumMe and TVSum datasets. Such discrepancy of importance score format in both Youtube and OVP datasets would cause the meta learning strategy to be ineffective or even counterproductive because our model is not tailored to handle the discrepancy in labels. Thus in the future, we can improve our framework to adapt to this situation. But on the positive side, our DMASum is still capable in both augmented and transfer settings and achieves comparable results to that of state-of-the-art models despite the above difficulties.

Table 3.4: F-score (%) of approaches in canonical, augmented and transfer settings on SumMe and TVSum datasets.

Method	SumMe			TVSum		
	C	A	T	C	A	T
DPP-LSTM [120]	38.6	42.9	40.7	54.7	59.6	58.7
SUM-GAN [122]	41.7	43.6	-	54.3	**61.2**	-
DR-DSN [123]	42.1	43.9	42.6	58.1	59.8	58.9
CSNet [148]	51.3	52.1	45.1	58.8	59.0	59.2
DMASum	**54.3**	**54.1**	**52.2**	**61.4**	**61.2**	**60.5**

3.4.4 Qualitative Evaluation

To better illustrate the important frames selection of different approaches, we provide qualitative results for an exemplary video in figure 3.7, which tells a story of how to cook a burger. Overall, we can observe that all summaries generated by the different models can cover the intervals with high importance scores. Moreover, according to the figure, the summaries produced by both our DMASum and SUM-GAN contain more peaks, which proves that our proposed model can effectively capture key-frames from the original video. Also, the summary of our model is more sparse and much closer to the entire storyline, i.e., the different cooking stages, which means our meta learning strategy can learn the latent mechanism of summarizing a video.

3.5 Limitation and Discussion

While our approach addresses the softmax bottleneck in video summarization through rank measures, certain limitations emerge. Notably, rank only captures relative ordering, ignoring the absolute magnitude of dataa crucial aspect in discerning video segment importance. Rank can sometimes misrepresent the actual data distribution, potentially obscuring subtle video nuances. Additionally, rank ties can arise from video frames with similar importance scores, introducing ambiguity. Furthermore, the computational overhead of rank structures can increase with larger video datasets. Lastly, adapting rank-based measures to neural architectures reliant on gradient-based optimization can add complexity. Thus, while valuable, rank mea-

sures have their constraints, prompting a call for hybrid strategies in future research to comprehensively address the softmax bottleneck.

3.6 Conclusion

We have presented the first work to introduce self-attention meta learning architecture to estimate the visual and sequential attentions jointly for video summarization. The self-attention formula was derived into a matrix factorization problem and key technical Softmax Bottleneck has been identified with both theoretical and empirical evidences. Our work also confirmed the importance of high-rank representation for video summarization tasks. A novel MoA module was proposed to replace the softmax, which can compare twice by query-key and self-query attentions. The Single-Video Meta Learning rule was designed and particularly tailored for video summarization tasks and significantly improved off-the-shelf Meta Learning, e.g. MAML. On two public datasets, our DMASum outperforms other methods in terms of both F1-score and achieved human-level performance using rank-order correlation coefficients. However, the datasets' limited size raises valid concerns about potential overfitting. Furthermore, inherent subjectiveness among human annotators can also influence results from human annotators. Future work could focus on further improve the generalisation for cross-dataset settings using an integrated framework.

CHAPTER 4

Action Quality Assessment with Temporal Parsing Transformer

Chapter 3 of this thesis focused on the challenge of aligning machine interpretations of video content with collective human perspectives, aiming for a synthesis of human consensus. This was achieved through the emblematic task of video summarization, with the objective being the extraction of salient and representative frames that echo human insights.

As we move to Chapter 4, the emphasis shifts from a broad human consensus to a more specific, rule-based understanding of actions, which is an essential facet of Interaction-level action understanding. The domain of Action Quality Assessment (AQA) represents this shift, where the challenge intensifies to discern subtle visual differences and evaluate actions based on domain-specific rules. This precise understanding of actions is crucial in many real-world applications like sports judging or medical procedures. Instead of just aligning with broad human consensus as in video summarization, AQA requires machines to fine-tune their understanding to granular levels, often dictated by specific human rules.

Existing state-of-the-art AQA methods typically rely on the holistic video representations for score regression or ranking, which limits the generalization to capture fine-grained intra-class variation. To overcome the above limitation, we propose a

temporal parsing transformer to decompose the holistic feature into temporal part-level representations. Specifically, we utilize a set of learnable queries to represent the atomic temporal patterns for a specific action. Our decoding process converts the frame representations to a fixed number of temporally ordered part representations. To obtain the quality score, we adopt the state-of-the-art contrastive regression based on the part representations. Since existing AQA datasets do not provide temporal part-level labels or partitions, we propose two novel loss functions on the cross attention responses of the decoder: a ranking loss to ensure the learnable queries to satisfy the temporal order in cross attention and a sparsity loss to encourage the part representations to be more discriminative. Extensive experiments show that our proposed method outperforms prior work on three public AQA benchmarks by a considerable margin.

4.1 Introduction

Action quality assessment(AQA), which aims to evaluate how well a specific action is performed, has attracted increasing attention in research community recently [130, 131]. In particular, assessing the action quality accurately has great potential in a wide range of applications such as health care [127] and sports analysis [131, 134, 135, 168]. To be more specific, AQA delves deeper by gauging how well a given action is executed. This evaluation is grounded in the subtle visual differences that can be observed when the same generic action (e.g., a jump, a throw, a surgical stitch) is performed with varying degrees of proficiency, expertise, or finesse. The goal of AQA is to generate a quality score or ranking that reflects these nuanced visual distinctions, making it particularly valuable in scenarios where precision and excellence of action execution are paramount, such as in sports, medical procedures, and performance arts.

In contrast to the conventional action recognition tasks [25, 26], AQA poses unique challenges due to the subtle visual differences. Previous works on AQA either use ranking-based pairwise comparison between test videos [128] or estimate the quality score with regression-based methods [135, 137]. However, these methods

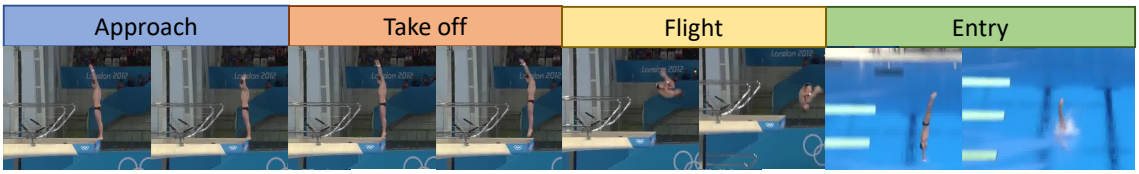


Figure 4.1: An action consists of multiple temporally ordered key phases.

typically represent a video with its *holistic representation*, via the global pooling operation over the output of the backbone network(e.g., I3D [26]). Since the videos to be evaluated usually are from the same coarse action category (e.g., diving) in AQA, it's crucial to capture *fine-grained intra-class variation* to estimate more accurate quality scores. Thus, we propose to decompose the holistic feature into more fine-grained temporal part-level representations for AQA.

To achieve this, a promising strategy is to represent the video by using a set of atomic action patterns. For example, a diving action consists of several key phases, such as *approach*, *take off*, *flight*, etc., as illustrated in Fig.4.1. The fine-grained patterns enable the model to describe the subtle differences, which is expected to improve the assessment of action quality effectively. Nevertheless, it remains challenging to learn such atomic patterns as the existing AQA datasets do not provide temporal part-level labels or partitions. An "atomic pattern" refers to the fundamental, indivisible components or units that together constitute a complex behavior or activity. In the context of action analysis, atomic patterns can be seen as the essential building blocks or sub-actions that, when sequenced or combined in various ways, form a complete, recognizable action. These patterns capture the granular details of an action, allowing for a more precise understanding and representation of the action's intricacies. For example, in the analysis of a diving action, atomic patterns might include the initial "approach" on the diving board, the "take-off" leap, the "flight" or mid-air maneuvers, and the eventual "entry" into the water. Each of these components is an atomic pattern, and their collective execution and interplay determine the overall quality and style of the dive.

In this work, we aim to tackle the aforementioned limitations by developing a regression-based action quality assessment strategy, which enables us to leverage the fine-grained atomic action patterns without any explicit part-level supervision. Our

key idea is to model the shared atomic temporal patterns, with a set of learnable queries for a specific action category. Similar to the decoding process of transformer applied in natural language modeling [169], we propose a temporal parsing transformer to decode each video into a fixed number of part representations. To obtain quality scores, we adopt the recent state-of-the-art contrastive regression framework [1]. Our decoding mechanism allows the part representations between test video and exemplar video to be implicitly aligned via a shared learnable query. Then, we generate a relative pairwise representation per part and fuse different parts together to perform the final relative score regression.

To learn the atomic action patterns without the part-level labels, we propose two novel loss functions on the cross attention responses of the decoder. Specifically, to ensure the learnable queries satisfy the temporal order in cross attention, we calculate an attention center for each query by weighted summation of the attention responses with their temporal clip orders. Then we adopt a marginal ranking loss on the attention centers to guide the temporal order. Moreover, we propose a sparsity loss for each query's attention distribution to guide the part representations to be more discriminative.

We evaluate our method, named as temporal parsing transformer(TPT), on three public AQA benchmarks: MTL-AQA [131], AQA-7 [130] and JIGSAWS [170]. As a result, our method outperforms previous state-of-the-art methods by a considerable margin. The visualization results show that our method is able to extract part-level representations with interpretable semantic meanings. We also provide abundant ablation studies for better understanding.

The main contributions of this chapter are three folds:

- We propose a novel temporal parsing transformer to extract fine-grained temporal part-level representations with interpretable semantic meanings, which are optimized with the contrastive regression framework.

- We propose two novel loss functions on the transformer cross attentions to learn the part representations without the part-level labels.

- We achieve the new state-of-the-art on three public AQA benchmarks, namely

MTL-AQA, AQA-7 and JIGSAWS.

4.2 Related Work

Fine-grained action parsing is also studied in the field of action segmentation or temporal parsing [171–176]. For example, Zhang et al. [177] proposed Temporal Query Network adopted query-response functionality that allows the query to attend to relevant segments. Dian et al. [178] proposed a temporal parsing method called TransParser that is capable of mining sub-actions from training data without knowing their labels. However, different from the above fields, part-level labels are not available in AQA task. Furthermore, most of the above methods focus more on frame-level feature enhancement, whereas our proposed method extracts part representations with interpretable semantic meanings.

4.3 Method

In this section, we introduce our temporal parsing transformer with the contrastive regression framework in detail.

4.3.1 Overview

The input of our network is an action video. We adopt the Inflated 3D ConvNets(I3D) [26] as our backbone, which first applies a sliding window to split the video into T overlapping clips, where each clip contains M consecutive frames. Then, each clip goes through the I3D network, resulting in time series clip level representations $\boldsymbol{V} = \{\boldsymbol{v}_t \in \mathbb{R}^D\}_{t=1}^T$, where D is feature dimension and T is the total number of clips. In our work, we do not explore spatial patterns, hence each clip representation $\boldsymbol{v}_t$ is obtained by average pooling across spatial dimensions. The goal of AQA is to estimate a quality score $\mathbf{s}$ based on the resulting clips representation $\boldsymbol{V}$. In contrastive regression framework, instead of designing a network to directly estimate raw score $\mathbf{s}$, it estimates a relative score between the test video and an exemplar video $\boldsymbol{V}_0$ with known quality score $\mathbf{s}_0$, which is usually sampled from training set.

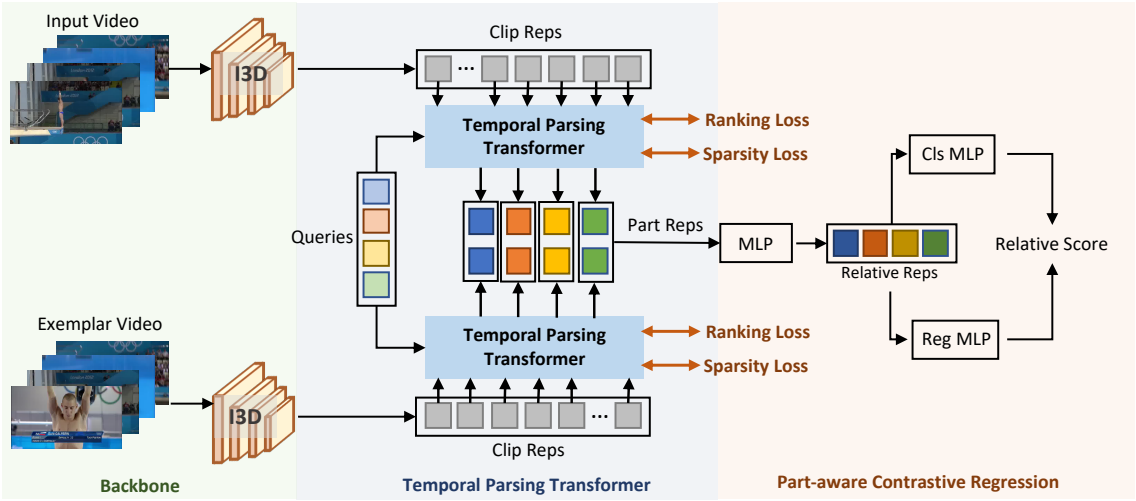


Figure 4.2: Overview of our framework. Our temporal parsing transformer converts the clip-level representations into temporal part-level representations. Then the part-aware contrastive regressor first computes part-wise relative representations and then fuses them to estimate the relative score. We adopt the group-aware regression strategy, following [1]. During training, we adopt the ranking loss and sparsity loss on the decoder cross attention maps to guide the part representation learning.

Then, contrastive regression aims to design a network $\mathcal{F}$ that estimates the relative score $\Delta \boldsymbol{s}$:

$$\Delta \mathbf{s} = \mathcal{F}(\boldsymbol{V}, \boldsymbol{V}_0), \tag{4.1}$$

then final score can be obtained by

$$\mathbf{s} = \mathbf{s}_0 + \Delta \mathbf{s}. \tag{4.2}$$

In our framework, we first adopt a temporal parsing transformer $\mathcal{G}$ to convert the clip level representations $\boldsymbol{V}$ into temporal part level representations, denoted by $\boldsymbol{P} = \{\boldsymbol{p}_k \in \mathbb{R}^d\}_{k=1}^K$, where d is the part feature dimension and K is the number of queries, i.e. temporal atomic patterns. Then for test video and exemplar video, we can have two set of aligned part representations $\boldsymbol{P}$ and $\boldsymbol{P}_0 = \{\boldsymbol{p}_k^0 \in \mathbb{R}^d\}_{k=1}^K$. Our new formulation can be expressed as:

$$\Delta \mathbf{s} = \mathcal{R}(\boldsymbol{P}, \boldsymbol{P}_0). \tag{4.3}$$

where $\mathcal{R}$ is the relative score regressor, and

$$\boldsymbol{P} = \mathcal{G}(\boldsymbol{V}), \boldsymbol{P}_0 = \mathcal{G}(\boldsymbol{V}_0). \tag{4.4}$$

An overview of our framework is illustrated in Fig.4.2. Below we describe the detailed structure of temporal parsing transformer $\mathcal{G}$ and part-aware contrastive regressor $\mathcal{R}$.

4.3.2 Temporal parsing transformer

Our temporal parsing transformer takes the clip representations as memory and exploits a set of learnable queries to decode part representations. Different from prevalent DETR architecture [8], our transformer only consists of a decoder module. We found that the encoder module does not provide improvements in our framework; it even hurts the performance. We guess it might because that clip-level self-attention smooths the temporal representations, and our learning strategy cannot decode part presentations in this way without part labels.

Note that our hypothesis, which revolves around decoding a sequence of actions into multiple sub-actions, naturally aligns with the capabilities of the DETR. The Transformer's self-attention mechanism gives it the capability to weigh the significance of different sub-actions in relation to one another, providing a more interconnected and comprehensive understanding of the entire action sequence. In comparison, while deep CNNs might excel in feature extraction from individual frames and bidirectional LSTMs in predicting temporal sequences, neither can inherently break down an action sequence into its constituent parts as effectively as the Transformer architecture.

We perform slight modifications to the standard DETR decoder. That is, the cross attention block in our decoder has a learnable parameter, temperature, to control the amplification of the inner product. Formally, in the i-th decoder layer, the decoder part feature $\{p_k^{(i)} \in \mathbb{R}^d\}$ and learnable atomic patterns(i.e. query set) $\{q_k \in \mathbb{R}^d\}$ are first summed as a query and then perform cross attention on the

embedded clip representation $\{\mathbf{v}_t \in \mathbb{R}^d\}$:

$$\alpha_{k,t} = \frac{\exp^{(\boldsymbol{p}_k^{(i)}+q_k)^T \cdot \boldsymbol{v}_t/\tau}}{\sum_{j=1}^{T} \exp^{(\boldsymbol{p}_k^{(i)}+q_k)^T \cdot \boldsymbol{v}_j/\tau}}, \tag{4.5}$$

where $\alpha_{k,t}$ indicates the attention value for query k to clip t, $\tau \in \mathbb{R}$ indicates the learnable temperature to enhance the inner product to make the attentions more discriminative. Unlike DETR [8], in our decoder, we do not utilize position embedding of clip id to the memory $\{\boldsymbol{v}_t\}$. We expect our query to represent atomic patterns, instead of spatial anchors, as in the detection task [179, 180]. We found that adding position encoding significantly drops the performance and makes our learning strategy fail, which will be shown in the experiment section.

In our experiments, we only utilize one-head attention in our cross attention blocks. The attention values are normalized across different clips, since our goal is to aggregate clip representations into our part representation. Then the updated part representation $\boldsymbol{p}_k^{(i)'}$ has the following form:

$$\boldsymbol{p}_k^{(i)'} = \sum_{j=1}^{T} \alpha_{k,j} \boldsymbol{v}_j + \boldsymbol{p}_k^{(i)}. \tag{4.6}$$

We then perform standard FFN and multi-head self-attention on decoder part representations. Similar to DETR [8], our decoder also has a multi-layer structure.

4.3.3 Part-aware contrastive regression

Our temporal parsing transformer converts the clip representations $\{\boldsymbol{v}_t\}$ into part representations $\{\boldsymbol{p}_k\}$. Given a test video and exemplar video, we can obtain two part representation sets $\{\boldsymbol{p}_k\}$ and $\{\boldsymbol{p}_k^0\}$. One possible way to estimate the relative quality score is to fuse each video's part representations and estimate the relative score. However, since our temporal parsing transformer allows the extracted part representations to be semantically aligned with the query set, we can compute the relative pairwise representation per part and then fuse them together. Formally, we utilize a multi-layer perceptron(MLP) f_r to generate the relative pairwise represen-

tation $\boldsymbol{r}_k \in \mathbb{R}^d$ for k-th part:

$$\boldsymbol{r}_k = f_r(Concat([\boldsymbol{p}_k; \boldsymbol{p}_k^0])). \tag{4.7}$$

The MLP f_r is shared across different parts. To balance the score distributions across the whole score range, we adopt the group-aware regression strategy to perform relative score estimation [1]. Specifically, it first calculates B relative score intervals based on all possible pairs in training set, where each interval has equal number of pair-samples. Then it generates a one-hot classification label $\{l_n\}$, where l_n indicates whether the ground truth score $\Delta\mathbf{s}$ lies in n-th interval, and a regression target $\gamma_n = \frac{\Delta\mathbf{s} - x_{left}^n}{x_{right}^n - x_{left}^n}$, where x_{left}^n, x_{right}^n denote the left and right boundary of n-th interval. Readers can refer to [1] for more details.

We adopt average pooling[1] on the relative part representations $\{\boldsymbol{r}_k\}$ and then utilize two two-layer MLPs to estimate the classification label $\{l_n\}$ and regression target $\{\gamma_n\}$. Different from [1], we do not utilize tree structure. Since we have obtained fine-grained part-level representations and hence the regression becomes simpler, we found that two-layer MLP works fine.

4.3.4 Optimization

Since we do not have any part-level labels at hand, it's crucial to design proper loss functions to guide the part representation learning. We have assumed that each coarse action has a set of temporally ordered atomic patterns, which are encoded in our transformer queries. To ensure that our query extracts different part representations, we constrain the attention responses in cross attention blocks for different queries. Specifically, in each cross attention process, we have calculated the normalized attention responses $\{\alpha_{k,t}\}$ by Eq.4.5, then we compute an attention center $\bar{\alpha}_k$ for k-th query:

$$\bar{\alpha}_k = \sum_{t=1}^{T} t \cdot \alpha_{k,t}, \tag{4.8}$$

[1]We note that it might be better to weight parts. However, part weighting does not provide improvements during our practice. We guess that it may be due to the self-attention process in decoder that the relations of parts are already considered.

where T is the number of clips and $\sum_{t=1}^{T} \alpha_{k,t} = 1$. Then we adopt two loss functions on the attention centers: ranking loss and sparsity loss.

Ranking loss To encourage that each query attends to different temporal regions, we adopt a ranking loss on the attention centers. We wish our part representations have a consistent temporal order across different videos. To this end, we define an order on the query index and apply ranking losses to the corresponding attention centers. We exploit the margin ranking loss, which results in the following form:

$$L_{rank} = \sum_{k=1}^{K-1} \max(0, \bar{\alpha}_k - \bar{\alpha}_{k+1} + m) + \max(0, 1 - \bar{\alpha}_1 + m) + \max(0, \bar{\alpha}_K - T + m), \quad (4.9)$$

where m is the hyper-parameter margin controlling the penalty, the first term guides the attention centers of part k and $k+1$ to keep order: $\bar{\alpha}_k < \bar{\alpha}_{k+1}$. From Eq. 4.8, we have the range of attention centers: $1 \leq \bar{\alpha}_k \leq T$. To constrain the first and last part where $k = 1$ and $k = K$, we assume there is two virtual centers at boundaries: $\bar{\alpha}_0 = 1$ and $\bar{\alpha}_{K+1} = T$. The last two terms in Eq. 4.9 constrain the first and last attention centers not collapsed to boundaries. The weights of the three terms in Eq. 4.8 are the same.

Sparsity loss To encourage the part representations to be more discriminative, we further propose a sparsity loss on the attention responses. Specifically, for each query, we encourage the attention responses to focus on those clips around the center μ_k, resulting in the following form:

$$L_{sparsity} = \sum_{k=1}^{K} \sum_{t=1}^{T} |t - \bar{\alpha}_k| \cdot \alpha_{k,t} \quad (4.10)$$

During training, our ranking loss and sparsity loss are applied to the cross attention block in each decoder layer.

Overall training loss In addition to the above auxiliary losses for cross attention, our contrastive regressor $\mathcal{R}$ generates two predictions for the group classification label $\{l_n\}$ and regression target $\{\gamma_n\}$, we follow [1] to utilize the BCE loss on each

group and square error on the ground truth regression interval:

$$L_{cls} = -\sum_{n=1}^{N} l_n \log(\tilde{l}_n) + (1 - l_n) \log(1 - \tilde{l}_n) \tag{4.11}$$

$$L_{reg} = \sum_{n=1}^{N} \mathbb{1}(l_n = 1)(\gamma_n - \tilde{\gamma}_n)^2 \tag{4.12}$$

where L_{reg} only supervises on the ground truth interval, $\tilde{l}_n$ and $\tilde{\gamma}_n$ are predicted classification probability and regression value. The overall training loss is given by:

$$L_{all} = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{rank} \sum_{i=1}^{L} L^i_{rank} + \lambda_{sparsity} \sum_{i=1}^{L} L^i_{sparsity}, \tag{4.13}$$

where i indicates layer id and L is the number of decoder layers, λ_{cls}, λ_{reg}, λ_{rank}, $\lambda_{sparsity}$ are hyper-parameter loss weights. The weights of λ_{cls}, λ_{reg}, λ_{rank} are all 1, as λ_{reg}, λ_{rank} are the basis loss and λ_{rank} is import to train the decoder. Weights for $\lambda_{sparsity}$ is set to 0.05.

4.4 Experiment

4.4.1 Experimental Settup

Datasets We perform experiments on three public benchmarks: MTL-AQA [131], AQA-7 [130], and JIGSAWS [170].

Choice of Datasets: The underlying motivation of this chapter is to delve into the understanding of human actions, specifically through the lens of explicit human-defined rules. In this vein, the chosen datasets resonate profoundly with our intent. From varying sports genres to intricate medical surgeries, each dataset encapsulates actions that are stringently governed by well-defined rules. This is in stark contrast to datasets such as SumMe or TVSum, which we explored in the preceding chapter. The latter datasets predominantly feature natural videos, devoid of clear rules or structural patterns. The salient attribute of our selected datasets the intrinsic adherence to explicit rules is pivotal in facilitating a nuanced understanding of

human actions as per our research objective.

MTL-AQA dataset is the largest dataset for AQA task. In MTL-AQA, 1412 fine-grained samples are collected from 16 different events with different views. The dataset mainly focuses on diving covering various categories. In this dataset, different annotations are available to support research on different tasks, including action quality assessment, action recognition, and comment generation. In addition, raw annotation of score and difficulty (DD) from the multiple judges is available. We split the dataset into 1059 training samples and 353 test data following the evaluation protocol suggested in paper [131].

AQA-7 dataset contains samples from seven different action categories, including gymnastic vaulting, big air skiing, big air snowboarding, synchronous diving - 3m springboard, synchronous diving - 10m platform, and trampoline. Following the setting in [130], we excluded the trampoline category with much longer videos than the other categories, resulting in 803 training videos and 303 testing videos.

JIGSAWS [170] is a surgical activities dataset that contains three tasks, namely Suturing (S), Needle Passing (NP), and Knot Tying (KT). To evaluate different features of videos, samples in the dataset are annotated with multiple scores, and the final score is the sum of all annotations. We applied four-fold cross-validation following [139] to align with previous work.

Evaluation Metrics Following prior work [1], we utilize two metrics in our experiments, the Spearmans rank correlation and relative L2 distance(R-ℓ_2). **Spearman's rank correlation** was adopted as our main evaluation metric to measure the difference between true and predicted scores. The Spearmans rank correlation is dened as follows:

$$\rho = \frac{\sum_i (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_i (p_i - \bar{p})^2 \sum_i (q_i - \bar{q})^2}} \tag{4.14}$$

It focuses on the ranking of test samples. In contrast, **relative L2 distance** measures the numerical precision of each sample compared with ground truth. Formally, it's defined as:

$$R\text{-}\ell_2 = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{|s_n - \hat{s}_n|}{s_{max} - s_{min}}\right)^2 \tag{4.15}$$

Table 4.1: Performance comparison on MTL-AQA dataset. 'w/o DD' means that training and test processes do not utilize difficulty degree labels, 'w/ DD' means experiments utilizing difficulty degree labels.

Method (w/o DD)	Sp. Corr.	R-$\ell_2(\times 100)$
Pose+DCT [134]	0.2682	-
C3D-SVR [135]	0.7716	-
C3D-LSTM [135]	0.8489	-
MSCADC-STL [131]	0.8472	-
C3D-AVG-STL [131]	0.8960	-
MSCADC-MTL [131]	0.8612	-
C3D-AVG-MTL [131]	0.9044	-
USDL [139]	0.9066	0.654
CoRe [1]	0.9341	0.365
TSA-Net [140]	0.9422	-
Ours	**0.9451**	**0.3222**
Method (w/ DD)	Sp. Corr	R-$\ell_2(\times\ 100)$
USDL [139]	0.9231	0.468
MUSDL [139]	0.9273	0.451
CoRe [1]	0.9512	0.260
Ours	**0.9607**	**0.2378**

Implementation Details

We adopt the I3D backbone pretrained on Kinetics [26] as our local spatial-temporal feature extractor. The Adam optimizer is applied with a learning rate 1×10^{-4} for the backbone and transformer module. The learning rate for the regression head is set to 1×10^{-3}. The feature dimension is set to 512 for the transformer block. We select 10 exemplars for each test sample during the inference stage to align with previous work [1] for fair comparisons. As for the data-preprocessing on AQA-7 and MTL-AQA datasets, we sample 103 frames following previous works for all videos. Since our proposed method requires more fine-grained temporal information, unlike previous work that segmented the sample frames into 10 clips, we segment the frames into 20 overlapping clips each containing 8 continuous frames. As for the JIGSAWS dataset, we uniformly sample 160 frames following [139] and divide them into 20 non-overlapping clips as input of the I3D backbone. We select exemplars from the same difficulty degree on MTL-dataset during the training stage. For AQA-7 and

Table 4.2: Performance comparison on AQA-7 dataset.

Sp. Corr	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. Corr.
Pose+DCT [134]	0.5300	0.1000	-	-	-	-	-
ST-GCN [181]	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM [135]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR [135]	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
JRG [138]	0.7630	0.7358	0.6006	0.5405	0.9013	0.9254	0.7849
USDL [139]	0.8099	0.7570	0.6538	0.7109	0.9166	0.8878	0.8102
CoRe [1]	0.8824	0.7746	0.7115	0.6624	0.9442	0.9078	0.8401
TSA-Net [140]	0.8379	0.8004	0.6657	0.6962	**0.9493**	0.9334	0.8476
Ours	**0.8969**	**0.8043**	**0.7336**	**0.6965**	0.9456	**0.9545**	**0.8715**
R-ℓ_2($\times$100)	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. R-ℓ_2
C3D-SVR [135]	1.53	3.12	6.79	7.03	17.84	4.83	6.86
USDL [139]	0.79	2.09	4.82	4.94	0.65	2.14	2.57
CoRe [1]	0.64	1.78	3.67	3.87	0.41	2.35	2.12
Ours	**0.53**	**1.69**	**2.89**	**3.30**	**0.33**	**1.33**	**1.68**

Table 4.3: Performance comparison on JIGSAW dataset.

Sp. Corr.	S	NP	KT	Avg.
ST-GCN [181]	0.31	0.39	0.58	0.43
TSN [182]	0.34	0.23	0.72	0.46
JRG [138]	0.36	0.54	0.75	0.57
USDL [139]	0.64	0.63	0.61	0.63
MUSDL [139]	0.71	0.69	0.71	0.70
CoRe [1]	0.84	0.86	0.86	0.85
Ours	**0.88**	**0.88**	**0.91**	**0.89**
R-ℓ_2	S	NP	KT	Avg.
CoRe [1]	5.055	5.688	2.927	4.556
Ours	**2.722**	**5.259**	**3.022**	**3.668**

JIGSAWS datasets, all exemplars come from the same coarse classes. The exemplar videos are chosen based on difficulty degree for MTL-AQA dataset. For MTL-AQA, the choice from outside the same difficulty degree will lead to performance drop.

4.4.2 Comparison to state-of-the-art

We evaluate our proposed framework against state-of-the-art techniques across three datasets, tabulated in Tab.4.1, Tab.4.2, and Tab.4.3. Consistently, our methodology leads the benchmarks, surpassing earlier attempts across various settings.

MTL-AQA dataset Analysis: The MTL-AQA dataset offers a unique challenge with its inclusion of difficulty degree labels. The final quality score of each

video results from the product of its raw score and respective difficulty. Two experimental settings employed here, namely 'w/o DD' and 'w/ DD', denote the non-utilization and utilization of the difficulty degree labels, respectively, during the training and test phases [1].

In the 'w/ DD' setup, we harness the difficulty labels by matching the test video with exemplar videos of analogous difficulty. The final quality score is then derived by multiplying the estimated raw score with its difficulty. As depicted in Tab.4.1, our approach achieves a Spearman Correlation (Sp. Corr.) of 0.9607 and R-ℓ_2 of 0.2378. This clearly outshines the performance of the contemporary tree-based contrastive regression methodology, CoRe [131]. This attests to the potency of our novel temporal parsing transformer. A notable highlight is our simplified contrastive regression utilizing two shallow MLPs, instead of the intricate tree structure as proposed in [1]. The efficiency of our transformer in deriving granular part representations eases the regression process.

The 'w/o DD' setup demonstrates our method's robustness even in the absence of difficulty labels. Here, our method records a Sp. Corr of 0.9451 and R-ℓ_2 of 0.3222, besting both CoRe and the recent TSA-Net [140]. Interestingly, while TSA-Net leverages an external VOT tracker [183] to pinpoint human positions, enhancing backbone features, our method focuses primarily on temporal parsing. This suggests that the synergy of our approach with attention mechanisms, akin to [140], might further elevate our results.

AQA-7 Dataset Analysis: Tab.4.2 showcases our method's prowess across the AQA-7 dataset. Leading in 5 categories, our results remain competitive in the remaining category. On average, our approach surpasses CoRe by a notable 3.14 Corr.($\times$100) and TSA-Net by 2.39 Corr.($\times$100). The minuscule R-ℓ_2 of 1.68($\times$100) further highlights the superior capabilities of our temporal parsing transformer. Note that the performance of [140] is close or slightly better than our performance, is is because AQA-7 has less samples per categories, which magnify the advantage of external dataset and model used in [140].

JIGSAW Dataset Analysis: The JIGSAW dataset, despite being the smallest, demands rigorous evaluation, prompting us to conduct 4-fold cross-validation for

Table 4.4: Ablation study of different components on MTL-AQA dataset.

Method	TPT	L_{rank}	$L_{sparsity}$	Sp. Corr.	R-ℓ_2
Baseline	×	×	×	0.9498	0.2893
	✓	×	×	0.9522	0.2742
	✓	✓	×	0.9583	0.2444
Ours	✓	✓	✓	**0.9607**	**0.2378**

each category, as recommended by prior studies [1, 139]. Our technique records an average correlation (Corr.) of 0.89 and R-ℓ_2 of 3.668, setting a new benchmark in state-of-the-art performance.

In essence, these experiments underscore the effectiveness and generalizability of our method. The consistent top-notch performance across diverse settings and datasets reaffirms the advantages of our temporal parsing transformer over other contemporaneous strategies.

4.4.3 Ablation Study

In this subsection, we perform ablation studies to evaluate the effectiveness of our proposed model components and designs. All of our ablation studies are performed on MTL-AQA dataset under 'w/ DD' setting. We build a baseline network that directly pool the clip features without transformer, and utilize the resulting holistic representation to perform contrastive regression.

Different model components In this work, we propose a novel temporal parsing transformer(TPT), and exploit the ranking loss(L_{rank}) and sparsity loss($L_{sparsity}$) on cross attention responses to guide the part representation learning. We first perform experiments to show the effectiveness of each design, the results are shown in Tab.4.4. We can observe that with only TPT, the performance only improves marginally from 0.9498 Corr. to 0.9522 Corr.. With the ranking loss, the performance is significantly improved, demonstrating the importance of temporally ordered supervision strategy. The sparsity loss further improves the performance, showing that the discrimination of parts is also important. We also conduct ablation experiments on the AQA-7 dataset as shown in Tab. 4.5. We can observe that with

Table 4.5: Ablation study of different components on AQA-7 dataset.

Sp. Corr	TPT	L_{rank}	$L_{spar.}$	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg.
Baseline	×	×	×	0.8597	0.7117	0.6625	0.6342	0.9336	0.9189	0.8229
	✓	×	×	0.8889	0.7837	0.6753	0.6722	0.9401	0.9279	0.8478
	✓	✓	×	0.8892	0.7999	**0.7367**	0.6722	0.9429	0.9440	0.8622
Ours	✓	✓	✓	**0.8969**	**0.8043**	0.7336	**0.6965**	**0.9456**	**0.9545**	**0.8715**
R-ℓ_2(×100)	TPT	L_{rank}	$L_{spar.}$	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg.
Baseline	×	×	×	0.70	2.18	4.03	4.08	0.78	2.40	2.36
	✓	×	×	0.76	1.73	4.13	3.50	0.47	1.64	2.04
	✓	✓	×	0.54	1.69	3.31	3.49	0.49	1.94	1.91
Ours	✓	✓	✓	**0.53**	**1.68**	**2.89**	**3.30**	**0.33**	**1.33**	**1.68**

only TPT component, the average performance of six categories is improved from 0.8229 Corr. to 0.8478 Corr. With our ranking loss and sparsity loss, the performance is further significantly improved from 0.8478 Corr. to 0.8715 Corr., showing the effectiveness of our temporally ordered supervision strategy.

Table 4.6: Ablation study of different number of queries on MTL-AQA dataset.

Query number	Sp. Corr.	R-ℓ_2
3	0.9562	0.2583
5	**0.9607**	**0.2378**
7	0.9572	0.2337

Number of queries We show the ablation study of the number of queries in Tab. 4.6. We found that too many queries hurt performance. As a result, we choose query number to be 5.

Number of decoder layers Our temporal parsing transformer has a multi-layer decoder structure, we show the ablation study of different decoder layers in Tab.4.7. We found that 2-layer decoder achieves comparable performance compared with 3-layer decoder. We finally select 2-layer decoder for model simplicity.

Different relative representation generation Since we have obtained part representations from TPT for each video, we may have two options to generate relative representation for contrastive regression. For the first option, we can first fuse the part representations with a pooling operation for each video, then each video takes the part-enhanced holistic representation to estimate the relative score.

Table 4.7: Ablation study of different number of decoder layers on MTL-AQA dataset.

Layer number	Sp. Corr.	R-ℓ_2
0 (baseline)	0.9498	0.2893
1	0.9563	0.2736
2	**0.9607**	0.2378
3	0.9594	**0.2303**

For the second option, which is our proposed strategy, we first compute a part-wise relative representation and then apply the AvgPool operation over the parts. We compare the results of above options in Tab.4.8. We can see that the part-wise strategy outperforms part-enhanced strategy. It's worth noting that the part-enhanced approach also outperforms our baseline network, which implies that each part indeed encodes fine-grained temporal patterns.

Table 4.8: Ablation study of different relative representation generation on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Part-enhanced holistic	0.9578	0.2391
Part-wise relative + AvgPool(ours)	**0.9607**	**0.2378**

Different part generation strategies Our method utilizes the temporal parsing transformer to extract part representations. In this ablation study, we compare our method with the other two baseline part generation strategy, shown in Tab. 4.9. The first strategy utilizes the adaptive pooling operation cross temporal frames to down-sample the origin T clip representation into K part representations. The second strategy replaces the above adaptive pooling with a temporal convolution with stride $\lfloor T/K \rfloor$, resulting in a representation with K size. We found that both strategies introduce minor improvements as they can not capture fine-grained temporal patterns.

Effect of position encoding Distinct from traditional transformers [8, 169], our transformer's decoding mechanism eschews temporal position encoding. A comparative analysis of various position encoding strategies, applied on the memory (clip)

Table 4.9: Ablation study of different part generation strategies on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Adaptive pooling	0.9509	0.2757
Temporal conv	0.9526	0.2758
TPT(ours)	**0.9607**	**0.2378**

Table 4.10: Ablation study of effect of order guided supervision on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
Baseline	0.9498	0.2893
Diversity loss	0.9538	0.2655
Ranking loss(ours)	**0.9607**	**0.2378**

and query (part), is tabulated in Tab.4.11. To infuse position encoding into the queries, we adopt the cosine series embedding of $\lfloor T/K \rfloor \times i$ to the i-th learnable query. This ensures the queries possess positional guidance uniformly distributed across temporal clips. We meticulously ensure the retention of ranking loss and sparsity, aiming for an impartial evaluation.

Tab.4.11 reveals an intriguing observation: the integration of position encoding appears to hinder the learning of temporal patterns. Delving deeper, one can surmise that the omission of positional encoding might be beneficial for several reasons. A pivotal point to consider is the nature of our loss function. Specifically, our loss function primarily aims at distinguishing various time steps. Introducing positional encoding inadvertently skews the model's focus. Instead of assimilating valuable semantic information, the model becomes excessively preoccupied with the time step. Such a tunnel-visioned approach diminishes the model's ability to grasp and exploit meaningful patterns, inevitably leading to a decline in performance. Consequently, circumventing positional encoding in our methodology is not merely an architectural choice, but a deliberate strategy to ensure the model emphasizes semantic relevance over mere temporal localization.

Effect of order guided training strategy Our ranking loss on the attention centers consistently encourages the temporal order of atomic patterns. To verify

Table 4.11: Ablation study on effect of positional encoding on MTL-AQA dataset.

Pos. Encode	Memory(clip)	Query(part)	Sp. Corr.	R-ℓ_2
	✓	✓	0.9526	0.2741
	✓	×	0.9532	0.2651
Proposed	×	×	**0.9607**	**0.2378**

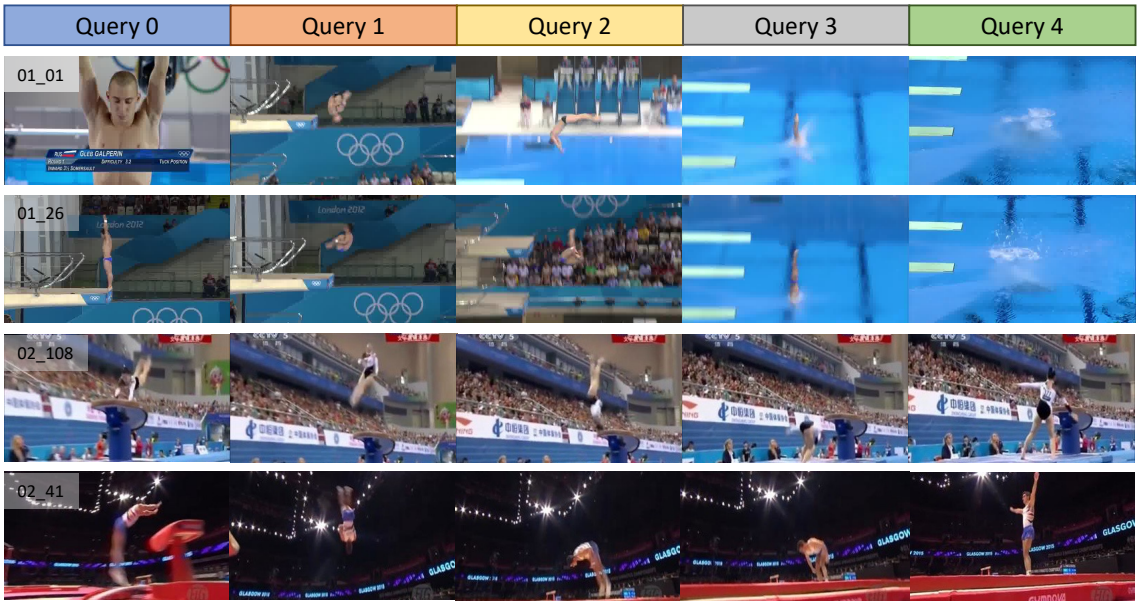


Figure 4.3: Visualization of the frames with the highest attention responses in decoder cross attention maps on MTL-AQA and AQA-7 datasets. Each row represents a test video from different representative categories (diving from MTL-AQA, gymnastic vault from AQA-7), whose ID is shown in the left first frame. Different columns correspond to temporally ordered queries (note that different categories do not share the same query embeddings). The above results show that our transformer is able to capture semantic temporal patterns with learned queries.

the importance of such order guided supervision, we replace the ranking loss to a diversity loss following the Associative Embedding [184] to push attention centers: $L_{div} = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \exp^{-\frac{1}{2\sigma^2}(\bar{\alpha}_i - \bar{\alpha}_j)^2}$. Compared with L_{rank}, L_{div} does not encourage the order of queries, but keeps diversity of part representations. As shown in Tab. 4.10, the performance significantly drops from 0.9607 Corr. to 0.9538 Corr., demonstrating the effectiveness of our order guided training strategy.

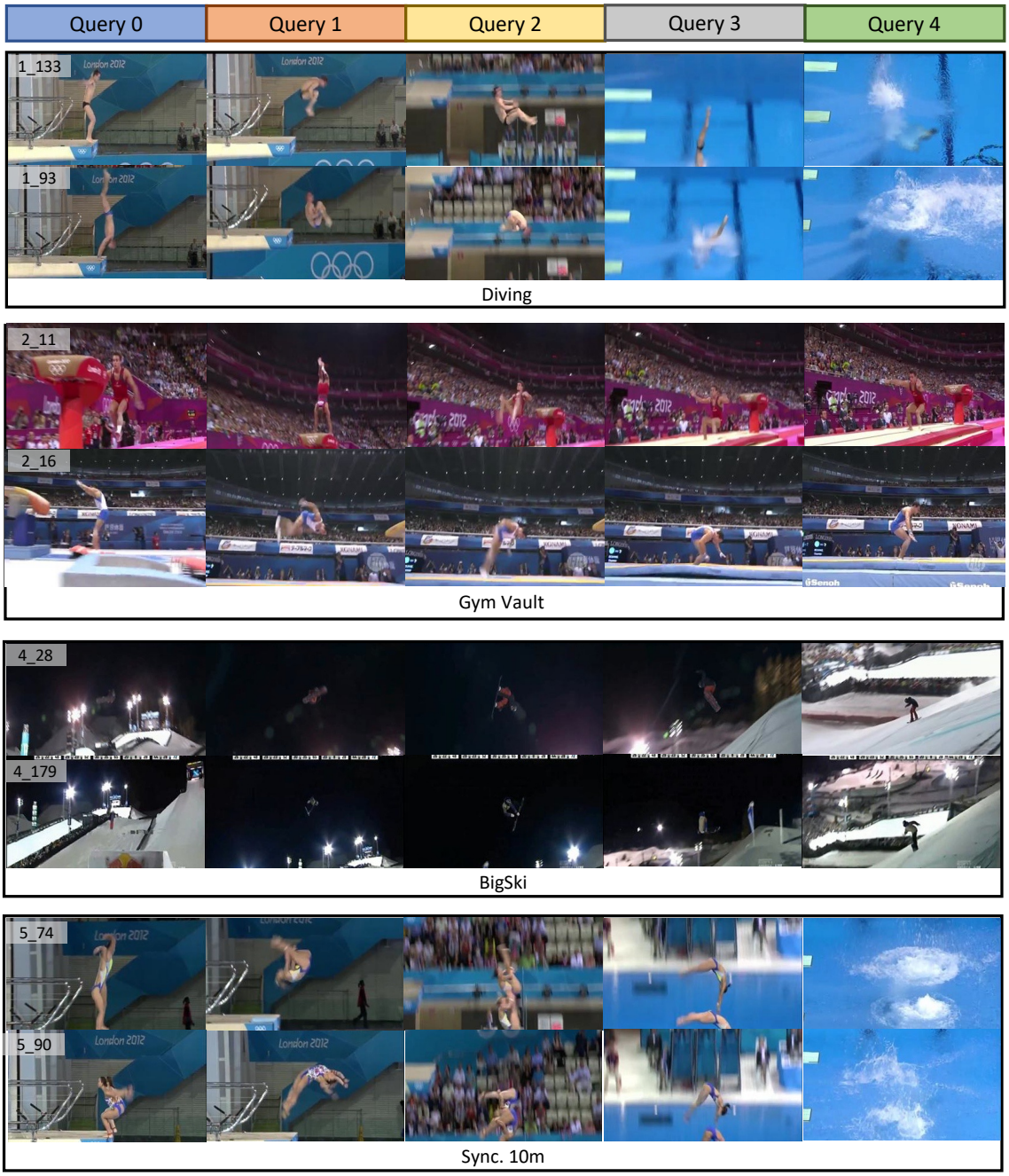


Figure 4.4: Visualization of the frames with highest attention responses in decoder cross attention maps on AQA-7 dataset. Each row represents a test video from different representative categories (diving, gymnastic vault, big air snowboarding, synchronous diving - 10m platform), whose ID is shown in the left first frame. Different columns correspond to temporally ordered queries (note that different categories do not share same query embeddings). The above results show that our transformer is able to capture semantic temporal patterns with learned queries.

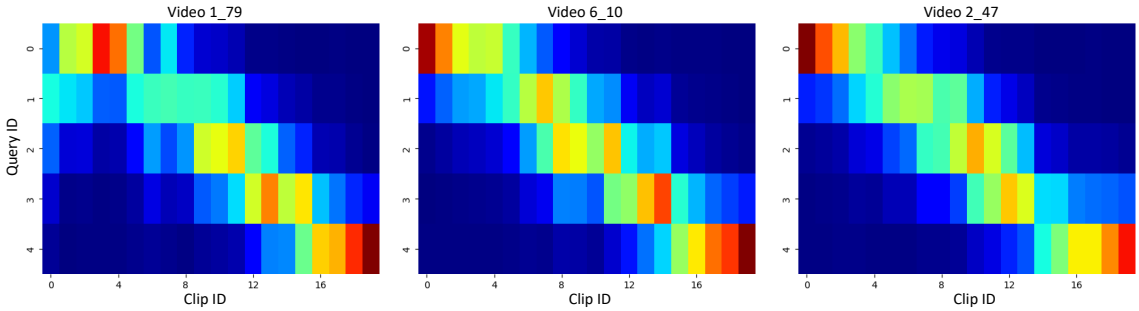


Figure 4.5: Visualization of cross attention maps on three video samples from MTL-AQA dataset, where video IDs are shown on the top. In each subfigure, each row indicates one query, and each column indicates one clip. We can observe that the bright grids(with high attention responses) have a consistent temporal order due to ranking loss, and the attention maps are sparse due to our sparsity loss.

4.4.4 Visualization results

We provide some visualization results in Fig.4.3, Fig.4.4, Fig.4.3and Fig.4.6. Samples are from MTL-AQA dataset trained under 'w/ DD' setting and AQA-7 dataset. In Fig.4.3, we visualize the clip frames with the highest attention responses in cross attention maps of the last decoder layer on MTL-AQA dataset. Since each clip consists of multiple frames, we select the middle frame of a clip as representative. We can observe that our transformer can capture semantic temporal patterns with learned queries.

In Fig.4.4, we visualize the frames with the highest attention responses in cross attention maps of the last decoder layer on AQA-7 dataset. Four representative categories are selected for showing, the other two categories are similar. We can observe that our transformer is capable of parsing a diving video into temporal patterns such as the take-off, the flight, and the entry with learned queries on AQA-7 dataset. The effect on other categories are similar.

In Fig.4.5, we visualize the cross attention maps on MTL-AQA dataset. We can observe that the attention responses have a consistent temporal order due to our designed ranking loss, and they are also sparse due to our sparsity loss.

In Fig.4.6, we visualize the cross attention maps on AQA-7 dataset for all categories. We can observe similar results from the MTL-AQA dataset that the attention responses have a consistent temporal order and are adaptive for different video sam-

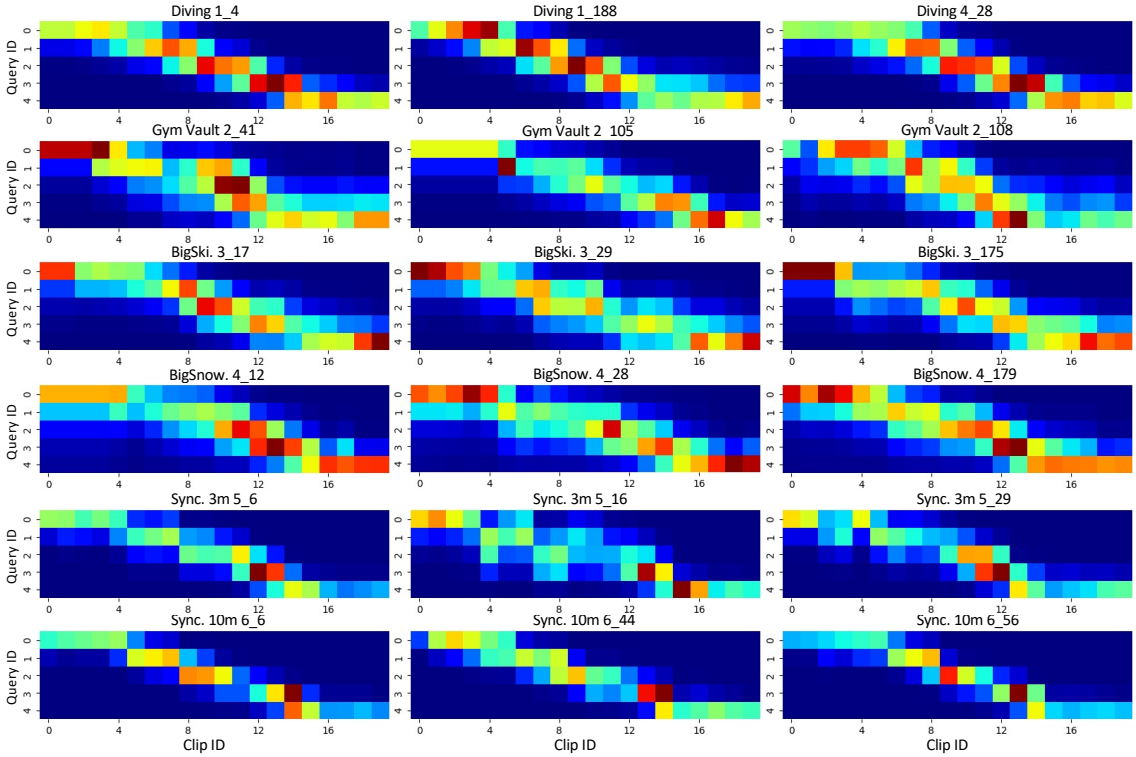


Figure 4.6: Visualization of cross attention maps on video samples from the AQA-7 dataset covering all categories (diving, gymnastic vault, big air skiing, big air snowboarding, synchronous diving - 3m springboard and synchronous diving - 10m platform), where video IDs and category names are shown at the top of each attention map. In each subfigure, each row indicates one query, and each column indicates one clip. We can observe that the bright grids(with high attention responses) have a consistent temporal order due to ranking loss, and the attention maps are sparse due to our sparsity loss.

ples, which demonstrates the effectiveness of our proposed ranking loss and sparsity loss. We also observe that the categories with more samples (diving category with 370 training samples) have more distinguishable cross attention responses of parts than categories with fewer samples (91 training samples from synchronous diving - 10m platform). We suppose that more training samples might be beneficial to learn atomic patterns.

4.5 Conclusion

In this chapter, we propose a novel temporal parsing transformer for action quality assessment. We utilize a set of learnable queries to represent the atomic temporal

patterns, and exploit the transformer decoder to convert clip-level representations to part-level representations. To perform quality score regression, we exploit the contrastive regression framework that first computes the relative pairwise representation per part and then fuses them to estimate the relative score. To learn the atomic patterns without part-level labels, we propose two novel loss functions on cross attention responses to guide the queries to attend to temporally ordered clips. As a result, our method is able to outperform existing state-of-the-art methods by a considerable margin on three public benchmarks. The visualization results show that the learnt part representations are semantic meaningful.

CHAPTER 5

Towards Cycle-Counterfactual Action Quality Assessment

In Chapter 4, we embarked on a comprehensive journey into Interaction-level action understanding, emphasizing the nuances of Action Quality Assessment (AQA) based on specific human rules. As we transition to Chapter 5, our exploration plunges deeper into the complexities of AQA. Where Chapter 4 showcased the strengths of the temporal parsing transformer in capturing atomic temporal patterns of actions, Chapter 5 sharpens the focus on the granularity of action analysis.

Building on the groundwork from the preceding chapter, Chapter 5 delves into the sophisticated realm of sub-actions or part-level actions within videos. While the previous chapter underscored the potency of modeling these sub-actions, our current discourse zooms in on the intricacies of assessing the quality of these sub-actions, a dimension often under-represented due to the limited supervision in AQA datasets.

Navigating this domain presents its own set of challenges. The primary obstacle is the ambiguity tied to associating these meticulous part representations with distinct quality scores, especially in the light of sparse supervision. Our answer to this challenge is the innovative Cycle-Counterfactual method. This approach leans into evaluating the constancy or discrepancies in assessments when nuances in the quality of sub-actions are introduced. Augmenting the robustness and fidelity of

such modeling, we integrate an advanced disentangling technique.

In Chapter 5, we seek to magnify the lens on AQA, intimately scrutinizing the sub-actions and gauging their influence on overarching quality. By weaving together the insights from Chapters 4 and 5, our narrative offers an all-encompassing view of AQA, melding depth with breadth. The efficacy of our methodology is underscored by thorough experiments, clearly delineating our approach's edge over prevalent methods in various AQA benchmarks.

5.1 Introduction

Action quality assessment (AQA), aiming to estimate how well a specific action is performed, has drawn growing attention because of its valuable potential for diverse real-world applications such as healthcare [127], sports analysis [131, 134, 135, 168]. Traditional video action recognition aims to classify different categories of action sequences correctly [22, 91, 110, 182, 185–190], and video-level coarse supervision is sufficient to solve the problem. On top of them, AQA mainly focuses on the fine-grained understanding of action quality, where the quality is estimated into a specific score. There are several works dedicated to pursuing more precise assessments and obtaining satisfactory results, which can be divided into two main directions, namely regression-based and ranking-based. Regression-based methods directly estimate the quality of actions via score regression [131, 134, 135, 137, 138], while ranking based-methods focus on estimating the relationship between pairs of videos [1, 128, 129, 191].

Most existing Action Quality Assessment (AQA) methods focus on the overall quality of the entire action sequence and do not explore the details of individual stages, i.e., sub-actions. However, take the sport diving as an example, a professional expert judging an athlete's performance would carefully observe each sub-action to assign a quality score. Therefore, it is important to evaluate the quality of sub-actions since they may contribute differently to the overall quality score and different combinations of sub-action quality may result in the same overall quality score. Figure 5.1 presents two action samples with similar overall quality scores but

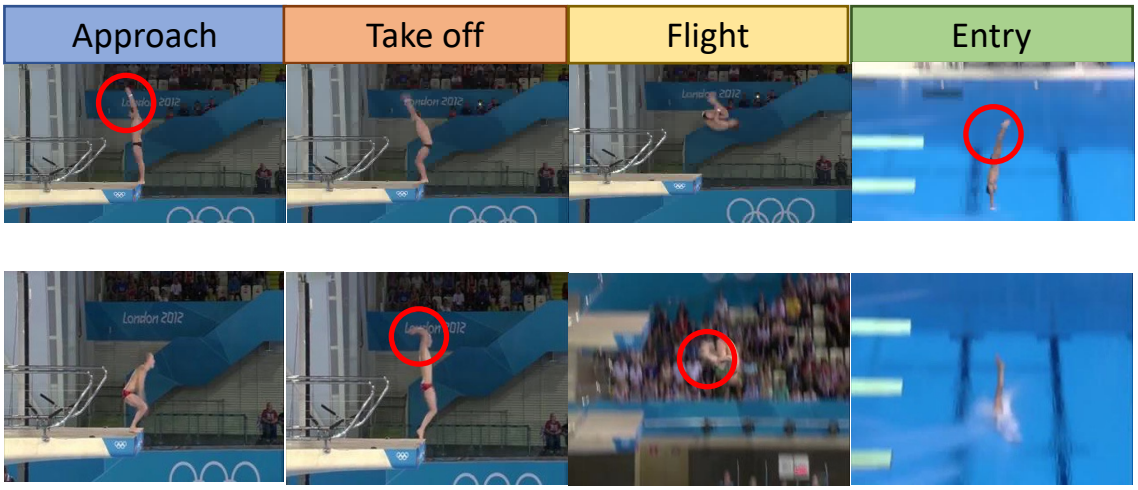


Figure 5.1: Two action samples are selected as our example. They have same overall quality score but different quality of sub-actions.

notable differences in sub-action quality, emphasizing the importance of sub-action quality evaluation. For instance, the first sample exhibits poor performance in the "Entry" sub-action due to a bad entry angle, while the second sample performs poorly in the "Flight" sub-action due to an unaligned leg. Such examples illustrate that the overall quality score is not sufficient to describe the quality of sub-actions and that sub-action quality is the basis for deriving the overall score. However, due to the limited supervision available in real-world sports events, where only the overall score is given, deep learning models face the challenge of learning a comprehensive process of how the score is derived, which may lead to overfitting to specific sub-action quality combinations.

In recent years, causal learning has gained significant attention in the field of computer vision for its ability to model relationships among variables and events [192–194]. Among the various types of causal models applied in computer vision, counterfactual reasoning has inspired several studies, such as generative zero-shot learning [195] and domain adaptation [196], where the ability to imagine what would happen if a particular variable were different is leveraged to generate faithful counterfactual samples based on the input. In the context of action quality assessment, it is important to explore the effect of individual sub-actions on the overall quality score. Although assigning scores to each sub-action is a natural approach to effectively leverage information from sub-actions for action quality score estimation, it is

impeded by the lack of annotated sub-action scores, rendering it impractical to train the model. This is a limitation as different sub-actions may have varying impacts on the overall quality score, and ignoring these sub-actions may result in incomplete or inaccurate assessments. Hence, it is essential to develop new methods that can exploit the information of sub-actions even in the absence of annotated sub-action scores.

Therefore inspired by the potential of counterfactual reasoning, we investigate whether we can change the quality of a particular sub-action in a given video while keeping the rest of the sub-actions unchanged in this work. If so, we can then explore the sub-action quality via the difference between the original samples and the counterfactual samples. To achieve this, we propose a novel generative cycle-counterfactual framework that leverages the imagination–the third ladder of causal theory [197] to better exploit the features of sub-actions. The framework disentangles the features of sub-actions into quality attributes and context attributes, allowing for targeted manipulation of the quality attributes while preserving the context information. Specifically, the quality of sub-actions is altered from the original sample with specific relative scores to generate counterfactual samples, where the relative score is built upon different quality centers. The framework then reverses different sub-actions in the counterfactual samples, resulting in diverse video features that contain the same overall quality score yet different quality of sub-actions. Through this cycle framework, the model is explicitly encouraged to take advantage of the quality of sub-actions, which positively enriches the learning space. Moreover, to ensure counterfactual faithfulness [195], we propose a regularization technique that eliminates the context information from the quality attribute and maintains the quality information fidelity after counterfactual generation. In summary, the main contributions of this paper are listed as follows:

- We identify the critical challenge of exploiting the quality of sub-actions in action quality assessment, which is a key factor in replacing human judges with algorithms. To tackle this issue, we propose a novel approach that leverages counterfactual reasoning, which enables us to explore the quality of sub-actions.

- The cycle-counterfactual framework consists of three main components: counter-

factual generation, cycle-counterfactual consistency, and disentanglement module. The generated counterfactual samples enable us to fully explore the information within sub-actions. The cycle-counterfactual consistency ensures that the generated counterfactual samples are consistent with the original samples, and the disentanglement module guarantees the faithfulness of the counterfactual generation process.

- Our proposed framework achieves state-of-the-art performance on three widely used action quality assessment datasets, including MTL-AQA [131], AQA-7 [130], and JIGSAWS [170], which demonstrates the effectiveness and robustness of our approach.

Within the larger context of interactive-level action understanding, this chapter delves into the niche realm of assessing actions via specific human rules. By spotlighting the importance of evaluating individual sub-action qualities, we address a pivotal aspect of the broader understanding spectrum. This approach to action quality assessment contributes meaningfully to the overall objective, presenting a detailed perspective that enhances the depth of interactive-level action understanding.

5.2 Related Work

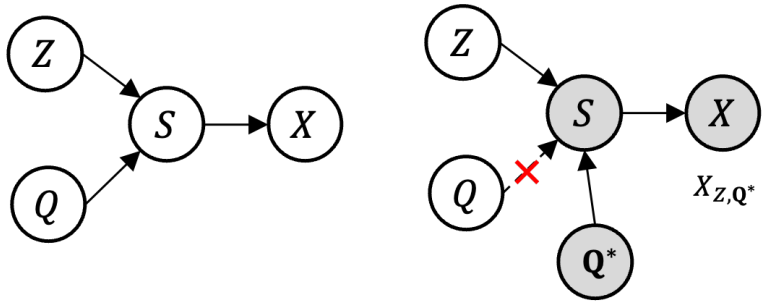
Here, we give a brief overview of the causal inference used in our framework. **Causality-Inspired Methods.** Counterfactual thinking and causal inference have inspired several studies in computer vision, including visual explanation [198–201], scene graph generation [202, 203], image recognition [203], video analysis [204, 205], zero-shot and few-shot learning [195,206,207], incremental learning [208], representation learning [209,210], semantic segmentation [211], and vision-language tasks [212–216]. For example, [216] proposed a counterfactual framework that is able to capture the language bias as the direct causal effect of questions on answers. Our framework first proposes to use counterfactual generation on Action quality assessment task aims to exploit sub-action quality by generating quality changed counterfactual samples.

5.3 Method

In this chapter, we propose a cycle counterfactual generative framework to take advantage of the informative sub-actions for overall action quality assessment.

5.3.1 Overview

Following previous works [139], corresponding scores of quality for the video action sequences are estimated via a classification framework. We adopt I3D to extract video frame into ordered clip features $\mathbf{X} \in \mathbb{R}^{T \times D}$, where T is the number of clips. Then, the video feature sequence is encoded into part-level representation (i.e., the features of sub-actions) $\mathbf{S} = \{\mathbf{s}_k \in \mathbb{R}^D\}_{k=1}^K$, where K is the number of action parts, and $K << T$. Each segment $\mathbf{s}_k$, which are able to provide finer level representation than global actions, not only indicates the potential quality information (i.e., how well the current sub-action performed) but also contains context information (camera view, type of sub-action). However, different information may affect the final score estimation in different aspects, hence, the context information is disentangled apart from the quality information, resulting in quality attribute $\mathbf{Q} = \{\mathbf{q}_k \in \mathbb{R}^{D^q}\}_{k=1}^K$ and quality-agnostic context attribute $\mathbf{Z} = \{\mathbf{z}_k \in \mathbb{R}^{D^z}\}_{k=1}^K$ based on $\mathbf{S}$. Then, quality classifier $f_q(\cdot)$ is adopt to predict the final quality score based on $\mathbf{Q}$. We also adopt a context classifier $f_z(\cdot)$, which takes $\mathbf{Z}$ as input to predict context information such as sub-action type. Since only video-level coarse scores (i.e., only one score for an entire video of one action) are available, the lack of explicit clip-level annotation is challenging to explore the quality relationship between each global action and its corresponding sub-actions, we propose a Cycle Counterfactual Generation framework, which is shown in Fig.5.3. The Cycle process consists of a forward and a backward counterfactual generation process, which aims to generate a set of counterfactual samples with the same overall quality score compared with the original one, but different sub-action quality. We can then exploit sub-action quality by learning to distinguish samples with the same sub-action quality and samples with different sub-action quality, which allows better estimation of the final quality score. We demonstrate the details of methods in the following



(a) Causal Graph (a) Counterfactual Notations

Figure 5.2: (a) Example of causal graph. (b) Example of counterfactual notations, white nodes are at the value of $Q = \mathbf{Q}$ while gray nodes are at the value $Q = \mathbf{Q}^*$. Note that S contains multiple parts. For ease of notation we omit the subtitle indices.

sections.

5.3.2 Counterfactual Generation

A comprehensive overview of causal inference is out of the scope of this chapter, we refer interested readers to our related works section and [195], and Figure 2 shows the assumption of Generative Causal Model (GCM) [217]. A random variable is denoted as a capital (e.g., X) and its observed matrix-valued sample is denoted as a bold letter (e.g., $\mathbf{X}$). Take diving as an example, the input video feature X is determined by several sub-actions such as approach, take off, flight and entry, which can be represented by a set of random variables S_1, S_2, ... S_K, where K is the number of sub-actions. Each sub-action S_k is jointly determined by its *quality attribute* Q_k and quality agnostic *sample attribute* (e.g., camera view, take off position, action type) Z_k.

The generative causal process is illustrated in Fig.5.2: $Z_k \rightarrow S_k, Q_k \rightarrow S_k$ and $S_1, S_2, ..., S_K \rightarrow X$.

Our GCM consists of a generation process and an inference process. Specifically, given Z and Q, we can generate X by sampling from the conditional distribution $X = P_\theta(X|Q, Z)$. While given X, we can infer Q and Z through the posterior $P_\phi(Z|X)$ and $P_\psi(Q|X)$. (Note that we omit the process of $S \rightarrow X$ for ease of notation.)

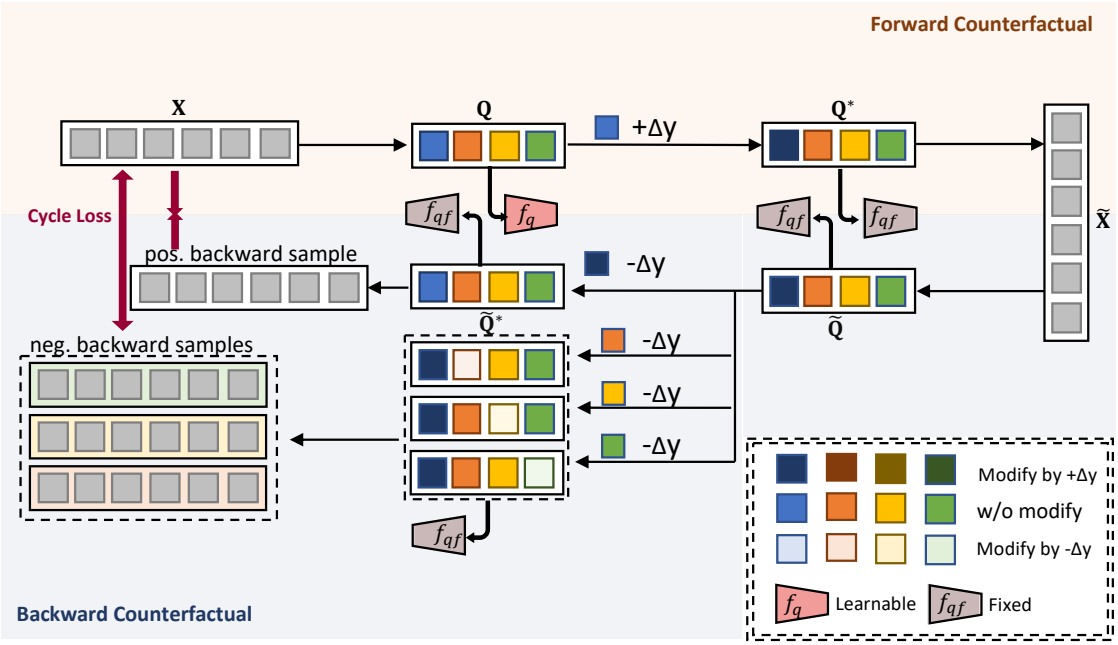


Figure 5.3: Overview of our Cycle-Counterfactual framework. The cycle-counterfactual framework consists of a forward counterfactual and a backward counterfactual process. During the forward counterfactual, the first sub-action's quality attribute (shown as blue color) is altered by Δy, resulting in $\mathbf{Q}^*$. The forward counterfactual sample $\tilde{\mathbf{X}}$ is then derived. In the backward process, the framework attempts to reverse the overall quality change by altering each part of the quality attribute $\tilde{\mathbf{Q}} \sim P_\psi(Q|X = \tilde{\mathbf{X}})$ by $-\Delta y$ in parallel, resulting in K backward samples with same quality score compared with $\mathbf{X}$. The first backward sample altered the first part (shown in deep blue) of $\tilde{\mathbf{Q}}$, resulting in no sub-action quality change compared with X, which we denote as the positive backward sample. The framework could exploit sub-action's quality by pulling close the distance between $\mathbf{X}$ and positive backward samples while pushing the negative samples from $\mathbf{X}$. We omit the disentangle module to better explain the idea of cycle-counterfactual.

Given a video feature $\mathbf{X}$ sampled from X, we use the GCM to generate counterfactual samples $\tilde{\mathbf{X}} = X_{\mathbf{Z},\mathbf{Q}^*}$ following the three steps of computing counterfactuals.

- **Abduction.** *"given the fact that* $Q = \mathbf{Q}, Z = \mathbf{Z}$*"*. We derive the endogenous quality attribute $\mathbf{Q}$ and context attribute $\mathbf{Z}$ given the evidence $X = \mathbf{X}$. In our GCM, we can sample from the posterior $\mathbf{Q} \sim P_\psi(Q|X = \mathbf{X})$ and $\mathbf{Z} \sim P_\phi(Z|X = \mathbf{X})$.

- **Action.** *"had* Q *been* $\mathbf{Q}^*$*"*. Here $\mathbf{Q}^*$ is the sample derived quality attribute $\mathbf{Q}$ that changed the k^{th} sub-action quality by a relative quality score Δy via the relative score embedding module $f_{embed}(\mathbf{Q}, k, \Delta y)$. (relative score embedding

module will be introduced latter in this section.)

- **Prediction.** *"X would be* $\tilde{\mathbf{X}}$*"*. conditioning on the sample inferred $Z = \mathbf{Z}$ (fact) and the intervention quality attribute $Q = \mathbf{Q}^*$ (counterfact), we can generate the counterfactual sample $\tilde{\mathbf{X}}$ from $P_\theta(X|Z = \mathbf{Z}, Q = \mathbf{Q})$, which we denote as $\tilde{\mathbf{X}} = X_{\mathbf{Z},\mathbf{Q}^*}$. We can also generate context changed counterfactual from $X_{\mathbf{Z}^j,\mathbf{Q}}$, where $\mathbf{Z}^j$ is context attribute derived from another video feature.

Relative Score Embedding. The relative score embedding module aims to alter the quality of the given *quality attribute* $\mathbf{Q}$'s k^{th} sub-action by a relative quality score Δy. A straightforward way to embed relative score Δy to a feature is to use MLPs to embed the relative score. However, the quality representation is not uniformly distributed. To be specific, the same relative score would reflect more obvious action quality variation, given a lower scored action than a higher one. So it is not reasonable to encode relative scores given different *quality attribute* with the same embedding function. To encode the relative score Δy while considering the impact of different *quality attribute*, we propose a relative score embedding module. The module utilizes the parameters $\mathbf{\Theta}_q \in \mathbb{R}^{D\times C}$ from the quality classifier $f_q(\cdot)$, where C is the number of quality class center. Each $\boldsymbol{\theta}_q^c \in \mathbb{R}^D$ represent the c^{th} quality class center. To take the input action quality into consideration, we first estimate the score of input action s_{pred} via the quality classifier $f_q(\mathbf{Q})$, then we can find the corresponding class center $\boldsymbol{\theta}_q^{pred}$. Based on s_{pred}, we can derive the target score $s_{target} = s_{pred} + \Delta y$. We can then find the class center for the target score $\boldsymbol{\theta}_q^{tgt}$. Then we sent the $\boldsymbol{\theta}_q^{pred}$ and $\boldsymbol{\theta}_q^{tgt}$ through several layers of MLPs and output the final relative score embedding as $\mathbf{r} \in \mathbb{R}^{D^q}$. We can then alter the k^{th} sub-action $\mathbf{q}_k$ from input *quality attribute* $\mathbf{Q}$ by $\mathbf{q}_k^* = \mathbf{q}_k + \mathbf{r}$. We denote the whole process as $\mathbf{Q}^* = f_{embed}(\mathbf{Q}, k, \Delta y)$.

Counterfactual Expectation. Since there's no ground truth, for generated counterfactual samples $\tilde{\mathbf{X}} = X_{\mathbf{Z},\mathbf{Q}^*}$, to guarantee the quality of the counterfactual sample is Δy higher than the fact sample $\mathbf{X}$, quality level supervision is needed. Since the quality of the counterfactual sample $\tilde{\mathbf{X}}$ is expected to be $y + \Delta y$ (y is the label of $\mathbf{X}$), a straightforward way is to send the quality attribute $\tilde{\mathbf{Q}}$ derived from the pos-

terior $P_\phi(Q|X=\tilde{\mathbf{X}})$ to quality classifier $f_q(\cdot)$ and let the prediction be as close as $y+\Delta y$. However, the generated samples will affect the learning of $f_q(\cdot)$. Intuitively, we hope the counterfactual samples should meet the expectations established by all real samples, rather than change the rules. So instead of directly applying $f_q(\cdot)$, we adopt a quality classifier $f_{qf}(\cdot)$ that copies all parameters from $f_q(\cdot)$. In this way, $f_{qf}(\cdot)$ plays a similar role to the GAN's discriminator [62], which regularizes the generation process of counterfactual samples. The $f_{qf}(\cdot)$ is applied to both $\tilde{Q}$ and $\mathbf{Q}^*$ to guarantee the generation process as shown in Fig. 5.3.

5.3.3 Cycle-Counterfactual Framework

The same overall quality score can be used to measure different actions and also the same actions with intra-variance (e.g., different athletes), where the key of the score estimation is the precise sub-action modeling. However, the prior that the quality is the same inferior to the modeling of sub-action quality, and our goal is to prevent the model from overfitting on the same quality but diverse sub-actions. Here, we propose a cycle-consistency training strategy based on counterfactual generation.

Forward Counterfactual. The framework first encodes the input video feature $\mathbf{X}$ to the quality attribute $\mathbf{Q}$ and context attribute $\mathbf{Z}$ via posterior $P_\psi(Q|X=\mathbf{X})$ and $P_\phi(Z|X=\mathbf{X})$. Then, the framework generates counterfactual sample $\hat{\mathbf{X}} = X_{\mathbf{Z},\mathbf{Q}^*}$ from $\mathbf{X}$, $\mathbf{Q}^*$ is obtained by update its corresponding sub-actions' quality. For instance, $\hat{\mathbf{q}_k} = \mathbf{q}_k + f_{embed}(\mathbf{q}_k + \Delta y)$, and $\hat{\mathbf{q}_k} \in \mathbf{Q}^*$. With the help of the Δy, the quality of counterfactual samples are different from the original one.

Backward Counterfactual. Based on the counterfactual sample $\tilde{\mathbf{X}}$, the quality attribute $\tilde{\mathbf{Q}} \sim P_\psi(Q|X=\tilde{\mathbf{X}})$ can be reversed via update its corresponding sub-actions' quality using the same relative score embedding as in forward process. For instance, $\tilde{\mathbf{Q}}^* = f_{embed}(\tilde{\mathbf{Q}}, k, -\Delta y)$. The reverse operation is applied separately and independently for each sub-action, where an individual action consists of K sub-actions, resulting in K different reversed counterfactual samples. Hence, the videos $\{\tilde{\mathbf{X}}'^{(k)}\}_{k=1}^{K}$ can be obtained with same overall quality score compared with $\mathbf{X}$ but the quality of corresponding sub-actions are different.

Cycle Consistency/Inconsistency Loss. After the forward and backward coun-

terfactual operation, it can be expected that one reversed counterfactual sample $\tilde{\mathbf{X}}'^{(o)}$ should be the same as the original $\mathbf{X}$, which results in no sub-action quality change. Then other backward samples (exclude $\tilde{\mathbf{X}}'^{(o)}$) should be far away from original $\mathbf{X}$, although they have same overall quality score compare with original $\mathbf{X}$. The loss function is expressed as:

$$\mathcal{L}_{cst} = -\log \frac{\exp\left(\mathbf{X} \cdot \tilde{\mathbf{X}}'^{(o)}/\tau\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{X} \cdot \tilde{\mathbf{X}}'^{(k)}/\tau\right)} + \|\mathbf{X} - \tilde{\mathbf{X}}'^{(o)}\|_2 \ , \tag{5.1}$$

where τ is a temperature hyper-parameter. The $\mathcal{L}_{cycle}$ pull close the distance between $\mathbf{X}$ and positive backward counterfactual sample $\tilde{\mathbf{X}}'^{(o)}$ and push away distance between $\mathbf{X}$ and negative backward samples. The proposed cycle counterfactual explicitly enhances the part-level (i.e., sub-actions) quality representation. Although the sub-action quality remains unknown, the relative quality among sub-actions across different samples $\{\tilde{\mathbf{X}}'^{(k)}\}_{k=1}^{K}$ is known. We then use MSE loss to regularize the part-level quality attribute to satisfy the relative score as follows:

$$\mathcal{L}_{rel} = \sum\nolimits_{k=1\setminus\{o\}}^{K-1} \|f_{rel}(\tilde{\mathbf{q}}_k^{(k)}, \mathbf{q}_k) - \Delta y\|_2 \ , \tag{5.2}$$

where f_{rel} is a regression head that predict the relative score between $\tilde{\mathbf{q}}_k^{(k)}$ and $\mathbf{q}_k$. In this way, the model explicitly learn the difference between part-level quality attribute, which improve the overall quality assessment. The overall cycle loss can be expressed as $\mathcal{L}_{cycle} = \mathcal{L}_{cst} + \mathcal{L}_{rel}$.

5.3.4 Feature Disentanglement

In our framework, *quality attribute* $\mathbf{Q}$ is designed to learn quality related information while *context attribute* $\mathbf{Z}$ contains quality-agnostic information such as sub-action type and camera view. However, the quality classifier $f_q(\cdot)$ and context classifier $f_z(\cdot)$ may still guide the model to aggregate desired information for $\mathbf{Q}$ and $\mathbf{Z}$, overlapping information can cause the ambiguity/redundancy during the model training. Besides, the model need to learn relative score across different sub-action, the context information (e.g., sub-actions class) remains in quality attribute will confuse

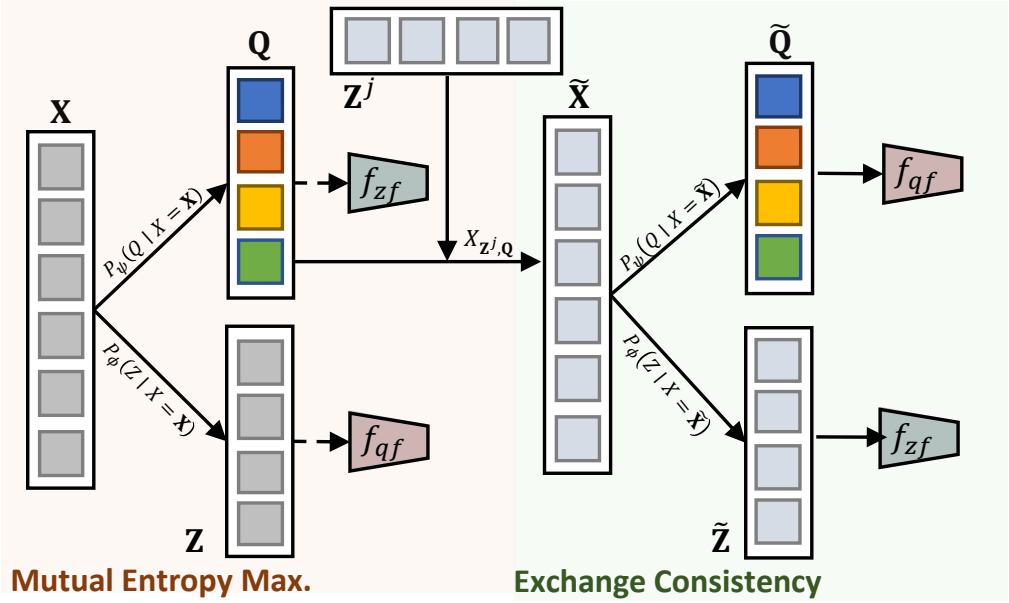


Figure 5.4: Overview of the disentangle module. The quality attribute $\mathbf{Q}$ is sent to the fixed context classifier $f_{zf}(\cdot)$ (copied parameters from $f_z(\cdot)$), while the context attribute $\mathbf{Z}$ is sent to the fixed quality classifier. The mutual entropy maximization loss maximizes the cross-entropy loss, as shown in dashed arrows. The exchange consistency loss force $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{Z}}$ derived from counterfactual $\tilde{\mathbf{X}} = X_{\mathbf{Z}^j,\mathbf{Q}}$ to keep their original quality and context information by minimize their cross-entropy loss as shown in solid arrows.

the model. Furthermore, to guarantee the counterfactual-faithful generation [195], the group disentanglement among $\mathbf{Q}$ and $\mathbf{Z}$ is desired to be satisfied. Hence, the disentanglement between $\mathbf{Q}$ and $\mathbf{Z}$ is as follows. The process of disentangling is shown in Fig.5.4.

Mutual Entropy Maximization. We maximize the cross-entropy loss of predicted quality when $\mathbf{Z}$ is fed into the quality classifier $f_q(\cdot)$. The cross-entropy loss of context information is forced to be maximized when $\mathbf{Q}$ is fed to the context classifier $f_z(\cdot)$. In this way, the quality agnostic features are encouraged to be irrelevant to the quality labels, and vice versa. The loss function is as follows:

$$\mathcal{L}_{mem} = -\left[\ell\left(y, f_q\left(\mathbf{Z}\right)\right) + \ell\left(y^z, f_z\left(\mathbf{Q}\right)\right)\right] \ , \tag{5.3}$$

where $\ell(\cdot)$ is cross-entropy loss. y and y^z indicate the quality label and context label of current input video.

Exchange Consistency. To ensure the counterfactual generated samples are al-

ways reasonable, i.e., contain the original quality and context information, the model training process is also regularized using the quality attribute $\tilde{\mathbf{Q}}$ and the context attribute $\tilde{\mathbf{Z}}$ derived from the counterfactual samples $\tilde{\mathbf{X}} = X_{\mathbf{Z}^j,\mathbf{Q}}$($\mathbf{Z}^j$ is the context attribute derived from j^{th} training sample) by:

$$\mathcal{L}_{ec} = \ell\left(y, f_q\left(\tilde{\mathbf{Q}}\right)\right) + \ell\left(y_j^z, f_z\left(\tilde{\mathbf{Z}}\right)\right) \ , \tag{5.4}$$

where y_j^z is the context label of j^{th} sample in training set. The model force the quality information remain after changing of context information, which preserve the quality information from context related attributes and vice versa. The overall disentangle loss can then be expressed as $\mathcal{L}_{dgl} = \mathcal{L}_{mem} + \mathcal{L}_{ec}$

5.3.5 Overall Optimization

In this subsection, we list all the objective functions needed for model optimization during training. As for quality, we calculate the mean value based on the quality class distribution and adopt MSE to minimize the distance between the predicted quality and ground-truth score. To be specific, the quality scores are discretized into category labels $y = \{0, 1, ..., C-1\}$. We convert the labels to a label vector $\boldsymbol{y} \in \mathbb{R}^C$. Given a quality classifier $f_q(\cdot)$, the loss function of quality regression can then be expressed as:

$$\mathcal{L}_{quality} = \|\boldsymbol{y}^T \cdot Softmax(f_q(\mathbf{Q})) - y\|_2 \ . \tag{5.5}$$

In counterfactual process, the counterfactual expectation adopts f_{qc} as classifier. the prediction function can be expressed as:

$$\mu^*(\mathbf{Q}) = \boldsymbol{y}^T \cdot Softmax(f_{qc}(\mathbf{Q})) \ . \tag{5.6}$$

During forward counterfactual, the $\mathbf{Q}^*$ and the quality attribute $\tilde{\mathbf{Q}}$ derived from the counterfactual sample are expected to express quality score $y + \Delta y$. So the

counterfactual expectation loss during forward process is:

$$\mathcal{L}_f = \|\mu^*(\mathbf{Q}^*) - y - \Delta y\|_2 + \|\mu^*(\tilde{\mathbf{Q}}) - y - \Delta y\|_2 \ . \tag{5.7}$$

Since we have K backward samples given a single input, the backward counterfactual expectation is derived as:

$$\mathcal{L}_b = \sum_{k=1}^{K}(\|\mu^*(\tilde{\mathbf{Q}}^{*(k)}) - y\|_2 + \|\mu^*(\tilde{\mathbf{Q}}'^{(k)}) - y\|_2) \ . \tag{5.8}$$

Then, the overall loss function during training can be expressed as:

$$\begin{aligned} \mathcal{L}_{train} = \mathcal{L}_{quality} + \lambda_1 \cdot \mathcal{L}_{cycle} + \lambda_2 \cdot \mathcal{L}_{dgl} \\ + \lambda_3(\mathcal{L}_f + \mathcal{L}_b). \end{aligned} \tag{5.9}$$

In summary, $\mathcal{L}_{cycle}$ helps the model to explore sub-action quality, while counterfactual expectation loss $\mathcal{L}_f$ and $\mathcal{L}_b$ regularize the the generation process. Disentangle loss $\mathcal{L}_{dgl}$ further enhanced the generation. The above functions together helps to optimize the quality loss $\mathcal{L}_{quality}$.

5.4 Experiments

5.4.1 Experimental Setup

Datasets. We perform experiments on three public benchmarks: MTL-AQA [131], AQA-7 [130], and JIGSAWS [170]. MTL-AQA contains 1412 fine-grained samples collected from 16 different events with different views. AQA-7 contains samples from seven different action categorie with 803 training videos and 303 testing videos. JIGSAWS is a surgical activities dataset that contains three tasks, namely Suturing (S), Needle Passing (NP), and Knot Tying (KT). Following prior work [1], we the Spearmans rank correlation and relative L2 distance(R-ℓ_2) as our evaluation metrics.

Implementation Details. We adopt the I3D model pretrained on Kinetics [26] as our backbone for local spatial-temporal feature extraction. The model is trained on 4 RTX 3090 GPUs, the total batch-size is set to 20. We use Adam as our optimizer and

Table 5.1: Performance comparison on MTL-AQA dataset. 'w/o DD' means that training and test processes do not utilize difficulty degree labels, 'w/ DD' means experiments utilizing difficulty degree labels.

Method (w/o DD)	Sp. Corr.	R-$\ell_2(\times 100)$
Pose+DCT [134]	0.2682	-
C3D-SVR [135]	0.7716	-
C3D-LSTM [135]	0.8489	-
MSCADC-STL [131]	0.8472	-
C3D-AVG-STL [131]	0.8960	-
MSCADC-MTL [131]	0.8612	-
C3D-AVG-MTL [131]	0.9044	-
USDL [139]	0.9066	0.654
CoRe [1]	0.9341	0.365
TSA-Net [140]	0.9422	-
Ours	**0.9468**	**0.3362**
Method (w/ DD)	Sp. Corr	R-$\ell_2(\times\ 100)$
USDL [139]	0.9231	0.468
MUSDL [139]	0.9273	0.451
CoRe [1]	0.9512	0.260
Ours	**0.9613**	**0.232**

the learning rate is set to 1×10^{-4} for the backbone, video encoder, video decoder and encoder. The learning rate for the classification heads for both quality and context are set to 1×10^{-3}. The feature dimension is set to 512 for the transformer block. The feature dimension for both quality attribute and context attribute is set to 128. We set the number of sub-action K to 5 in our experiment. The λ_1, λ_2, λ_3 are set to 0.2, 0.2 and 0.1 respectively during optimization. We adopt transformer decoder similar to [191] to model our posterior $P_\phi(Z|X)$ and $P_\psi(Q|X)$. As for conditional distribution $P_\theta(X|Q,Z)$, we adopt transformer decoder with T learnable positional embeddings. As for the data-preprocessing on AQA-7 and MTL-AQA datasets, we follow the pipeline from [191] that sample 103 frames for all videos. We then split the frames into 20 overlapping clips, each segment containing 8 consecutive frames. As for the JIGSAWS dataset, we uniformly sample 160 frames following [139] and divide them into 20 non-overlapping clips as input of the I3D backbone.

Table 5.2: Performance comparison on AQA-7 dataset.

Sp. Corr	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. Corr.
Pose+DCT [134]	0.5300	0.1000	-	-	-	-	-
ST-GCN [181]	0.3286	0.5770	0.1681	0.1234	0.6600	0.6483	0.4433
C3D-LSTM [135]	0.6047	0.5636	0.4593	0.5029	0.7912	0.6927	0.6165
C3D-SVR [135]	0.7902	0.6824	0.5209	0.4006	0.5937	0.9120	0.6937
JRG [138]	0.7630	0.7358	0.6006	0.5405	0.9013	0.9254	0.7849
USDL [139]	0.8099	0.7570	0.6538	0.7109	0.9166	0.8878	0.8102
CoRe [1]	0.8824	0.7746	0.7115	0.6624	0.9442	0.9078	0.8401
TSA-Net [140]	0.8379	0.8004	0.6657	**0.6962**	0.9493	0.9334	0.8476
Ours	**0.8993**	**0.8082**	**0.7368**	0.6956	**0.9552**	**0.9482**	**0.8738**
R-ℓ_2($\times$100)	Diving	Gym Vault	BigSki.	BigSnow.	Sync. 3m	Sync. 10m	Avg. R-ℓ_2
C3D-SVR [135]	1.53	3.12	6.79	7.03	17.84	4.83	6.86
USDL [139]	0.79	2.09	4.82	4.94	0.65	2.14	2.57
CoRe [1]	0.64	1.78	3.67	3.87	0.41	2.35	2.12
Ours	**0.49**	**1.72**	**2.75**	**3.19**	**0.29**	**1.27**	**1.62**

5.4.2 Comparison with State-of-the-art Methods

We juxtapose our method's performance with leading algorithms across three pivotal benchmark datasets, as showcased in Tab. 5.1, Tab. 5.2, and Tab. 5.3.

MTL-AQA Dataset: On the **MTL-AQA** dataset, we meticulously curated our experiments, adopting two distinct settings, concurring with established paradigms [1]. Pertinently, the MTL-AQA dataset bequeaths a label indicative of the degree of difficulty. The quality score for each video is a resultant of the raw score being multiplied by this nuanced difficulty coefficient.

Under the ambit of the 'w/o DD' setting, both training and test epochs were abstinent from employing the difficulty degree labels. Conversely, during the 'w/ DD' regime, we deduced the raw score, and subsequently juxtaposed it with the difficulty to derive the final quality metric. Intriguingly, our algorithm evinces dominion over prevalent methods in both paradigms. As elucidated in Tab. 5.1, within the 'w/ DD' milieu, our strategy garners a Spearman's Correlation (Sp. Corr.) of 0.9613 and a Root Mean Squared Error (R-ℓ_2) of 0.2322, thereby overshadowing the tree-based CoRe [1] mechanism. A salient observation underscores that our performance exhibits only marginal advancement over the recently innovated TPT [191]. Given the contrastive architecture of TPT, which is resource-intensive during training and entails sampling copious exemplars during testing, comparisons with TPT

were intentionally omitted to maintain a fair playing field.

In the landscape of 'w/o DD', our method secures 0.9451 in Sp. Corr. and 0.3222 in R-ℓ_2, thereby surpassing stalwarts like CoRe, TSA-Net [140], and TPT [191]. It warrants mention that TSA-Net is reliant on an external VOT tracker [183] to pinpoint human coordinates, a tangent distinct from the quintessential challenge of gauging sub-action quality with only video level labels. In light of this, it's plausible to anticipate an enhancement in our model's efficacy by amalgamating the attention module proposed in [140].

Scrutiny on AQA-7 Dataset: In the context of the **AQA-7** dataset, our method clinches the pinnacle of performance in 5 out of the 7 categories, while delivering comparable prowess in the remaining ones. A comprehensive glimpse at Tab. 5.2 reveals that our approach outpaces CoRe by a significant 3.37 Corr.($\times$100) and TSA-Net by 2.58 Corr.($\times$100). Moreover, our method manifests a minimal R-ℓ_2 of 1.62($\times$100), reinforcing the robustness of our Cycle-counterfactual paradigm.

Insights from JIGSAW Dataset: Venturing into the compact **JIGSAW** dataset, we adhered to a 4-fold cross-validation for each category, aligning with antecedent methodologies [1, 139]. Our architecture demonstrates an average correlation of 0.90 and an R-ℓ_2 of 3.567, unequivocally setting a new gold standard in AQA evaluation.

Our proposed method's advantage stems from its unique architectural choices. While many models require heavy computational resources or additional modules, our method efficiently leverages the Cycle-counterfactual paradigm, allowing for precise sub-action quality discernment. This efficiency is evident when compared to models like TPT [191], which demands resource-intensive training. Thus, our architecture's simplicity and effectiveness position it favorably against current AQA evaluation methods.

5.4.3 Ablation Study

In this subsection, we perform ablation studies to evaluate the effectiveness of our proposed components and designs. All of our ablation studies are performed on MTL-AQA dataset under 'w/o DD' setting. We build a baseline network that first

Table 5.3: Performance comparison on JIGSAW dataset.

Sp. Corr.	S	NP	KT	Avg.
ST-GCN [181]	0.31	0.39	0.58	0.43
TSN [182]	0.34	0.23	0.72	0.46
JRG [138]	0.36	0.54	0.75	0.57
USDL [139]	0.64	0.63	0.61	0.63
MUSDL [139]	0.71	0.69	0.71	0.70
CoRe [1]	0.84	0.86	0.86	0.85
Ours	**0.89**	**0.90**	**0.91**	**0.90**
R-ℓ_2	S	NP	KT	Avg.
CoRe [1]	5.055	5.688	2.927	4.556
Ours	**2.693**	**5.021**	**2.989**	**3.567**

Table 5.4: Ablation study of different components on MTL-AQA dataset.

Method	Cls.	C.Cycle	C.Exp.	Dgl.	Sp. Corr.	R-ℓ_2
Baseline	×	×	×	×	0.9501	0.2835
	✓	×	×	×	0.9523	0.2761
	✓	✓	×	×	0.9532	0.2631
	✓	✓	✓	×	0.9584	0.2394
Ours	✓	✓	✓	✓	**0.9613**	**0.2322**

encode the input feature into part-level quality attribute $\mathbf{Q}$, then pool $\mathbf{Q}$ and predict the quality score based on a regression model.

Different Model Components. In this work, we propose a novel cycle-counterfactual frame to exploit the sub-action quality explicitly without the explicit annotation. We perform experiments to show the effectiveness of each component in the proposed framework as shown in Tab. 5.4. We can observe that with only cycle-counterfactual training (based on quality classifier), the performance only improves marginally from 0.9524 Corr. to 0.9551 Corr.. With counterfactual expectation, the performance is significantly improved, demonstrating the importance of semantic-level supervision besides the cycle-consistency loss. The Disentangle module further improves the performance, showing that the purity of quality attribute is necessary during counterfactual generation across different sub-actions.

Different Relative Score Embedding Strategy. We adopt quality class centers

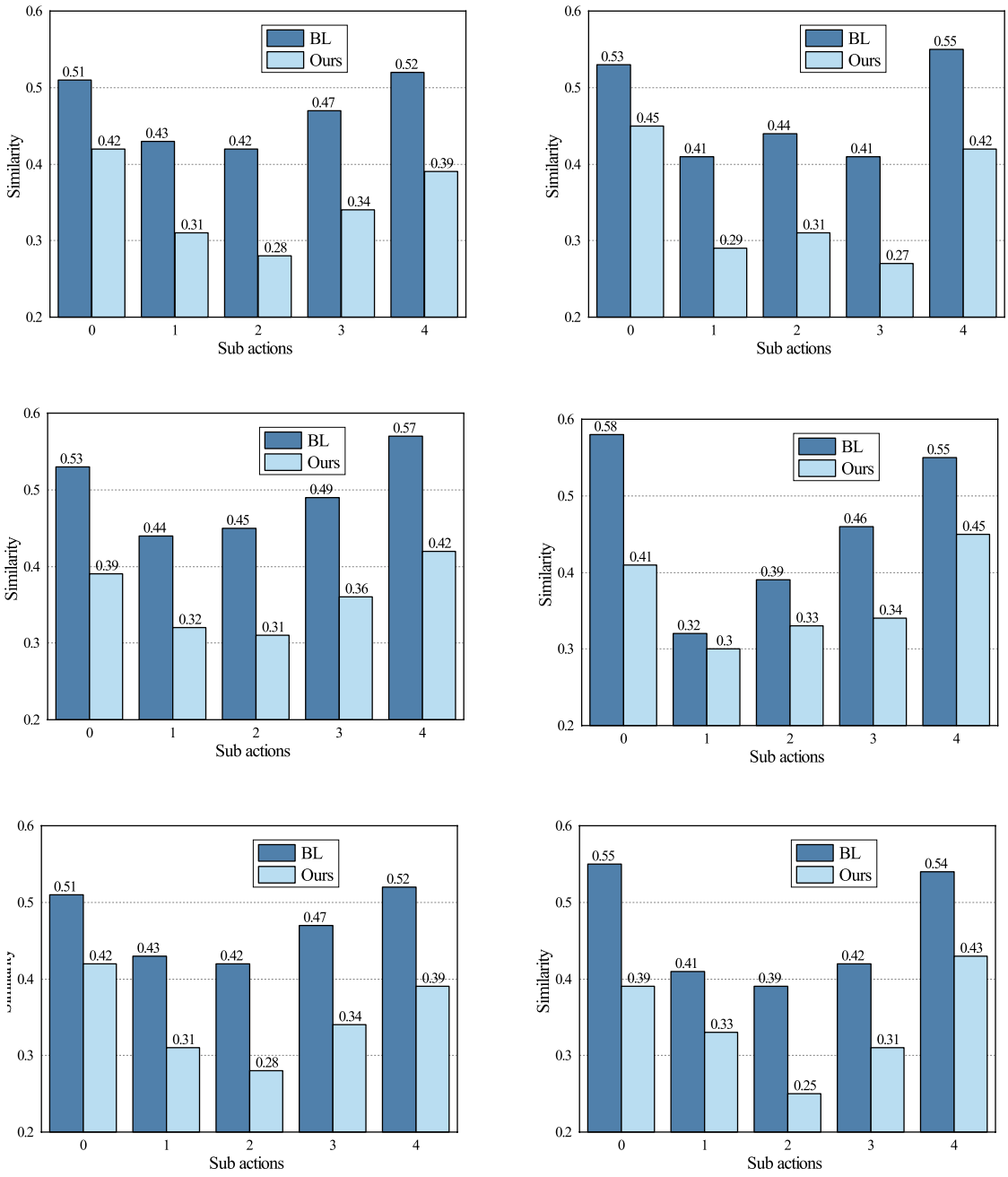


Figure 5.5: Here we select 6 non-overlapping groups from test set. The difference between the maximum and minimum gt score within each group is less than 5. Each sub-figure shows the similarity(y-axis) of sub-actions quality(x-axis) across samples of the group. We can observe that for each sub-action, the quality attribute is less similar given similar overall scores under our model.

as prior to better modeling the variance of relative score representation based on different initial scores. To verify the importance of such prior learning strategy, we replace our relative score embedding by 1) directly mapping the scalar-formed relative score to a vector that represents the relative embedding; 2) Like word em-

bedding [218], we can treat relative scores (-1, -0.5, 0.5 ,1), each score corresponds to a learnable embedding that represents the semantic meaning. As shown in Tab. 5.5, the performance drops from 0.9613 Corr. to 0.9578, which indicates the effectiveness of our relative embedding strategy.

Table 5.5: Ablation study of relative score embedding strategies on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
MLP	0.9552	0.2492
Word embedding	0.9587	0.2427
Class center prior(ours)	**0.9613**	**0.232**

Quality of Counterfactual Samples and Effect of Counterfactual Expectation. Tab. 5.6 shows the performance of the positive backward counterfactual samples given different generation regularizations. The last row in Tab 5 shows the performance under counterfactual expectation, which adopts fixed quality classifier trained using real samples. The performance of counterfactual samples only drops a negligible margin compared to that from real samples, demonstrating the high quality of the counterfactual samples. The counterfactual expectation aims to use the knowledge learned from real samples to guide the generation. To verify the effectiveness of the counterfactual expectation, we replace the fixed quality classifier with: 1) shared quality classifier for both real samples and counterfactual samples. 2) fixed quality in the first half training process and shared quality classifier during the rest of the training process. As shown in Tab. 5.6, the shared quality classifier severely impact performance. Although the performance of the hybrid strategy is much better than the shared one, the fixed classifier still gains the best performance, which illustrates the effectiveness of our proposed counterfactual expectation.

Effectiveness of the Disentangle Module. To verify the effectiveness of our proposed disentangle losses, we first replace the disentangle loss by maximizing the cosine similarity between the quality attribute and context attribute. As shown in Tab. 5.7, although the result of cosine similarity is better than without disentanglement (as shown in Table 4), it is not as good as the results of only applying mutual

Table 5.6: Ablation study of effect of counterfactual expectation on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
learnable $f_q(\cdot)$	0.9529	0.2789
hybrid $f_q(\cdot)$	0.9568	0.2475
Fixed $f_q(\cdot)$ (ours)	**0.9577**	**0.2438**

entropy maximization. The possible reason is that the mutual entropy maximization widens the distance between the quality attribute and context attribute at the semantic level, while semantic information is ignored by cosine similarity. The exchange consistency further improves the performance, which illustrates the strength of the proposed disentangle module.

Table 5.7: Ablation study of effect of Disentangle module on MTL-AQA dataset.

Method	Sp. Corr.	R-ℓ_2
cosine similarity	0.9588	0.2412
mutual entropy max.	0.9596	0.2375
exchange consistency	0.9591	0.2391
Ours	**0.9613**	**0.232**

5.4.4 Qualitative Results

In Fig.5.5, we select 6 groups of test data from MTL-AQA dataset, each group has similar quality score. We can see in all sub-figures, the similarity of sub-action quality attributes trained under our model is much smaller compared with quality attributes trained using the baseline model. Although the baseline model divide the video feature into sub-actions, the lack of sub-action supervision still prevent the baseline model from correctly model the quality of sub-actions. This indicates that our proposed cycle-counterfactual framework is capable of exploiting sub-action quality with only video level supervision.

5.4.5 Impact of Dataset Selection and Generalization

The cycle-counterfactual framework's efficacy is intimately linked to the nature of the chosen dataset. The diversity of a dataset plays a pivotal role in bolstering the model's robustness; a limited or homogenous dataset may inadvertently lead the model to overfit and consequently diminish its ability to generalize across varied action types. Additionally, the sheer size of the dataset matters. A capacious dataset empowers the framework to discern and learn intricate relationships between actions and their quality. Conversely, a diminutive dataset might fail to capture all action nuances, leading to suboptimal performance on unseen data. It's worth noting that while the framework might excel on one dataset, it doesn't guarantee equivalent success on another. Rigorous evaluations, such as cross-dataset validations, provide a more comprehensive insight into the model's generalization prowess. In summation, a judicious dataset selection is imperative for both optimal performance and broad generalization to novel actions or sub-actions.

5.5 Discussion and Limitation

The cycle-counterfactual framework, though displaying promising outcomes, is not without its constraints. One of the primary concerns is its generative assumptions. When the quality of a sub-action is altered, it could inadvertently impact other sub-actions, which casts doubts on the authenticity of the generated counterfactual samples. Furthermore, the framework's performance hinges significantly on the diversity of the training data it's exposed to. A dataset with limited variations can lead the model to produce counterfactuals that might not resemble realistic scenarios.

Another challenge emerges when considering scalability. The framework's ability to maintain its efficiency, especially with larger datasets, remains largely unprobed, raising potential issues about its capacity for consistent counterfactual generation. Lastly, the absence of ground truth annotations for sub-action scores creates a hurdle. Without these, ascertaining the accuracy and reliability of the counterfactuals becomes a formidable task. While the cycle-counterfactual framework offers a novel

approach, these limitations suggest areas that would benefit from further exploration and refinement to ensure broader and steadfast applicability.

5.6 Conclusion

In this chapter, we proposed a novel cycle-counterfactual framework to explicitly explore the quality of sub-actions with only video-level supervision provided, resulting in a more robust action quality assessment system. Through the cycle-counterfactual process, the framework generated counterfactual samples with the same overall score but different sub-action quality representations. The framework can then explore the quality of sub-actions by make comparison among the counterfactual samples and the fact samples. To further guarantee the faith of counterfactual generation, we proposed a disentangling module. As a result, our method is able to outperform existing state-of-the-art models by a considerable margin.

CHAPTER 6

Discriminative Latent Semantic Graph for Video Captioning

As we advance in our exploration of Interaction-level Action Understanding, each chapter has sought to unravel the layers of human interpretation in relation to visual data. Chapter 3 laid the foundation by probing into action understanding through the lens of human consensus, using video summaries as the medium. This gave us an insight into a collective interpretation of actions, drawing from a broader perspective. Meanwhile, Chapters 4 and 5 took a more structured approach, diving into action understanding steered by specific human-imposed rules. These rules, while giving a structured insight, might not always encapsulate the depth and intricacies of real-world actions.

With Chapter 6, we are venturing into a more advanced terrain. Here, we're exploring action understanding via the intricacies of human language. This paradigm shift allows us to capture the nuances, emotions, and contexts which might have been overlooked in the more rigid rule-based evaluations. Human language, with its rich lexicon and semantics, offers a broader and deeper canvas to decode and describe actions, making it a pivotal tool in achieving advanced interaction-level action understanding.

While traditional encoder-decoder frameworks have been instrumental in video

captioning, they often fall short in capturing object-level interactions and the intricacies of spatio-temporal data. To address this, our primary contributions in this chapter encompass:

Enhanced Object Proposal: Through our Conditional Graph, we seamlessly integrate spatio-temporal information into latent object proposals. This ensures actions and their interplays are represented comprehensively.

Visual Knowledge: The Latent Proposal Aggregation method we introduce dynamically selects visual words with superior semantic relevance. This innovation aids in generating descriptions that are both accurate and profound in context.

Sentence Validation: Our Discriminative Language Validator ensures that generated captions truly resonate with the intended semantic depth. It refines and authenticates the narrative, preserving the essence of pivotal semantic concepts.

Trials on MVSD and MSR-VTT datasets have reinforced the potential of our approach. Metrics such as BLEU-4 and CIDEr notably underscore our advancements, suggesting that integrating natural language intricacies can indeed propel us closer to genuine interaction-level action comprehension.

6.1 Introduction

With the tremendous growth of video materials uploaded to various online video platforms, *e.g.* YouTube, research in automatic video captioning has received increasing attention in recent years. Thorough video caption can leads to huge practical impacts, *e.g.* content-based video retrieval and recommendation. Despite the remarkable progress of computer vision and natural language processing in video analysis and language understanding, video captioning is still a very challenging task. The task requires to explore not only complex object interactions and relationships at frame-level, but also high-level story-line from video sequence. Such a task can be seen as a leap from the recognition to comprehension level.

One of the main challenges of video captioning is that there is no explicit mapping between video frames and words in captions. The model needs to extract summarize visual words at a much higher semantic level. Figure 6.1 illustrate an

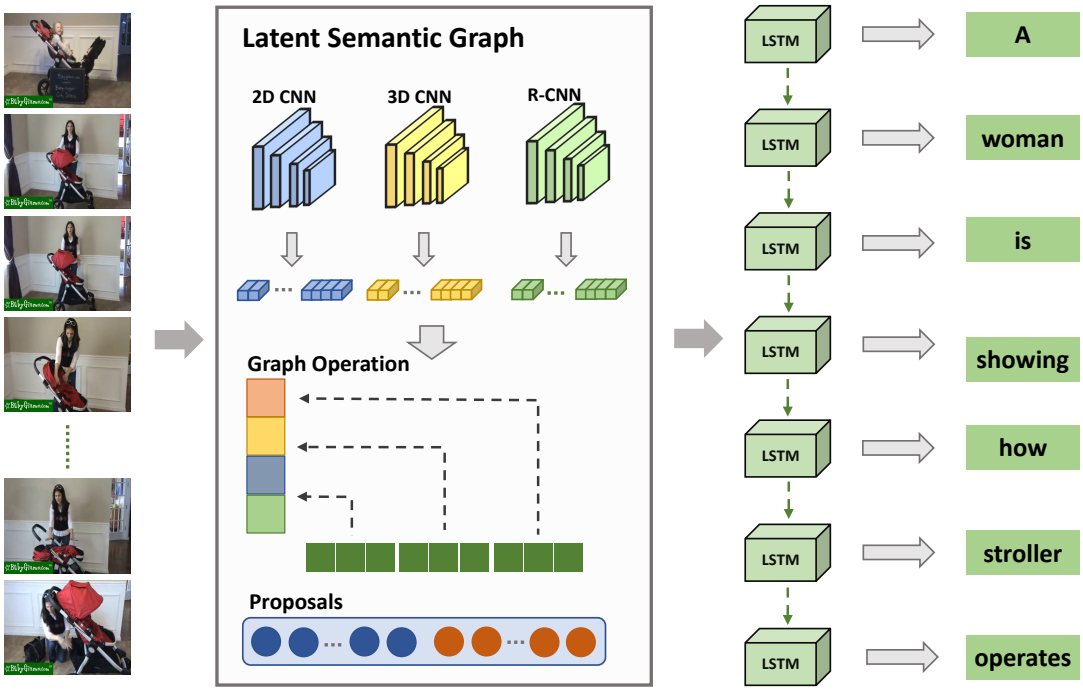


Figure 6.1: An illustration of the video captioning task. The key challenge is that there is no explicit mapping between video frames and the captions. The model needs to jointly consider 2D-CNN, 3D-CNN, and Object proposals from R-CNN and extract high-level semantic visual words to construct a compact caption.

intuitive example of video captioning. From a human perspective, we can interpret the overall process into several sub-tasks: 1) to detect and recognize the main objects in the video, *i.e.,* "woman" and "stroller"; 2) to infer the action imposed to these objects, *i.e.,* "showing" and "operates"; 3) organizing the contents into a sentence with grammatical structures, *i.e.,* "A woman is showing how stroller operates". Early studies typically adopt encoder-decoder frameworks [143–145] that model video captioning as a machine translation task. These methods focus on modeling the static frame and object features and temporal changes among embedding. To overcome the drawback of embedding-based frameworks, the recent rise of graph neural networks (GNNs) has shown particular advantages in modeling relationships between objects [219, 220]. However, applying GNNs into the video captioning task is not a trivial thing. Previous GNNs are mainly built on object features without jointly considering the frame-based spatio-temporal contexts in the entire video sequence. The other challenge is that the output caption needs to ensure the readability and grammatical structure rather than producing a list of discrete concept.

To examine whether the expression of a sentence is natural or not, [220] utilizes a generative adversarial network (GAN) to control the fidelity of generated sentences. However, video captioning requires a finer-level of supervision to distinguish the real/fake sentence distribution as well as ensure the grammatical correctness of a sentence.

The above challenges motivates us to design a new framework for video captioning with three sub-tasks, *i.e.* **Enhanced Object Proposal, Visual Knowledge,** and **Sentence Validation**. First, Fusion is about extracting the spatio-temporal contexts from video frames and incorporating such information in the object entities. Note that the number of frames and object proposals in videos is far more than that of words in captions. Therefore, the second task visual knowledge summary aims to reduce such duplicated and redundant proposals into more compact visual words. Such high-level visual words should be easier encoded by a sequential model to produce a caption. The last sentence validation task aims to examine both the fidelity and the readability of the generated caption.

According to the above motivations, we design a **Discriminative Latent Semantic Graph (D-LSG)** framework with the following insights: 1) **Graph** model for feature fusion from multiple base models *e.g.* 2D/3D CNN and R-CNN remain unexplored. These features are often heterogeneous in data distribution, dimensions, and structure. 2D CNN represents the frame contents while 3D CNN extract the temporal frame changes. We consider such frame-level information as the conditions of all region-level object proposals. Therefore, the conditional graph is not in the traditional form of semi-positive indefinite affinity matrix. 2) **Latent Semantic** refers to the higher-level semantic knowledge that can be extracted from the enhanced object proposals. Rather than incorporating external auxiliary knowledge graph as [221], our key idea is to construct a dynamic graph that connects enhanced object proposals with randomly initialized nodes. In other words, the great volume of enhanced object proposals are summarized into high-level visual knowledge via the dynamic graph. 3) **Discriminative** module is designed as a plug-in language validator. Generated and ground truth captions can be reconstructed into visual knowledge so as to compare with that extracted from the enhanced object propos-

als. We adopt the Multimodel Low-rank Bi-linear (MLB) [222] pooling as metrics to provide finer-level supervision to carry out the sentence validation task.

In summary, this chapter's focus on video captioning aligns directly with the thesis's main goals. In Chapter 3, we tackled action understanding using human consensus via video summary. Chapters 4 and 5 then shifted the focus to specific human rules for action understanding. Here, in Chapter 6, we're taking the next logical step: using human language to gain deeper insights into action understanding. By addressing the challenges of video captioning and introducing the D-LSG framework, we're pushing the boundaries of interaction-level action understanding, a core objective of this thesis. Our contributions include:

- To identify Enhanced Object Proposal, Visual Knowledge, and Sentence Validation sub-tasks in a unified framework for future video summarization tasks.
- A Condition Graph Operation is proposed to enhance region-level object proposal representations with spatio-temporal information of base features of video frames.
- Latent Proposal Aggregation with a dynamic graph model is proposed to compress enhanced object proposals into visual knowledge with higher semantic meanings in a latent space.
- A Discriminative model is plug-in as a validation network that can distinguish generated sentences from ground truth captions and encourage the generated captions to be more content-relevant and semantic-richer.
- Quantitative and qualitative experiments on two datasets, MSVD [223] and MSR-VTT [163], demonstrate significant performance boost on all evaluation while achieving significant performance improvement on CIDEr.

6.2 Related Work

Discriminative modeling. So far, many studies have investigated image and video captioning using discriminative modeling. The work of [84] applied a discriminator

model to distinguish ground truth captions from generated captions, which relies on the Gumbel-Softmax approximation [85], whereas [69] utilizes the policy gradient and their discriminator focuses on caption naturalness and image relevance. The work of [224] designed a multi-discriminator system that encourages better multi-sentence video description. However, discriminative modeling for caption generation suffers from stability issues and requires pre-trained generators. Thus, instead of using the Gumbel-Softmax or policy gradient based method, we propose a semantic relevance discriminative graph based on Wasserstein gradient penalty [225], which can directly feed output and graph-based latent semantic concepts from the generator and do not need to pre-train the generator. The proposed discriminative encoder can be plug into an end-to-end model to reconstruct captions into visual knowledge so that the fidelity and sentence structure can be validated and key semantic entities can be preserved.

Graph Neural Networks. Efficient and practical neural network algorithms for processing graph-structured data have become one of the most important machine learning subareas. Recently, pioneer work [219, 221, 226] have tried to adopt graph neural networks to video captioning. For example, Pan *et al.* [226] proposed a spatio-temporal graph model for video captioning that exploits object interactions in space and time. Another graph based video captioning research [219] proposed an object relational graph based encoder, which captures more detailed object interaction features to enrich visual representation. However, the weights they used to summarize object features in temporal space are the same as the frame features, which may lead to degraded temporal information in summarized object features. In comparison, our conditional graph jointly consider object, contexts, and motion information at both region and frame levels. The work of [221] proposed a joint commonsense and relation reasoning method that applies auxiliary databases for pre-training knowledge graphs as prior knowledge for image and video captioning. Our dynamic graph in the Latent Proposal Aggregation module is able to extract high-level latent semantic concepts without an external dataset for training.

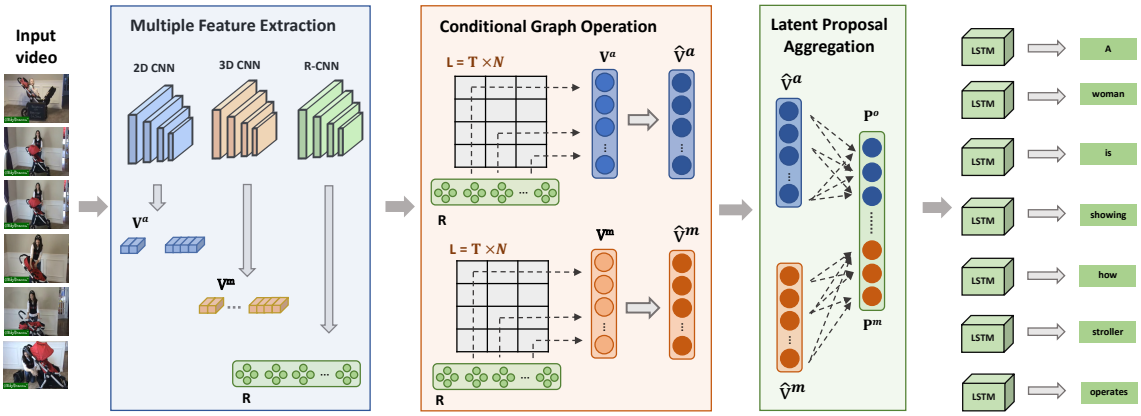


Figure 6.2: Overview of proposed LSG framework. Base features from 2D/3D CNN and R-CNN provides object and contexts features at frame and region levels. Conditional Graph Operation is applied to appearance and motion channels to compute Enhanced Object Proposals $\hat{V}^a$ and $\hat{V}^m$. T frames of $\hat{V}^a$ and $\hat{V}^m$ are selected into K Visual Knowledge before LSTM captioning.

6.3 Methodology

The video captioning problem is essentially modeled as a sequence to sequence process. Formally, given a sequence of T frames from video $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, we aim to build an end-to-end model to generate the caption $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T'}\}$ for the given video. Note $T \neq T'$, which forms an open task that is even very challenging for humans. In this chapter, we identify three key sub-tasks, namely Enhanced Object Proposal, Visual Knowledge, and Sentence Validation, details of which are introduced as follows.

6.3.1 Architecture Design

The overview of our proposed model is illustrated in Figure 6.2. The Latent Semantic Graph (LSG) consists of three parts: (1) Multiple Feature Extraction; (2) Conditional Graph Operation; (3) Latent Proposal Aggregation. A plug-in discriminative caption validator is illustrated in Figure 6.3. We introduce how our proposed models can address the sub-tasks next.

Multiple Feature Extraction. Given input video frames $\boldsymbol{X}$, the model first extracts visual context representations. In this work, 2D CNNs and 3D CNNs are employed to extract *appearance features* $\boldsymbol{V}^a = \{\boldsymbol{v}_t^a\}_{t=1}^T$ and *motion features*

$\boldsymbol{V}^m = \{\boldsymbol{v}_t^m\}_{t=1}^T$ respectively. Object proposals are extracted by R-CNNs to capture key entities with *region features* $\boldsymbol{R} = \{\boldsymbol{r}_t\}_{t=1}^T$ from each frame, where $\boldsymbol{r}_t = \{\boldsymbol{r}_t^i \in \mathbb{R}^{D_r}\}_{i=1}^N$ and N denotes the number of region features in each frame. Thus, the total number of object proposals is denoted as $L = T \times N$. Next, to make full use of motion information, we concatenate the appearance features and motion features, and apply LSTM models to learn better representations of motion features.

Enhanced Object Proposal. In video captioning, one of the essential tasks is to detect and recognize the entities. The weak object proposals in region feature are enhanced by their visual contexts of appearance and motion, respectively, which result in *enhanced appearance proposals* $\hat{\boldsymbol{V}}^a \in \mathbb{R}^{T \times D_g}$ and *enhanced motion proposals* $\hat{\boldsymbol{V}}^m \in \mathbb{R}^{T \times D_g}$ in a graph structure, where D_g is the feature dimension used in graph operation. $\hat{\boldsymbol{V}}^a$ and $\hat{\boldsymbol{V}}^m$ together form the enhanced object proposals.

Visual Knowledge. The Latent Proposal Aggregation (LPA) module introduces a dynamic graph that can summarize the enhanced appearance and motion features to latent semantic proposals as K dynamic *visual words*: $\boldsymbol{P}^o \in \mathbb{R}^{K \times D_g}$ and $\boldsymbol{P}^m \in \mathbb{R}^{K \times D_g}$. Note $K \ll T$.

Language Decoder. Visual knowledge extracted by the LPA is then used to generate corresponding captions. We adopt the language generation decoder that are commonly used in VAQ and video captioning fields [59, 61, 221, 227]. The language decoder consists of an *attention LSTM* network for weighting *dynamic visual words* and a *language LSTM* network for caption generation. At each time step, *attention LSTM* takes current word embedding and global visual vector $\bar{\boldsymbol{p}} = [\sum_{k=1}^K \boldsymbol{p}_k^o, \sum_{k=1}^K \boldsymbol{p}_k^m] \in \mathbb{R}^{2 \times D_g}$ as input and output current hidden state $\boldsymbol{h}_t^{attn}$.The $\boldsymbol{h}_t^{attn}$ is then treated as the query of the attention operation to weight sum the object and motion visual words to context feature $\boldsymbol{c}_t^{op}, \boldsymbol{c}_t^{mp} \in \mathbb{R}^{D_g}$. The *language LSTM* then takes the current context features and current *attention LSTM* hidden states and output current predicted word probability distribution $\boldsymbol{c} \in \mathbb{R}^{D_{vocab}}$, where D_{vocab} is the vocabulary size.

6.3.2 Latent Semantic Graph

There has been significant research investigating the dependencies between objects and complex content in generating video captions. However, learning spatio-temporal dependencies remains a challenging issue. Compared to conventional spatio-temporal convolution and recursive neural networks, graph models provide a new solution to model dependencies. In this work, we propose the LSG model that can efficiently encode object-level features from videos as highly summarized *visual words* with higher semantic level. To progressively generate the high-level concepts representing visual features, the essential parts of the LSG model is divided into two components: conditional graph operation and latent proposal aggregation.

Conditional Graph Operation. In video captioning, one of the key challenges is to model the complex object-level interactions and relationships. Another challenge is to learn informative object-level features that are in context of frame-based background information. To encode object-level information as highly summarized latent semantic objects and motion *visual words* conditioned on frame-based background information, we first aggregate object-level features into appearance and motion features respectively via graph operation.

Since we have the object-level *region features*, frame-level *motion features* and *appearance features* after the Multiple Feature Extraction step mentioned in Section 3.1, we build a graph neural network to model object-level interactions, where each region feature $\boldsymbol{r}^j$ out of all L region features is regarded as a node. To modeling the frame-based conditioning, instead of relying only on the local region features, we take the full picture into account, which takes both frame-level motion and appearance features and object-level *region features*. Specifically, we pass messages of the region features to frame-level features at each frame t:

$$\hat{\boldsymbol{v}}_t^a = \boldsymbol{v}_t^a + \sum_{j=1}^{L} \mathcal{F}_{kernel}\left(\boldsymbol{v}_t^a, \boldsymbol{r}^j\right) \boldsymbol{W}_a \boldsymbol{r}^j \ , \tag{6.1}$$

where $\hat{\boldsymbol{v}}_t^a$ represents the t^{th} representation of *enhanced appearance proposal*, and $\boldsymbol{W}_a \in \mathbb{R}^{D_g \times D_r}$ denotes learnable parameters. Specially, $\mathcal{F}_{kernel}$ is a kernel function that aims to encode relations between frame-level features $\boldsymbol{v}_t^a$ and detailed region

features $\boldsymbol{r}^j$. In this study, we define $\mathcal{F}_{kernel}$ as:

$$\mathcal{F}_{kernel}\left(\boldsymbol{v}_t^a, \boldsymbol{r}^j\right) = \psi(\boldsymbol{v}_t^a)\phi(\boldsymbol{r}^j)^T \ , \tag{6.2}$$

where ψ and ϕ are linear functions followed by Tanh activation function. This step aims to project features from different modalities to a common feature space and compute the similarity to represent the degree of connectivity between region features and frame-level features in the graph. Alternatively, the equation can be written as:

$$\hat{\boldsymbol{V}}^a = \boldsymbol{V}^a + \boldsymbol{A}(\boldsymbol{V}^a, \boldsymbol{R})\boldsymbol{R}\boldsymbol{W}_a \ , \tag{6.3}$$

where $\boldsymbol{A} \in \mathbb{R}^{T\times L} = \mathcal{F}_{softmax}(\psi(\boldsymbol{V}^a)\phi(\boldsymbol{R})^T)$ denotes the relation coefficient matrix between *appearance features* and *region features*. Meanwhile, region features are aggregated to motion features as $\hat{\boldsymbol{V}}^m$ in the same process. With $\hat{\boldsymbol{V}}^a$ and $\hat{\boldsymbol{V}}^m$ that contain object-level information on the condition of frame-level features, we then need to summarize the enhanced proposals to obtain informative semantic concept candidates or proposals with less redundancy.

Latent Proposal Aggregation. To further summarize the *enhanced object proposals*, we propose a latent proposal aggregation method to generate visual words dynamically based on the enhanced features $\hat{\boldsymbol{V}}^a$ and $\hat{\boldsymbol{V}}^m$ inspired by [228]. First, we augment the original enhanced proposal nodes $\{\hat{\boldsymbol{v}}_t^a\}_{t=1}^T$ and $\{\hat{\boldsymbol{v}}_t^m\}_{t=1}^T$ with a set of additional latent nodes, and then aggregate information from the enhanced proposals to the latent nodes in a graph structured manner. Specifically, we introduce a set of *object visual words* $\boldsymbol{P}^o = \{\boldsymbol{p}_k^o \in \mathbb{R}^{D_g}\}_{k=1}^K$, which means potential object candidates in the given video. Note that K indicates the number of visual words, so that we can summarize the enhanced proposals into informative dynamic visual words. The aggregation process is defined as:

$$\boldsymbol{p}_k^o = \sum_{j=1}^{T} \mathcal{F}_{kernel}\left(\theta_k^o, \hat{\boldsymbol{v}}_j^a\right) \boldsymbol{W}_{op}\hat{\boldsymbol{v}}_j^a \ , \tag{6.4}$$

where $\boldsymbol{p}_k^o$ denotes the k^{th} object visual word and $\theta_k^o \in \mathbb{R}^{D_g}$ denotes learnable parameters for the k^{th} object visual word. Following the same process, we can derive

the *motion visual words* $\boldsymbol{P}^m = \{\boldsymbol{p}_k^m \in \mathbb{R}^{D_g}\}_{k=1}^K$ that represent potential motion candidates in the given video. Therefore, with LSG, we extract the high-level representation and summarize information as dynamic visual words from a video that models both object-level interaction and frame-level condition. The latent semantic visual words are then feed into the language decoder to generate captions as mentioned in Section 3.1. Although the output sequence is generated based on the visual words, there's still potential to obtain video description with more meaningful semantic concepts for more informative caption generation.

6.3.3 Discriminative Language Validation

While other discriminative models for video captioning mainly focus on fluency and visual relevance of the generated descriptions, we aim to generate meaningful captions from the perspective of semantic concepts. In our approach, we design a discriminative model as a language validation process that encourages the generated captions to contain more informative semantic concepts via reconstructing the visual words or knowledge based on the input sentences under the condition of corresponding true visual words encoded by LSG. Specifically, based on the visual knowledge $\boldsymbol{P}^o$ and $\boldsymbol{P}^m$ encoded from input video features, we propagate information from the generated captions to reconstruct visual knowledge and discriminate the reconstructed visual words from ground-truth and generated captions in an adversarial training manner. The process of the discriminative modeling is summarized in Algorithm 1 and described as follows.

Given the output word sequence from the language decoder, the discriminative model aims to distinguish the generated captions and ground-truth with regard to the semantic concepts in the corresponding sentences. To prevent the discriminative model from easily distinguishing between real and fake samples without learning useful information (the ground truth caption is the one-hot integer datatype while the generated caption contains probability distributions) and to stabilize the training process, we employ the WGAN-GP architecture as it uses the earth-mover distance to capture the difference between real and fake samples, which is well suited to our problem.

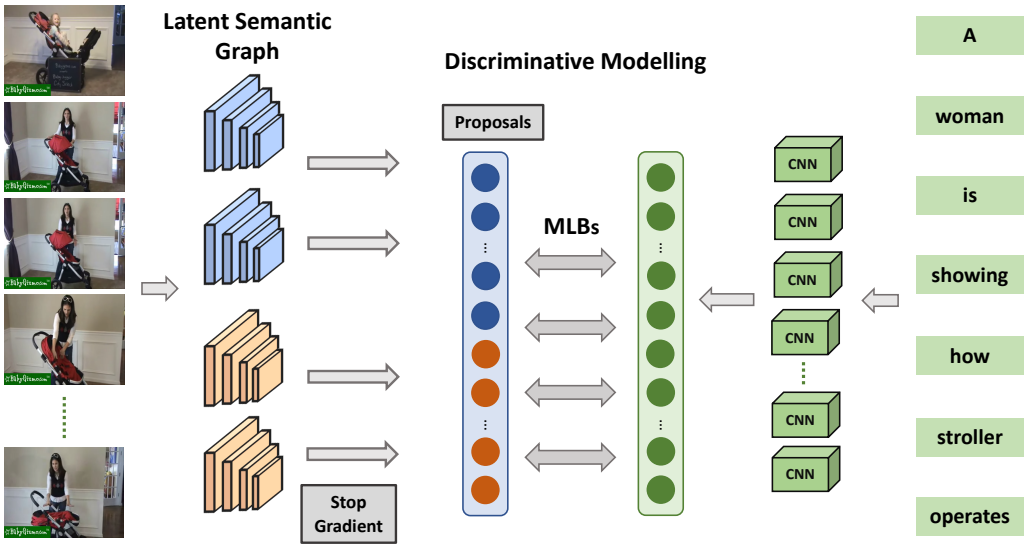


Figure 6.3: An overview of our discriminative modeling process. We score an input sentence in a semantic concept perspective of view. The model reconstructs the visual knowledge based on the input sentence and output comparison scores with visual knowledge encoded from the corresponding video.

The discriminative model first extracts sentence features $\boldsymbol{S} \in \mathbb{R}^{T' \times D^s}$ from the given caption $C \in \mathbb{R}^{T' \times D_{vcab}}$ by using several layers of 1D CNN with residual connections. Then, we adopt a graph-based structure to obtain the reconstructed object and motion visual words $\hat{\boldsymbol{P}}^o$ and $\hat{\boldsymbol{P}}^m$ from the caption:

$$\hat{\boldsymbol{P}}^o = \boldsymbol{A}(\boldsymbol{P}^o, \boldsymbol{S})\boldsymbol{S}\boldsymbol{W}_w \ . \tag{6.5}$$

Note that visual words $\boldsymbol{P}$ from the LSG model stops gradient descent before passing to the discriminative model so that it does not affect the caption generation. Following the same process, we can derive the aggregated motion visual words $\hat{\boldsymbol{P}}^m$. Then, we compare the visual words $\boldsymbol{P}$ with the aggregated visual words $\hat{\boldsymbol{P}}$ by Multimodal Low-rank Bi-linear pooling (MLB) [222], which is recognized to be efficient in tasks such as VQA. To be specific, for each pair of visual words $\hat{\boldsymbol{p}}_i$ and $\boldsymbol{p}_i$:

$$e_i = \sigma \left(\tanh \left(U^T \hat{\boldsymbol{p}}_i \right) \odot \tanh \left(V^T \boldsymbol{p}_i \right) \right) \ , \tag{6.6}$$

where σ is the sigmoid activation function, $\odot$ denotes the Hadamard product, U and V are learnable parameters. Note that e_i is a scalar which means the score of

the input aggregated proposal compared to the original one. e_i^o is score of i^{th} object visual word pair and e_i^m is score of i^{th} motion visual word pair. Then the overall comparison score can be expressed as:

$$e^o = \frac{1}{K'}\sum_{i=1}^{K'} e_i^o \ , e^m = \frac{1}{K'}\sum_{i=1}^{K'} e_i^m \ , \tag{6.7}$$

where K' is the number of visual words selected out of K for reconstruction and comparison. The intuition behind is that the captions contain less semantic information than video. Instead of adding e^o and e^m as the discriminative model's output, we weight them adaptively based on the sentence feature since sentences have different proportions of object and motion concepts. We calculate the output of the discriminative model as following:

$$\beta_s = \frac{e^{a_o^T S}}{e^{a_o^T S} + e^{a_m^T S}} \ , \tag{6.8}$$

$$D(C|\boldsymbol{P}) = \beta_s e^o + (1-\beta_s)e^m \ , \tag{6.9}$$

where $a_o, a_m \in \mathbb{R}^{D_s}$ are learned parameters, and $S \in \mathbb{R}^{D_s}$ is mean pooled sentence feature. $D(C|\boldsymbol{P})$ is the output of the discriminative model that learns to give real captions large values and minimize the values of generated captions. For the ground-truth caption $C_r = \{y_t\}_{t=1}^T$ and generated caption C_g, the loss function of the discriminative model is defined as:

$$\mathcal{L}_D = D(C_g|\boldsymbol{P}) - D(C_r|\boldsymbol{P}) + \lambda(\left\|\nabla_{\hat{C}} D(\hat{C})\right\|_2 - 1)^2 \ , \tag{6.10}$$

where $\hat{C}$ is sampled along straight lines between real caption C_r and generated caption C_g. $(\left\|\nabla_{\hat{C}} D(\hat{C})\right\|_2 - 1)^2$ is the gradient penalty term that forces the gradient from the generated sample to the real sample to be as small as possible to satisfy the Lipschitz constraint [225]. Thus, for the generator, the loss function is calculated as:

$$\begin{aligned} \hat{\mathcal{L}_G} &= -D(C_g|\boldsymbol{P}) \ , \\ \mathcal{L}_G &= \mathcal{L}_C + \beta\hat{\mathcal{L}_G} \ , \end{aligned} \tag{6.11}$$

where β is the hyperparameter that controls the weight of $\hat{\mathcal{L}_G}$. $\mathcal{L}_C$ is the caption generation loss.

Algorithm 2: Discriminative modeling algorithm

/* θ^{Disc} : Parameters of Discriminative model; */
/* n_{Disc} : Number of Discriminative model iterations per generator iteration; */
/* θ^{LSG} : Parameters of LSG model; */
/* SG : Stop Gradient; */

Function $\mathcal{F}_{Disc}(\boldsymbol{P}^o, \boldsymbol{P}^m, \boldsymbol{C})$:
  Require : object visual words $\boldsymbol{P}^o$; motion visual words $\boldsymbol{P}^m$; word sequence $\boldsymbol{C}$.
  $\boldsymbol{S} = CNNs(\boldsymbol{C})$
  $\hat{\boldsymbol{P}}^o = \boldsymbol{A}(\boldsymbol{P}^o, \boldsymbol{S})\boldsymbol{S}\boldsymbol{W}_w$
  $e_i = \sigma\left(\tanh\left(U^T\hat{\boldsymbol{p}}_i\right) \odot \tanh\left(V^T\boldsymbol{p}_i\right)\right)$
  $e^o = \frac{1}{K'}\sum_{i=1}^{K'} e_i^o$
  $e^m = \frac{1}{K'}\sum_{i=1}^{K'} e_i^m$
  $\beta_s = \frac{e^{a_o^T S}}{e^{a_o^T S}+e^{a_m^T S}}$
  return $\beta_s e^o + (1-\beta_s)e^m$

Initialize: θ^{LSG}, θ^{Disc}
for *i = 1 to epoch number* **do**
  Sample i^{th} minibatch of video $\boldsymbol{V}_i$ and corrsponding caption $\boldsymbol{C}_i$
  $\boldsymbol{P}^o$, $\boldsymbol{P}^m$, $\boldsymbol{C}^g = \mathcal{F}_{LSG}(\boldsymbol{V}_i)$
  $\boldsymbol{P}^o_{SG}$, $\boldsymbol{P}^m_{SG}$, $\boldsymbol{C}^g_{SG} = \mathcal{F}_{LSG}(\boldsymbol{V}_i)$
  for *t = 1 to* n_{disc} **do**
    $\mathcal{L}_D = \mathcal{F}_{Disc}(\boldsymbol{P}^o_{SG}, \boldsymbol{P}^m_{SG}, \boldsymbol{C}^g_{SG}) - \mathcal{F}_{Disc}(\boldsymbol{P}^o_{SG}, \boldsymbol{P}^m_{SG}, \boldsymbol{C}) + \lambda(\left\|\nabla_{\hat{C}}\mathcal{F}_{Disc}(\boldsymbol{P}^o_{SG}, \boldsymbol{P}^m_{SG}, \hat{\boldsymbol{C}}_i)\right\|_2 - 1)^2$
    Update θ^{Disc} with $\mathcal{L}_D$
  $\hat{\mathcal{L}_G} = -\mathcal{F}_{Disc}(\boldsymbol{P}^o_{SG}, \boldsymbol{P}^m_{SG}, \boldsymbol{C}^g)$
  $\mathcal{L}_G = \mathcal{L}_C(\boldsymbol{C}^g, C) + \beta\hat{\mathcal{L}_G}$
  Update θ^{LSG} with $\mathcal{L}_G$

Overall, the LSG model aims to summarize input video into high-level visual words to generate informative captions, and the discriminative modeling enhances the generated captions to be more semantically relevant.

Table 6.1: Comparison between the proposed D-LSG and the state-of-the-art methods on MSVD and MSR-VTT datasets. B@4, M, R and C denote BLUE-4, METEOR, ROUGE-L and CIDEr, respectively.

Method	MSVD				MSR-VTT			
	B@4	M	R	C	B@4	M	R	C
PickNet [229]	52.3	33.3	69.6	76.5	41.3	27.7	59.8	44.1
MARN [230]	48.6	35.1	71.9	92.2	40.4	28.1	60.7	47.1
OA-BTG [59]	56.9	36.2		90.0	41.4	28.2	-	46.9
RMN [61]	54.6	36.5	73.4	94.4	42.5	28.4	61.6	49.6
STG [226]	52.2	36.9	73.9	93.0	40.5	28.3	60.9	47.1
ORG-TRL [219]	54.3	36.4	73.9	95.2	43.6	28.8	62.1	50.9
C-R Reasoning [221]	57.0	36.8	-	96.8	-	-	-	-
D-LSG	**60.9**	**37.6**	**75.2**	**100.8**	**44.6**	**28.8**	**62.3**	**51.2**

6.4 Experiments

In this section, we present our experimental results on two public datasets: MSVD [223] and MSR-VTT [163]. We compare our D-LSG with other state-of-the-art methods and an in-depth ablation study is provided to better understand our method.

6.4.1 Experimental Setup

Datasets. 1) MSVD contains 1970 different YouTube short video clips with an average video length of 10.2s. For each video, we used around 40 captions as only English was considered in all experiments. Following [61], we divided the dataset into three parts with 100 clips for validation, 1200 clips for training, and the remaining 670 clips for testing. 2) MSR-VTT is another dataset for open domain video captioning which consists of 10,000 video clips with an average video length of 14.8s and each of them is annotated with 20 English expressions. They are divided into 20 categories, such as music and movie. For fair comparison, the standard splits are 6513 training videos, 497 validation videos and 2990 test videos.

There are other notable datasets available for video captioning. For instance, YouCookII is a specialized cooking domain dataset that encompasses 89 recipes with 15.4K video clips, each accompanied by a single ground-truth caption. TVC, on the other hand, is a dataset hailing from the TV domain. It contains a massive 262K

caption descriptions paired with 108K video segments. An interesting characteristic of TVC is that the captions not only delineate the video contents but can also correspond to the subtitles. Despite the potential advantages of these datasets, we opted for MSVD and MSR-VTT due to computational resource constraints, which restricted the scope of data we could efficiently process and analyze in our experiments.

Evaluation Metrics. For a fair comparison, the quality of the generated captions in this study is evaluated by four evaluation metrics: BLEU-4 [231], METEOR [232], CIDEr [233] and ROUGE-L [234]. BLEU-4 measures the fraction of overlapping n-grams (here n = 4) between predicted sentences and reference sentences. METEOR calculates the precision and recall between predicted sentence and references based on uni-gram, which extends exact word matching to various match levels. CIDEr evaluates the consensus between a predicted sentence and reference sentences of the corresponding image or video based on the number of overlapping units such as n-gram. ROUGE-L computes recall and precision scores of the longest common subsequences (LCS) between the generated and each reference sentence. For all metrics, a higher value represents better performance of the generated captions.

Data Preprocessing and Feature Extraction. We follow the process of [61] for corpus preprocessing and feature extraction. For corpus preprocessing, captions are first converted to lower case and punctuations are removed. Then, captions with more than 26 words are truncated and captions with less than 26 words are zero-padded. Besides, words that appear less than twice and five times are deleted in MSVD and MSR-VTT, respectively. For feature extraction, 2D and 3D CNN feature extractors are InceptionResNetV2 (IRV2) [235] and I3D [236]. Features from 26 frames are uniformly sampled in each video. Faster-RCNN [7] is adopted to extract the 36 region features for each frame out of the 26 sampled frames.

Implementation Details. The Adam optimizer is applied with a learning rate 8×10^{-4} for LSG model. For the discriminative model, we applied the Adam optimizer with ascent learning rates from 2×10^{-4} to 8×10^{-4}. The size of hidden states for all LSTM models is 1024 and 1536 in MSVD and MSR-VTT datasets, respectively. The feature size for all graph operations is set to 1024 for both datasets. Layer

Table 6.2: Ablation Study of the proposed D-LSG on MSVD and MSR-VTT datasets. B@4, M, R and C denote BLUE-4, METEOR, ROUGE-L and CIDEr, respectively. CGO only denotes the model only applies Conditional Graph Operation. LPA only indicates the model only applies Latent semantic Aggregation.

Method	Component			MSVD				MSR-VTT			
	CGO	LPA	D	B@4	M	R	C	B@4	M	R	C
baseline	×	×	×	53.9	35.4	72.8	92.7	42.1	27.5	61.1	48.2
CGO only	✓	×	×	56.5	36.8	73.6	96.1	43.2	28.2	61.9	50.3
LPA only	×	✓	×	55.8	36.1	73.2	95.4	42.8	28.1	61.7	50.1
LSG	✓	✓	×	57.9	37.2	74.1	99.4	44.5	28.5	62.0	50.7
D-LSG	✓	✓	✓	**60.9**	**37.6**	**75.2**	**100.8**	**44.6**	**28.8**	**62.3**	**51.2**

normalization is applied on top of the LSTM layer and graph nodes to speed up convergence. The word embedding size is set to 300 without the use of any pre-trained embedding such as glove. Feature dimension in CNN for discriminative model word feature extraction is set to 512. The training batch size is set to 128 for both datasets. Beam search is applied during inference with size 5.

6.4.2 Quantitative Evaluation

We compare our proposed D-LSG model with the state-of-the-art models on the MSVD and MSR-VTT datasets to evaluate our model's performance, and the results are listed in Table 6.1. The results illustrate that our model achieves the best performance on the MSVD and MSR-VTT datasets for all evaluation metrics, which indicates the effectiveness of our proposed model for the video captioning task. The detailed analysis of results on the MSVD and MSR-VTT datasets is shown below.

Comparison with encoder-decoder models. We first compare our model with traditional encoder-decoder based models, including PickNet [145] and MARN [230]. We can observe from Table 6.1 that the performance gains significant improvement, which means object information plays an important role in video captioning task.

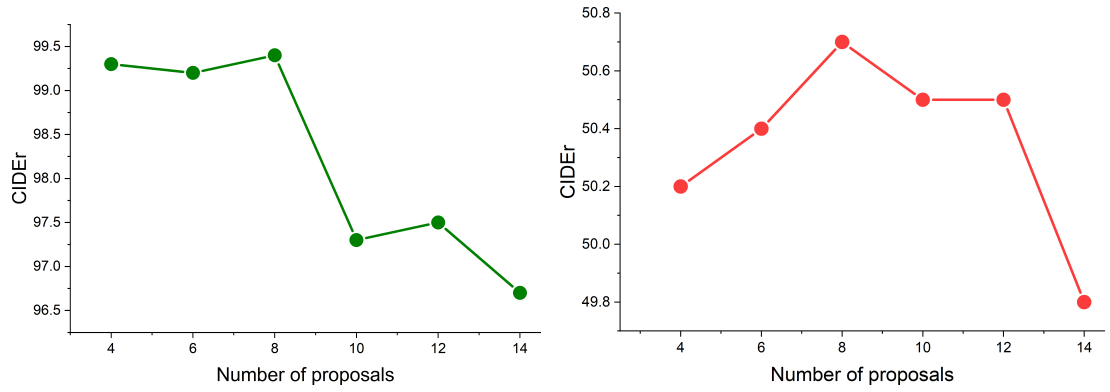
Comparision with object-based models. We then compare with several recent studies that consider detailed object information, including OA-BTG [59] and RMN [61]. We can observe that our D-LSG gains better performance in all metrics on both datasets, proving the effectiveness of utilizing graph-based models to learn

object-level features. OA-BTG builds a bi-directional temporal graph based on object features. However, the interaction between different objects is less encoded in the graph structure. RMN utilizes an attention mechanism for encoding the object-level features. However, the object interactions are only considered in the motion modeling module. We observe that out D-LSG model provides more obvious improvement on CIDEr than BLUE-4, which indicates that modeling interactions between different objects helps generate rich semantic captions.

Comparision with GNN-based models. Finally, we compare our model with the most recent approaches that adopt GNN based methods, including ORG-TRL [219], S-T Graph [226], and C-R Reasoning [221]. Our model outperforms the mentioned GNN based models and achieves excellent performance in BLUE-4 and CIDEr metrics. The BLUE-4 metric focuses on the fluency and logic of the generated captions and the CIDEr metric mainly focuses on content-relevant words in videos. The performance proves that D-LSG successfully captures high-level semantic concepts. In contrast, while ORG-TRL employs GCN to model object interactions, ORG-TRK does not take frame-level information into account when conducting object-level graph convolution. Our proposed LSG has better performance, indicating that aggregating object-level information conditioned on frame-level features helps the GNN learn better object representations. C-R Reasoning utilizes extra datasets to build semantic knowledge graphs. On the contrary, we propose to use discriminative modeling to enhance the semantic extraction via reconstructing visual words. This demonstrates that D-LSG can succeed in extracting semantic information without using auxiliary databases.

6.4.3 Ablation Study

We then verify the effectiveness of the proposed D-LSG method through ablation studies on the MSVD and MSR-VTT dataset as shown in Table 6.2: (1) baseline: the model inputs the concatenation of appearance feature and motion feature to the language decoder directly. (2) CGO only: the model only employs conditional graph operation. It aggregates the region features to frame-level appearance feature and motion feature, and feeds the enhanced object proposals directly to the language de-



(a) MSVD (b) MSR-VTT

Figure 6.4: CIDEr of different visual words number in LSG on both MSVD and MSR-VTT datasets.

coder without Latent Proposal Aggregation. (3) LPA only: the model summarizes the frame-level feature to visual words via Latent Proposal Aggregation without modeling object-level information. (4) LSG: the model includes the complete latent semantic graph. (5) D-LSD: the model combines LSG and the dsicriminative modeling part.

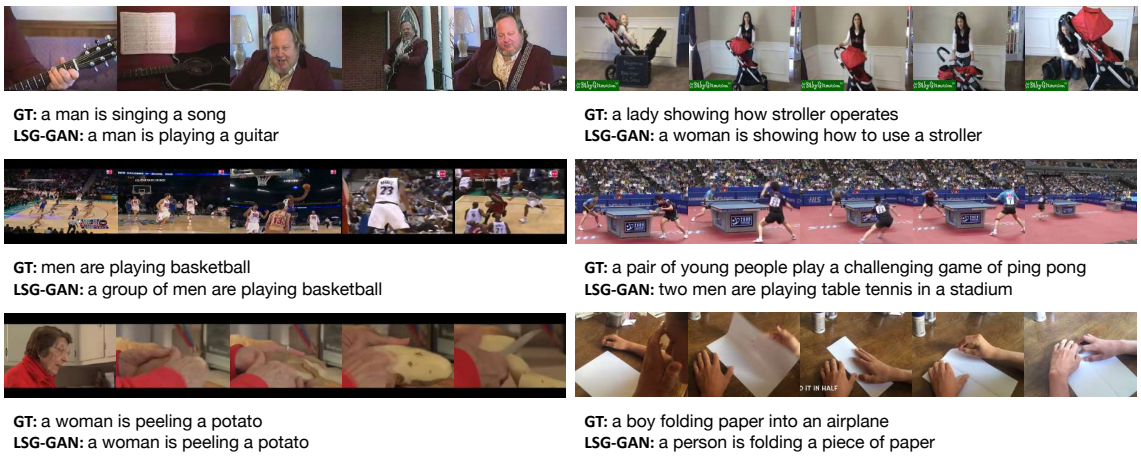


Figure 6.5: Qualitative results of four videos from the MSVD and MSR-VTT datasets. The first line in each example is one of the ground truth captions and the second line is generated by our D-LSG method.

Effect of Graph. Comparing the results of CGO only and LSG, we observe a noticeable performance decrease on both datasets, which indicates the importance of summarizing frame-level features to latent concepts or visual words. Comparing the results of LAP only and LSG, the performance also decreases. This is because LAP

does not employ the conditional graph operation so that the visual words obtained in this model lack detailed object-level information. Since the performance drop of the LAP only model compared with the CGO only model is more pronounced, we can conclude that summarizing only frame-level information is not representative enough for semantic concepts, which also implies CGO's effectiveness modeling detailed object information.

Effect of latent proposal number. We also evaluated how the number of visual words affects the quality of the generated captions on the MSVD and MSR-VTT datasets. Figure 6.4 illustrates the performance on CIDEr using different numbers of visual words for both MSVD and MSR-VTT dataset. As for the MSVD dataset, a small number of latent proposals provides better performance on CIDEr. However, when the number of proposals increases, the performance drops significantly. The intuition behind this is that the videos and captions in the MSVD dataset are short, so the data do not have enough semantic information to construct a large number of visual words, which results in performance drops. On the contrary, MSR-VTT with longer videos and captions suffers more performance drop when the number of proposals is small, which means that a small number of visual knowledge is not enough to represent the semantic concept of a given video. The above examples imply that The LSG model is able to summarize video content into semantic concepts with a proper number of visual words.

Effect of discriminative modeling based on Graph. For the MSVD dataset, comparing LSG and D-LSG, we observe that METEOR and ROUGE-L have a slight improvement, while BLEU-4 and CIDEr show large improvements, especially for CIDEr. Though the advantage is less apparent on the MSR-VTT dataset, the increase of CIDEr is also noticeable when comparing the improvement on other evaluation metrics. Since the mechanism of CIDEr is to punish words that are less informative of the video content, it may indicate that the dsicriminative structure can enrich the semantic concepts of the generated sentences, which means the model is capable of helping LSG capture and summarize key semantic concepts more effectively from input video features.

6.4.4 Qualitative Evaluation

Figure 6.5 provides a visual demonstration of the generated captions based on the MSVD and MSR-VTT datasets. When juxtaposed with the ground truth captions, several insightful observations can be drawn. Firstly, it's evident that the generated captions effectively encapsulate crucial objects present in the scenes, such as "man", "guitar", and "stroller". This encapsulation also extends to the depiction of various actions or motions, like "playing" and "showing". Notably, even nuanced actions such as "peeling" and "folding", which might be infrequently represented in the dataset, are accurately recognized and reflected in the captions.

Additionally, a broader understanding of the video content can be inferred from the captions. This is manifested in the way visual words extracted by the model encapsulate the essence of the video content. A notable aspect is the model's ability to recognize and articulate the broader context or setting of the video. This capability is exemplified by the model's detection and mention of the "stadium" in the example at the bottom-right, which indicates the model's adeptness at capturing frame-based background information instead of solely emphasizing detailed object-centric information.

6.4.5 Multilingual Adaptation of D-LSG

The D-LSG framework's adaptation for multilingual captioning involves modifying its core modules. For the *Enhanced Object Proposal*, integration of multilingual entity recognizers can provide accurate object labels aligned with the target language. The *Visual Knowledge* module should be fine-tuned with a target-language-annotated dataset to capture language-specific visual semantics. Finally, for *Sentence Validation*, using a target-specific language model can refine the fluency and structure of generated captions. Fine-tuning and employing language-specific tools are crucial for effective multilingual adaptation.

6.5 Conclusion

We have presented the first work to introduce graph neural networks and discriminative modeling to process spatio-temporal information for the video captioning task jointly. As for the Latent Semantic Graph, from the experiment results, we conclude that the Conditional Graph Operation effectively models detailed object-level interactions and relationships. Besides, considering frame-level conditions is conducive to object-level interactive representation learning. The Latent Proposal Aggregation component also succeeded in summarizing high-level visual knowledge from input video features. Also, the discriminative modeling enriched the generated captions' semantic information via visual knowledge reconstruction and discriminative training. On two public datasets, our D-LSG model has outperformed the current state-of-the-art approaches, which verifies the effectiveness of our method.

CHAPTER 7

Conclusion

7.1 Contributions

In this thesis, we attempt to approach interaction-level action understanding. We first identify three main challenges to realising interactive-action understanding: 1) understand actions given human consensus; 2) understand actions based on specific human rules; 3) Directly understand actions in videos via human natural language. In Chapter 3, we explore the task video summary and proposed models using self-attention mechanism and meta-learning to solve the first challenge. In Chapters 4 and 5, we explore the second challenge with the task action quality assessment by proposing two novel models based on the transformer and counterfactual generation, respectively. Lastly, in Chapter 6, we propose models using the graph neural network trained in an adversarial fashion for video captioning. Detailed contributions are listed as follows.

Video Summary: In Chapter 3, we introduced the Dual Mixture Attention model (DMASum) to emulate human-like attention in the video summary task. By integrating visual semantics and motion data, and addressing the Softmax Bottleneck via our *Query Twice* module, we achieved a more refined attention mechanism.

Importantly, our DMASum surpassed other models in performance, even matching human-level accuracy on selected benchmarks.

Action Quality Assessment: In Chapters 4 and 5, our focus was on action quality assessment. Chapter 4 introduced a temporal parsing transformer, using unsupervised learning to capture temporal sub-actions without relying on expensive labeled data. In Chapter 5, we further refined sub-action quality assessment through a generative cycle counterfactual framework. This approach emphasized understanding sub-action variations, leading to superior performance against other models across three datasets.

Video Captioning: Chapter 6 showcased the D-LSG model, a blend of graph neural networks and GANs for video captioning. Through sequential processes like multiple feature extraction and latent semantic aggregation, the framework distilled object and motion information efficiently. A critical achievement was the model's ability to ensure the accuracy and cohesiveness of the generated captions. Notably, D-LSG set a new benchmark in performance against existing methods on two datasets.

Key Highlights: 1. The DMASum model's pioneering approach in video summarization, equating human accuracy in some benchmarks. 2. The introduction of unsupervised learning methods for discerning temporal sub-actions, removing the dependency on exhaustive labeled data. 3. D-LSG's innovative use of combined technologies with GNNs an GANs for video captioning, setting a new performance standard.

7.2 Future Work

For future work towards realizing interactive-level action understanding, one branch is to keep developing the tasks have been explored in this thesis, namely, video summary, action quality assessment, and video captioning. Another branch is to identify new tasks that are meaningful for more intelligence action understanding. The works we found interesting to further study are listed as follows.

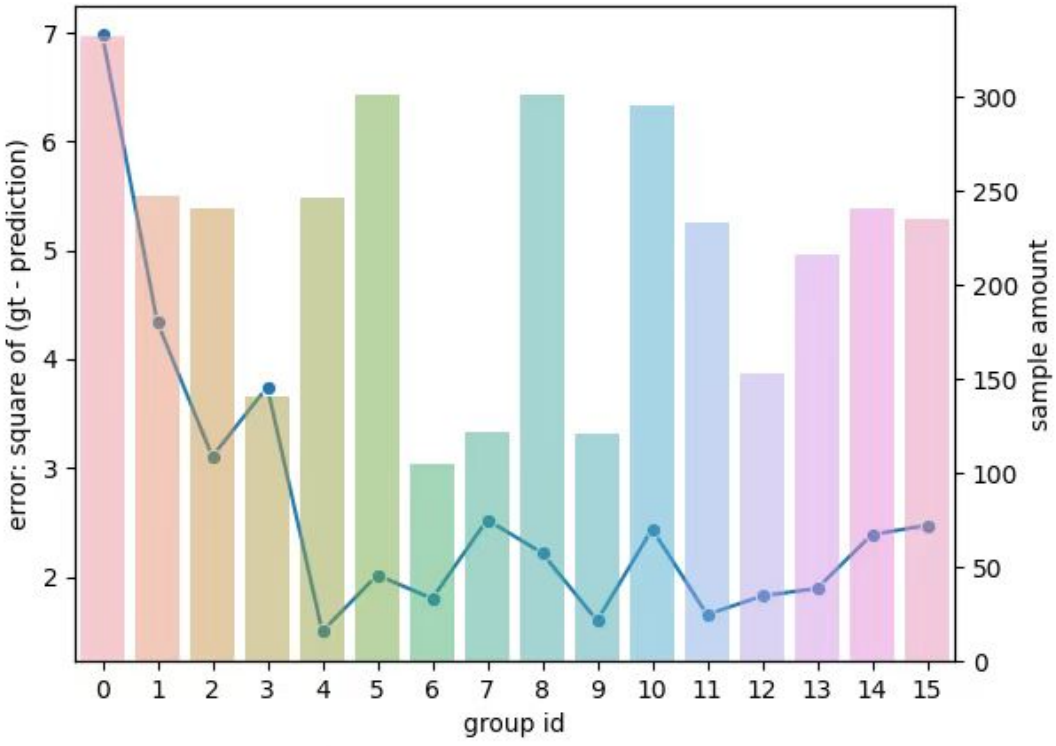


Figure 7.1: The mean errors of adopting different groups as examples during the inference stage using the contrastive regression framework. The blue line indicates the mean error of different groups while the colored bars indicates the number of samples inside each group. We divide the relative score of input videos and exemplar videos into 16 groups (from 0 to 15) according to training data. The smallest group ID (group 0) indicates the exemplar video is much better than the input video, while the largest group ID (group 15) indicates the quality of input video is much better than the exemplar videos. Group IDs in middle (group 7, 8) means the input and exemplar videos have similar quality score.

7.2.1 High-order Contrastive Regression for Action Quality Assessment

We adopt the contrastive regression framework in Chapter 4 to derive the quality score of a given video. In the contrastive regression framework, we sample an exemplar video for the current input video, and we take both videos as input and derive the relative score. The output can then be expressed as the summation of the relative score and the score of the exemplar video. However, the way we sample exemplars has a significant impact on the final performance. As shown in Figure

7.1, the performance derived from exemplars similar to the input video is much better than those derived from exemplars with larger relative scores. Besides, among exemplars with scores far from input video scores, the exemplars better than the input video (see groups 14, 15) perform much better than those worse than the input video. The above two observations drive us to think about the strategy of sampling better exemplars for more accurate contrastive regression. One feasible way is to sample multiple examples for training the contrastive regression framework. In this way, we can have worse, closer and better examples compared with the current input video to eliminate the problem of example selection during training and testing. Besides, adopting multiple exemplars is capable of capturing the high-order contrastive information based on the input video, so that we can derive more accurate relative scores. We can derive the first-order contrastive information between input video and examples. With the multiple exemplars, we can then derive the high-order contrastive information based on the multiple first-order contrastive information, which will enhance the first-order relative score. Also, we can have more robust final outputs by modelling the relations among the multiple relative scores. A possible way to model the high-order contrastive information is to adopt the transformer architecture that takes the multiple first-order contrastive information as input and performs self-attention to capture the high-order relationship. However, processing multiple exemplars during training could consume high computational costs, leading to slow convergence and training speed. The drawback will be carefully considered in our future work.

7.2.2 Visual-language Joint Processing for Video Captioning

In Chapter 6, we have verified the importance of aligning language features with visual content. However, we only align the two modalities implicitly, where the language semantic is forced to align with the video content, but the two modalities are trained separately. It means that the information from different modalities is not propagated fully. Therefore, it is reasonable to mix the information during training to fully align the semantic information from the two modalities, which we call visual-

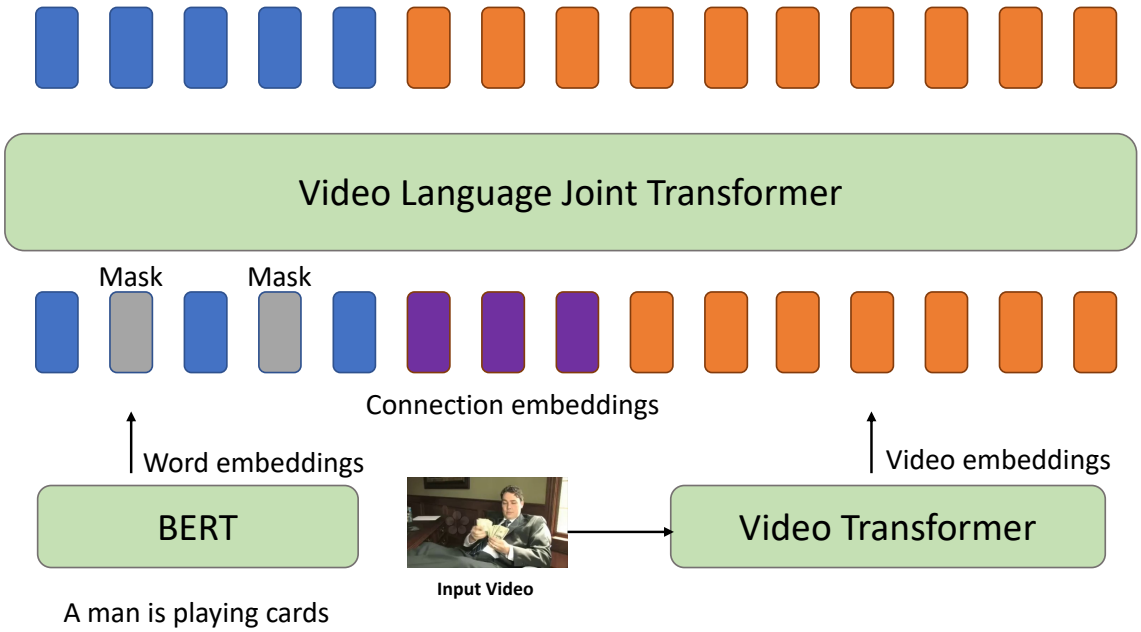


Figure 7.2: Overview of the initial framework for video captioning that is considered as our next step. The video language joint transformer model takes input from both sentences and videos. The sentence is first converted to a set of word embeddings via language models such as BERT. The input video is converted to a set of video embedding via video transformers such as video vit and video swin transformer. Based on the embeddings from words and videos, we then consider performing self-attention via the transformer encoder. Note that the word embeddings and the video embeddings can only communicate via the connection embeddings for fast convergence. The predicted words are then derived through masked language modeling.

language joint processing for video captioning. Since the words in sentences are tokenized into word embeddings with state-of-the-art language pretraining models such as BERT [28], it is necessary to convert video content into embeddings with semantic meaning so that we can align those embeddings from different modalities. Thanks to the development of the Transformer model in the computer vision community, many transformer-based methods can convert video frames into local embeddings, such as ViViT [237] and Video Swin Transformer [238]. To encode the embeddings from two modalities jointly, it is reasonable to adopt a transformer encoder to aggregate information. Considering that the two modalities are heterogeneous and joint process them in one transformer layer consumes much computational cost, in our future work, it is desired to add the connection embeddings into the transformer encoder. Hence, the word embeddings and the video embeddings can only communicate via the connection embeddings for fast convergence and easy training. The detailed process of possible future solution for video captioning is shown in Figure 7.2.

7.2.3 Multi-modal Action Understanding

Besides the tasks expolored in thesis, there's some other video-based tasks good for interactive action understanding. We study and verify the effectiveness of aligning features from different modalities in Chapter 6, which inpsire us to consider multi-modal action understanding the next future direction. Humans understand the world with multiple sensory streams - seeing objects, hearing sounds, reading texts and tasting flavours. Hence, real intelligence should also take benfitis from multi-modal sensories. A possible next step is to adopt video together with text and sound to understand actions more rubustly. Hence, real intelligence should also take benefit from multi-modal sensories. A possible next step is to adopt video together with text and sound to understand actions more robustly.

Bibliography

[1] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Group-aware contrastive regression for action quality assessment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7919–7928, 2021. (document), 2.5.2, 4.1, 4.2, 4.3.3, 4.3.4, 4.4.1, 4.1, 4.4.1, 4.2, 4.3, 4.4.2, 5.1, 5.4.1, 5.1, 5.2, 5.4.2, 5.3

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 1, 2.1, 2.2.1, 3.3.3

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015. 1, 2.2.1, 3.4.1

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 1, 2.1, 2.2.1

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. 1, 2.1, 2.2.1

[6] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1328–1338, 2019. 1, 2.2.1, 2.5.1

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015. 1, 2.2.1, 6.4.1

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020. 1, 2.2.1, 4.3.2, 4.3.2, 4.3.2, 4.4.3

[9] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021. 1, 2.2.1

[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015. 1

[11] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7151–7160, 2018. 1

[12] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018. 1

[13] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 1451–1460, Ieee, 2018. 1

[14] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural information processing systems*, vol. 33, pp. 17721–17732, 2020. 1, 2.2.1

[15] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation," in *European conference on computer vision*, pp. 312–329, Springer, 2016. 1, 2.2.1

[16] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018. 1, 2.2.1

[17] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9157–9166, 2019. 1, 2.2.1

[18] 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*, pp. 1725–1732, 2014. 1, 2.2.1, 2.3.2

[19] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015. 1, 2.2.2, 2.3.2

[20] 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*, pp. 4694–4702, 2015. 1, 2.2.2, 2.3.2

[21] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018. 1, 2.3.2

[22] 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*, vol. 35, no. 1, pp. 221–231, 2012. 1, 2.3.2, 5.1

[23] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European conference on computer vision*, pp. 140–153, Springer, 2010. 1, 2.3.2

[24] 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*, pp. 4489–4497, 2015. 1, 2.2.1, 2.3.2

[25] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013. 1.1, 4.1

[26] 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*, pp. 6299–6308, 2017. 1.1, 2.3.2, 2.3.2, 2.5.2, 4.1, 4.3.1, 4.4.1, 5.4.1

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 2.1, 2.2.1

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2.1, 7.2.2

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 2.1, 2.2.2

[30] 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*, pp. 5998–6008, 2017. 2.1, 3.2, 3.3

[31] D. Yu and L. Deng, *Automatic speech recognition*, vol. 1. Springer, 2016. 2.1

[32] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015. 2.1

[33] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015. 2.1

[34] B. Zhai, I. Perez-Pozuelo, E. A. Clifton, J. Palotti, and Y. Guan, "Making sense of sleep: Multimodal sleep stage classification in a large, diverse population using movement and cardiac sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–33, 2020. 2.1

[35] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017. 2.1

[36] Y. Bai, Y. Guan, J. Q. Shi, and W.-F. Ng, "Towards automated fatigue assessment using wearable sensing and mixed-effects models," in *2021 International Symposium on Wearable Computers*, pp. 129–131, 2021. 2.1

[37] Y. Bai, Y. Guan, and W.-F. Ng, "Fatigue assessment using ecg and actigraphy sensors," in *Proceedings of the 2020 International Symposium on Wearable Computers*, pp. 12–16, 2020. 2.1

[38] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12786–12796, 2022. 2.1

[39] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022. 2.1

[40] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 11320–11327, 2020. 2.1, 2.3.2

[41] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, and G. Hua, "Two-stream consensus network for weakly-supervised temporal action localization," in *European conference on computer vision*, pp. 37–54, Springer, 2020. 2.1, 2.3.2

[42] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1298–1307, 2019. 2.1, 2.3.2

[43] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2678–2687, 2016. 2.1

[44] K. Duarte, Y. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," *Advances in neural information processing systems*, vol. 31, 2018. 2.1

[45] L. Cao, Z. Liu, and T. S. Huang, "Cross-dataset action detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1998–2005, IEEE, 2010. 2.1

[46] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2914–2923, 2017. 2.1

[47] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7508–7517, 2020. 2.2.1

[48] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7760–7768, 2020. 2.2.1

[49] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018. 2.2.1

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. 2.2.1

[51] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *arXiv preprint arXiv:1708.05038*, 2017. 2.2.1

[52] S. Dupond, "A thorough review on the current advance of neural network structures," *Annual Reviews in Control*, vol. 14, pp. 200–230, 2019. 2.2.2

[53] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018. 2.2.2

[54] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018. 2.2.2

[55] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014. 2.2.2

[56] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1549–1557, IEEE, 2018. 2.2.2

[57] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based lstm networks for 3d action recognition and detection," *IEEE Transactions on image processing*, vol. 27, no. 7, pp. 3459–3471, 2018. 2.2.2

[58] J. Wang, Y. Bai, Y. Long, B. Hu, Z. Chai, Y. Guan, and X. Wei, "Query twice: Dual mixture attention meta learning for video summarization," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 4023–4031, 2020. 2.2.2

[59] J. Zhang and Y. Peng, "Object-aware aggregation with bidirectional temporal graph for video captioning," in *CPVR*, pp. 8327–8336, 2019. 2.2.2, 2.3.2, 2.5.3, 6.3.1, 6.1, 6.4.2

[60] Y. Hu, Z. Chen, Z.-J. Zha, and F. Wu, "Hierarchical global-local temporal modeling for video captioning," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 774–783, 2019. 2.2.2, 2.3.2, 2.5.3

[61] G. Tan, D. Liu, W. Meng, and Z.-J. Zha, "Learning to discretely compose reasoning module networks for video captioning," in *IJCAI-PRICAI*, 2020. 2.2.2, 2.3.2, 2.5.3, 6.3.1, 6.1, 6.4.1, 6.4.2

[62] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. 2.2.3, 5.3.2

[63] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 2.2.3

[64] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 185–200, 2018. 2.2.3

[65] C. You, G. Li, Y. Zhang, X. Zhang, H. Shan, M. Li, S. Ju, Z. Zhao, Z. Zhang, W. Cong, *et al.*, "Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle)," *IEEE transactions on medical imaging*, vol. 39, no. 1, pp. 188–203, 2019. 2.2.3

[66] S. Ruan, Y. Zhang, K. Zhang, Y. Fan, F. Tang, Q. Liu, and E. Chen, "Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13960–13969, 2021. 2.2.3

[67] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5802–5810, 2019. 2.2.3

[68] W. Liao, K. Hu, M. Y. Yang, and B. Rosenhahn, "Text to image generation with semantic-spatial aware gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18187–18196, 2022. 2.2.3

[69] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in *ICCV*, pp. 2970–2979, 2017. 2.2.3, 6.2

[70] A. Cherian and A. Sullivan, "Sem-gan: semantically-consistent image-to-image translation," in *2019 ieee winter conference on applications of computer vision (wacv)*, pp. 1797–1806, IEEE, 2019. 2.2.3

[71] H. Emami, M. M. Aliabadi, M. Dong, and R. B. Chinnam, "Spa-gan: Spatial attention gan for image-to-image translation," *IEEE Transactions on Multimedia*, vol. 23, pp. 391–401, 2020. 2.2.3

[72] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, pp. 2849–2857, 2017. 2.2.3

[73] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," *arXiv preprint arXiv:1803.07422*, 2018. 2.2.3

[74] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, and J. Liao, "Pd-gan: Probabilistic diverse gan for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9371–9381, 2021. 2.2.3

[75] Y. Jang, G. Kim, and Y. Song, "Video prediction with appearance and motion conditions," in *International Conference on Machine Learning*, pp. 2225–2234, PMLR, 2018. 2.2.3

[76] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*, 2018. 2.2.3

[77] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion gan for future-flow embedded video prediction," in *proceedings of the IEEE international conference on computer vision*, pp. 1744–1752, 2017. 2.2.3

[78] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8794–8802, 2019. 2.2.3

[79] Q. N. Tran and S.-H. Yang, "Efficient video frame interpolation using generative adversarial networks," *Applied Sciences*, vol. 10, no. 18, p. 6245, 2020. 2.2.3

[80] J. van Amersfoort, W. Shi, A. Acosta, F. Massa, J. Totz, Z. Wang, and J. Caballero, "Frame interpolation with multi-scale deep loss functions and generative adversarial networks," *arXiv preprint arXiv:1711.06045*, 2017. 2.2.3

[81] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5232–5239, 2019. 2.2.3

[82] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2019. 2.2.3

[83] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Free-form video inpainting with 3d gated convolution and temporal patchgan," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9066–9075, 2019. 2.2.3

[84] R. Shetty, M. Rohrbach, L. Anne Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," in *ICCV*, pp. 4135–4144, 2017. 2.2.3, 6.2

[85] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016. 2.2.3, 6.2

[86] Z. Zhao, S. Xiao, Z. Song, C. Lu, J. Xiao, and Y. Zhuang, "Open-ended video question answering via multi-modal conditional adversarial networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 3859–3870, 2020. 2.2.3

[87] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2, pp. 107–123, 2005. 2.3.1

[88] H. Wang, A. Klser, C. Schmid, and L. Cheng-Lin, "Action recognition by dense trajectories," in *Proceedings of the IEEE international conference on computer vision*, pp. 3169–3176, 2011. 2.3.1

[89] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on computer vision*, pp. 143–156, Springer, 2010. 2.3.1

[90] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013. 2.3.1

[91] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014. 2.3.2, 5.1

[92] 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*, pp. 1933–1941, 2016. 2.3.2

[93] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020. 2.4.1

[94] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang, "Embracing single stride 3d object detector with sparse transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8458–8468, 2022. 2.4.1

[95] X. Chen, S. Shi, B. Zhu, K. C. Cheung, H. Xu, and H. Li, "Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection," in *European Conference on Computer Vision*, pp. 680–697, Springer, 2022. 2.4.1

[96] L. Fan, F. Wang, N. Wang, and Z.-X. ZHANG, "Fully sparse 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 351–363, 2022. 2.4.1

[97] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3d object detection in monocular video," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pp. 135–152, Springer, 2020. 2.4.1

[98] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960. 2.4.1

[99] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning birds-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*, pp. 1–18, Springer, 2022. 2.4.1

[100] T. Wang, J. Pang, and D. Lin, "Monocular 3d object detection with depth from motion," in *European Conference on Computer Vision*, pp. 386–403, Springer, 2022. 2.4.1

[101] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016. 2.4.2

[102] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 501–518, Springer, 2016. 2.4.2

[103] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015. 2.4.2

[104] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pp. 298–372, Springer, 2000. 2.4.2

[105] P. Li, T. Qin, *et al.*, "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 646–661, 2018. 2.4.2

[106] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018. 2.4.2

[107] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020. 2.4.2

[108] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint arXiv:1806.04807*, 2018. 2.4.2

[109] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 284–293, 2019. 2.5.1

[110] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019. 2.5.1, 5.1

[111] Y. Zhu, Y. Long, Y. Guan, S. Newsam, and L. Shao, "Towards universal representation for unseen action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9436–9445, 2018. 2.5.1

[112] J. Wang, B. Hu, Y. Long, and Y. Guan, "Order matters: Shuffling sequence generation for video prediction," in *Proc. BMVA British Mach. Vis. Conf.*, pp. 275.1–275.14, 2019. 2.5.1

[113] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal cycle-consistency learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1801–1810, 2019. 2.5.1

[114] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2714–2721, 2013. 2.5.1, 3.4.1

[115] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems*, pp. 2069–2077, 2014. 2.5.1, 3.1, 3.4.1

[116] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3090–3098, 2015. 2.5.1, 3.4.1

[117] D. Liu, G. Hua, and T. Chen, "A hierarchical visual model for video object summarization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2178–2190, 2010. 2.5.1

[118] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, IEEE, 2012. 2.5.1

[119] X. Wang, Y.-G. Jiang, Z. Chai, Z. Gu, X. Du, and D. Wang, "Real-time summarization of user-generated videos based on semantic recognition," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM 14, (New York, NY, USA), p. 849852, Association for Computing Machinery, 2014. 2.5.1

[120] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*, pp. 766–782, Springer, 2016. 2.5.1, 3.1, 3.3, 3.4.1, 3.1, 3.2, 3.4.3, 3.4

[121] A. Kulesza, B. Taskar, *et al.*, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012. 2.5.1, 3.4.1

[122] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 202–211, 2017. 2.5.1, 3.1, 3.4.1, 3.1, 3.2, 3.4

[123] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2.5.1, 3.1, 3.4.1, 3.1, 3.2, 3.4

[124] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 383–399, 2018. 2.5.1, 3.3

[125] H. Wei, B. Ni, Y. Yan, H. Yu, X. Yang, and C. Yao, "Video summarization via semantic attended networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2.5.1, 3.1, 3.4.2, 3.1

[126] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkila, "Rethinking the evaluation of video summaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7596–7604, 2019. 2.5.1, 3.1, 3.4.1, 3.4.2, 3.4.2

[127] Q. Zhang and B. Li, "Relative hidden markov models for video-based evaluation of motion skills in surgical training," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 6, pp. 1206–1218, 2014. 2.5.2, 4.1, 5.1

[128] H. Doughty, D. Damen, and W. Mayol-Cuevas, "Who's better? who's best? pairwise deep ranking for skill determination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6057–6066, 2018. 2.5.2, 4.1, 5.1

[129] H. Doughty, W. Mayol-Cuevas, and D. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7862–7871, 2019. 2.5.2, 5.1

[130] P. Parmar and B. Morris, "Action quality assessment across multiple actions," in *2019 IEEE winter conference on applications of computer vision (WACV)*, pp. 1468–1476, IEEE, 2019. 2.5.2, 4.1, 4.1, 4.4.1, 5.1, 5.4.1

[131] P. Parmar and B. T. Morris, "What and how well you performed? a multitask learning approach to action quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 304–313, 2019. 2.5.2, 4.1, 4.1, 4.4.1, 4.1, 4.4.2, 5.1, 5.1, 5.4.1, 5.1

[132] A. S. Gordon, "Automated video assessment of human performance," in *Proceedings of AI-ED*, vol. 2, 1995. 2.5.2

[133] M. Jug, J. Perš, B. Dežman, and S. Kovačič, "Trajectory based assessment of coordinated human activity," in *International Conference on Computer Vision Systems*, pp. 534–543, Springer, 2003. 2.5.2

[134] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," in *European Conference on Computer Vision*, pp. 556–571, Springer, 2014. 2.5.2, 4.1, 4.1, 4.2, 5.1, 5.1, 5.2

[135] P. Parmar and B. Tran Morris, "Learning to score olympic events," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 20–28, 2017. 2.5.2, 4.1, 4.1, 4.2, 5.1, 5.1, 5.2

[136] 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*, pp. 4489–4497, 2015. 2.5.2

[137] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y.-G. Jiang, and X. Xue, "Learning to score figure skating sport videos," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 12, pp. 4578–4590, 2019. 2.5.2, 4.1, 5.1

[138] J.-H. Pan, J. Gao, and W.-S. Zheng, "Action assessment by joint relation graphs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6331–6340, 2019. 2.5.2, 4.2, 4.3, 5.1, 5.2, 5.3

[139] Y. Tang, Z. Ni, J. Zhou, D. Zhang, J. Lu, Y. Wu, and J. Zhou, "Uncertainty-aware score distribution learning for action quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9839–9848, 2020. 2.5.2, 4.4.1, 4.1, 4.4.1, 4.2, 4.3, 4.4.2, 5.3.1, 5.1, 5.4.1, 5.2, 5.4.2, 5.3

[140] S. Wang, D. Yang, P. Zhai, C. Chen, and L. Zhang, "Tsa-net: Tube self-attention network for action quality assessment," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 4902–4910, 2021. 2.5.2, 4.1, 4.2, 4.4.2, 5.1, 5.2, 5.4.2

[141] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *ICCV*, pp. 2712–2719, 2013. 2.5.3

[142] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *German conference on pattern recognition*, pp. 184–195, Springer, 2014. 2.5.3

[143] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *ICCV*, pp. 4534–4542, 2015. 2.5.3, 6.1

[144] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, pp. 4507–4515, 2015. 2.5.3, 6.1

[145] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction network for video captioning," in *CVPR*, pp. 7622–7631, 2018. 2.5.3, 6.1, 6.4.2

[146] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *European conference on computer vision*, pp. 505–520, Springer, 2014. 3.1, 3.1, 3.4.1

[147] L. Yuan, F. E. Tay, P. Li, L. Zhou, and J. Feng, "Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization," *arXiv preprint arXiv:1904.08265*, 2019. 3.1, 3.4.2, 3.1

[148] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8537–8544, 2019. 3.1, 3.4.2, 3.1, 3.4

[149] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5179–5187, 2015. 3.1, 3.4.1, 3.7a

[150] B. Zhao, X. Li, and X. Lu, "Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7405–7414, 2018. 3.1

[151] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. 3.2

[152] X. Li, J. Song, L. Gao, X. Liu, W. Huang, X. He, and C. Gan, "Beyond rnns: Positional self-attention with co-attention for video question answering," in *The 33rd AAAI Conference on Artificial Intelligence*, vol. 8, 2019. 3.2

[153] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8739–8748, 2018. 3.2

[154] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 3.2, 3.3, 3.4.2, 3.4.2, 3.1, 3.4.3

[155] X. He, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, "Unsupervised video summarization with attentive conditional generative adversarial networks," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2296–2304, 2019. 3.2, 3.4.2, 3.4.2, 3.1

[156] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017. 3.2, 3.3.4

[157] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018. 3.2

[158] Y.-X. Wang and M. Hebert, "Learning to learn: Model regression networks for easy small sample learning," in *European Conference on Computer Vision*, pp. 616–634, Springer, 2016. 3.2

[159] X. Li, H. Li, and Y. Dong, "Meta learning for task-driven video summarization," *IEEE Transactions on Industrial Electronics*, 2019. 3.2, 3.4.2, 3.4.2, 3.1

[160] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank rnn language model," *arXiv preprint arXiv:1711.03953*, 2017. 3.3.3, 3.3.3

[161] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016. 3.3.3

[162] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011. 3.4.1

[163] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, pp. 5288–5296, 2016. 3.4.1, 6.1, 6.4

[164] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 540–555, Springer, 2014. 3.4.1

[165] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *European conference on computer vision*, pp. 540–555, Springer, 2014. 3.4.1

[166] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945. 3.4.1

[167] D. Zwillinger and S. Kokoska, *CRC standard probability and statistics tables and formulae*. Crc Press, 1999. 3.4.1

[168] G. Bertasius, H. Soo Park, S. X. Yu, and J. Shi, "Am i a baller? basketball performance assessment from first-person videos," in *Proceedings of the IEEE international conference on computer vision*, pp. 2177–2185, 2017. 4.1, 5.1

[169] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 4.1, 4.4.3

[170] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, *et al.*, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI workshop: M2cai*, vol. 3, p. 3, 2014. 4.1, 4.4.1, 5.1, 5.4.1

[171] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017. 4.2

[172] 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*, pp. 780–787, 2014. 4.2

[173] 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*, pp. 2127–2136, 2017. 4.2

[174] P. Lei and S. Todorovic, "Temporal deformable residual networks for action segmentation in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6742–6751, 2018. 4.2

[175] J. Li, P. Lei, and S. Todorovic, "Weakly supervised energy-based learning for action segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6243–6251, 2019. 4.2

[176] F. Yi, H. Wen, and T. Jiang, "Asformer: Transformer for action segmentation," *arXiv preprint arXiv:2110.08568*, 2021. 4.2

[177] C. Zhang, A. Gupta, and A. Zisserman, "Temporal query networks for fine-grained video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4486–4496, 2021. 4.2

[178] D. Shao, Y. Zhao, B. Dai, and D. Lin, "Intra-and inter-action understanding via temporal action parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 730–739, 2020. 4.2

[179] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor detr: Query design for transformer-based detector," *arXiv preprint arXiv:2109.07107*, 2021. 4.3.2

[180] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3651–3660, 2021. 4.3.2

[181] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018. 4.2, 4.3, 5.2, 5.3

[182] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, pp. 20–36, Springer, 2016. 4.3, 5.1, 5.3

[183] L. ehovin Zajc, "A modular toolkit for visual tracking performance evaluation," *SoftwareX*, vol. 12, p. 100623, 2020. 4.4.2, 5.4.2

[184] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, vol. 30, 2017. 4.4.3

[185] 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*, pp. 4305–4314, 2015. 5.1

[186] Q. Wang, G. Sun, J. Dong, Q. Wang, and Z. Ding, "Continuous multi-view human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3603–3614, 2021. 5.1

[187] W. Lin, X. Liu, Y. Zhuang, X. Ding, X. Tu, Y. Huang, and H. Zeng, "Unsupervised video-based action recognition with imagining motion and perceiving appearance," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 5.1

[188] H. Wu, X. Ma, and Y. Li, "Spatiotemporal multimodal learning with 3d cnns for video action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1250–1261, 2021. 5.1

[189] N. Nigam, T. Dutta, and H. P. Gupta, "Factornet: Holistic actor, object, and scene factorization for action recognition in videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 976–991, 2021. 5.1

[190] H. Luo, G. Lin, Y. Yao, Z. Tang, Q. Wu, and X. Hua, "Dense semantics-assisted networks for video action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3073–3084, 2021. 5.1

[191] Y. Bai, D. Zhou, S. Zhang, J. Wang, E. Ding, Y. Guan, Y. Long, and J. Wang, "Action quality assessment with temporal parsing transformer," in *European Conference on Computer Vision*, pp. 422–438, Springer, 2022. 5.1, 5.4.1, 5.4.2

[192] M. Prosperi, Y. Guo, M. Sperrin, J. S. Koopman, J. S. Min, X. He, S. Rich, M. Wang, I. E. Buchan, and J. Bian, "Causal inference and counterfactual prediction in machine learning for actionable healthcare," *Nature Machine Intelligence*, vol. 2, no. 7, pp. 369–375, 2020. 5.1

[193] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, "Causal inference for recommender systems," in *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 426–431, 2020. 5.1

[194] D. C. Castro, I. Walker, and B. Glocker, "Causality matters in medical imaging," *Nature Communications*, vol. 11, no. 1, p. 3673, 2020. 5.1

[195] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang, "Counterfactual zero-shot and open-set visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15404–15414, 2021. 5.1, 5.2, 5.3.2, 5.3.4

[196] N. Calderon, E. Ben-David, A. Feder, and R. Reichart, "Docogen: Domain counterfactual generation for low resource domain adaptation," *arXiv preprint arXiv:2202.12350*, 2022. 5.1

[197] J. Y. Halpern, "The book of why, judea pearl. basic books (2018)," *Artif. Intell.*, vol. 277, 2019. 5.1

[198] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *International Conference on Machine Learning*, pp. 2376–2384, PMLR, 2019. 5.2

[199] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, "Grounding visual explanations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 264–279, 2018. 5.2

[200] P. Wang and N. Vasconcelos, "Scout: Self-aware discriminant counterfactual explanations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8981–8990, 2020. 5.2

[201] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning," *arXiv preprint arXiv:1910.01442*, 2019. 5.2

[202] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, "Counterfactual critic multi-agent training for scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4613–4623, 2019. 5.2

[203] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020. 5.2

[204] Z. Fang, S. Kong, C. Fowlkes, and Y. Yang, "Modularized textual grounding for counterfactual resilience," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6378–6388, 2019. 5.2

[205] A. Kanehira, K. Takemoto, S. Inayoshi, and T. Harada, "Multimodal explanations by predicting counterfactuality in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8594–8602, 2019. 5.2

[206] Z. Yue, H. Zhang, Q. Sun, and X.-S. Hua, "Interventional few-shot learning," *Advances in neural information processing systems*, vol. 33, pp. 2734–2746, 2020. 5.2

[207] Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik, "A causal view of compositional zero-shot recognition," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1462–1473, 2020. 5.2

[208] X. Hu, K. Tang, C. Miao, X.-S. Hua, and H. Zhang, "Distilling causal effect of data in class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3957–3966, 2021. 5.2

[209] J. Wu and R. Mooney, "Self-critical reasoning for robust visual question answering," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 5.2

[210] S. Zhang, T. Jiang, T. Wang, K. Kuang, Z. Zhao, J. Zhu, J. Yu, H. Yang, and F. Wu, "Devlbert: Learning deconfounded visio-linguistic representations," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 4373–4382, 2020. 5.2

[211] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Causal intervention for weakly-supervised semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 655–666, 2020. 5.2

[212] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, "Counterfactual samples synthesizing for robust visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10800–10809, 2020. 5.2

[213] D. Teney, E. Abbasnedjad, and A. v. d. Hengel, "Learning what makes a difference from counterfactual examples and gradient supervision," in *European Conference on Computer Vision*, pp. 580–599, Springer, 2020. 5.2

[214] J. Qi, Y. Niu, J. Huang, and H. Zhang, "Two causal principles for improving visual dialog," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10860–10869, 2020. 5.2

[215] X. Yang, H. Zhang, and J. Cai, "Deconfounded image captioning: A causal retrospect," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5.2

[216] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, "Counterfactual vqa: A cause-effect look at language bias," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12700–12710, 2021. 5.2

[217] J. Pearl *et al.*, "Models, reasoning and inference," *Cambridge, UK: Cambridge-UniversityPress*, vol. 19, no. 2, 2000. 5.3.2

[218] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. 5.4.3

[219] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, "Object relational graph with teacher-recommended learning for video captioning," in *CVPR*, pp. 13278–13288, 2020. 6.1, 6.2, 6.1, 6.4.2

[220] Y. Yang, J. Zhou, J. Ai, Y. Bin, A. Hanjalic, H. T. Shen, and Y. Ji, "Video captioning by adversarial lstm," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5600–5611, 2018. 6.1

[221] J. Hou, X. Wu, X. Zhang, Y. Qi, Y. Jia, and J. Luo, "Joint commonsense and relation reasoning for image and video captioning.," in *AAAI*, pp. 10973–10980, 2020. 6.1, 6.2, 6.3.1, 6.1, 6.4.2

[222] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016. 6.1, 6.3.3

[223] D. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 190–200, 2011. 6.1, 6.4

[224] J. S. Park, M. Rohrbach, T. Darrell, and A. Rohrbach, "Adversarial inference for multi-sentence video description," in *CVPR*, pp. 6598–6608, 2019. 6.2

[225] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, pp. 5767–5777, 2017. 6.2, 6.3.3

[226] B. Pan, H. Cai, D.-A. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-temporal graph for video captioning with knowledge distillation," in *CPVR*, pp. 10870–10879, 2020. 6.2, 6.1, 6.4.2

[227] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018. 6.3.1

[228] S. Zhang, X. He, and S. Yan, "Latentgnn: Learning efficient non-local relations for visual recognition," in *ICML*, pp. 7374–7383, PMLR, 2019. 6.3.2

[229] Y. Chen, S. Wang, W. Zhang, and Q. Huang, "Less is more: Picking informative frames for video captioning," in *ECCV*, pp. 358–373, 2018. 6.1

[230] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory-attended recurrent network for video captioning," in *CVPR*, pp. 8347–8356, 2019. 6.1, 6.4.2

[231] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002. 6.4.1

[232] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation*, pp. 376–380, 2014. 6.4.1

[233] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CPVR*, pp. 4566–4575, 2015. 6.4.1

[234] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004. 6.4.1

[235] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017. 6.4.1

[236] 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*, pp. 6299–6308, 2017. 6.4.1

[237] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6836–6846, 2021. 7.2.2

[238] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3202–3211, 2022. 7.2.2

APPENDIX A

Statements of Authorship

Statement of Authorship for the joint/multi-authored paper in **Chapter 3: Query Twice: Dual Mixture Attention Meta Learning for Video Summarization.**

Title of Paper	Query Twice: Dual Mixture Attention Meta Learning for Video Summarization
Publication Status	Published
Publication Details	Published in MM'20: Proceedings of the 28th ACM International Conference on Multimedia October 2020, Pages 4023–4031. Junyan Wang, **Yang Bai**, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, Xiaolin Wei

Student Confirmation

Student Name	Yang Bai	
Contribution to the paper	1. Conception of the idea 2. Implementation of Meta-learning component 3. Writing part of the paper	
Signature		26/09/2023