THE UNIVERSITY OF SYDNEY

DOCTORAL THESIS

Deep Learning for Spatial and Temporal Video Localization

Author:
Rui SU

Supervisor:
Prof. Dong XU

Co-Supervisor:
Dr. Wanli OUYANG

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

School of Electrical and Information Engineering
Faculty of Engineering

June 17, 2021

Abstract of thesis entitled

Deep Learning for Spatial and Temporal Video Localization

Submitted by

Rui SU

for the degree of Doctor of Philosophy

at The University of Sydney

in June, 2021

In this thesis, we propose to develop novel deep learning algorithms for the video localization tasks including spatio-temporal action localization, temporal action localization, spatio-temporal visual grounding, which require to localize the spatio-temporal or temporal locations of targets from videos.

First, we propose a new Progressive Cross-stream Cooperation (PCSC) framework for the spatio-temporal action localization task. The basic idea is to utilize both spatial region (*resp.*, temporal segment proposals) and features from one stream (*i.e.*, the Flow/RGB stream) to help another stream (*i.e.*, the RGB/Flow stream) to iteratively generate better bounding boxes in the spatial domain (*resp.*, temporal segments in the temporal domain). By first using our newly proposed PCSC framework for spatial localization and then applying our temporal PCSC framework for temporal localization, the action localization results are progressively improved.

Second, we propose a progressive cross-granularity cooperation (PCG-TAL) framework to effectively take advantage of complementarity between the anchor-based and frame-based paradigms, as well as between two-view clues (*i.e.*, appearance and motion) for the temporal action

localization task. The whole framework can be learned in an end-to-end fashion, whilst the temporal action localization performance can be gradually boosted in a progressive manner.

Finally, we propose a two-step visual-linguistic transformer based framework called STVGBert for the spatio-temporal visual grounding task, which consists of a Spatial Visual Grounding network (SVG-net) and a Temporal Boundary Refinement network (TBR-net). Different from the existing works for the video grounding tasks, our proposed framework does not rely on any pre-trained object detector. For all our proposed approaches, we conduct extensive experiments on publicly available datasets to demonstrate their effectiveness.

Deep Learning for Spatial and Temporal Video Localization

by

Rui SU
B.E. *South China University of Technology* M.Phil. *University of Queensland*

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

at

University of Sydney
June, 2021

Declaration

I, Rui SU, declare that this thesis titled, "Deep Learning for Spatial and Temporal Video Localization", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed: _____

Date: June 17, 2021

For Mama and Papa

Acknowledgements

First, I would like to thank my supervisor, Prof. Dong Xu, for his valuable guidance and support during the past four years of my Ph.D. study. Under his supervision, I have learned useful research techniques including how to write high quality research papers, which help me to becoman an independent researcher.

I would also like to specially thank Dr. Wanli Ouyang and Dr. Luping Zhou, who offer me valuable suggestions and inspirations in our collaborated works. With the useful discussions with them, I work out many difficult problems and am also encouraged to explore unknowns in the future.

Last but not least, I sincerely thank my parents and my wife from the bottom of my heart for everything.

Rui SU
University of Sydney
June 17, 2021

List of Publications

JOURNALS:

[1] **Rui Su**, Dong Xu, Luping Zhou, and Wanli Ouyang, "Progressive Cross-stream Cooperation in Spatial and Temporal Domain for Action Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[2] **Rui Su**, Dong Xu, Lu Sheng, and Wanli Ouyang, "PCG-TAL: Progressive Cross-granularity Cooperation for Temporal Action Localization," *IEEE Transactions on Image Processing*, 2020.

CONFERENCES:

[1] **Rui SU**, Wanli Ouyang, Luping Zhou, and Dong Xu, "Improving action localization by progressive cross-stream cooperation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12016-12025, 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the massive use of digital cameras, smart phones, and webcams, a large amount of videos data are available on the Internet, which brings the increasing demands of an intelligent system to help users analyze the contents of the videos. In this context, video localization, as one of the most discussed video analysis problems, has attracted more and more attentions from both academics and industries in recent decades.

In this thesis, we mainly investigate video localization in three tasks: (1) spatio-temporal action localization; (2) temporal action localization; (3) and spatio-temporal visual grounding. Specifically, given an untrimmed video, the spatio-temporal action localization task requires to find out the spatial coordinates (bounding boxes) of action instances in every action frames (frames that contain action instances), which are then associated into action tubes in order to indicate the spatio-temporal locations for action instances. For the temporal action localization task, it aims to only capture the starting and the ending time points of action instances within the untrimmed videos without caring about the spatial locations of the action instances. Given an untrimmed video with textual queries describing objects appearing within the video, the goal of the spatio-temporal visual grounding task is to localize the queried target objects from the given video by outputting the corresponding spatial coordinates (bounding boxes). We develop novel algorithms by using a deep neural network to solve the aforementioned tasks, and conduct extensive experiments to evaluate our proposed methods. A brief introduction for our contributions to these three tasks are provided as follows.

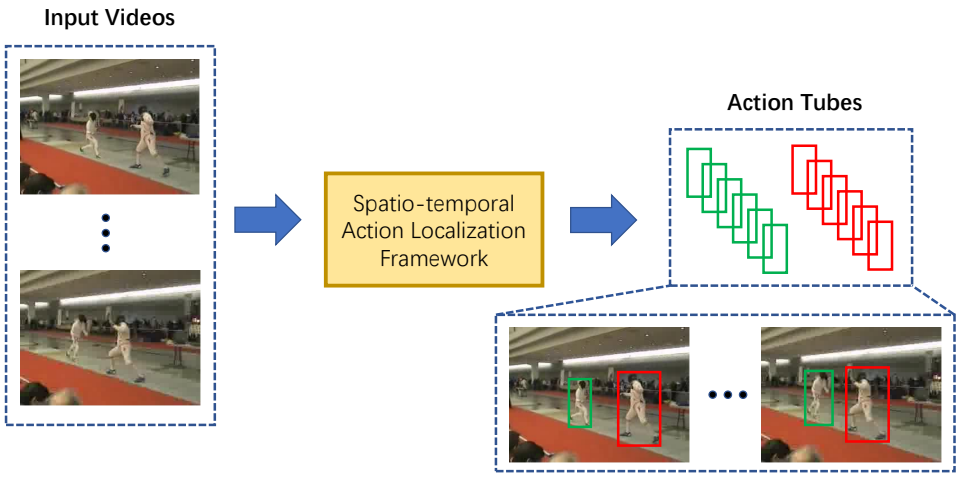


Figure 1.1: Illustration of the spatio-temporal action localization task. The spaito-temporal action localization framework takes untrimmed videos as input and generates the required action tubes

1.1 Spatial-temporal Action Localization

Recently, deep neural networks have significantly improved performance in various computer vision tasks [20, 53, 51, 52, 73, 43, 100] including human action detection. Human action detection, also known as the spatio-temporal action localization, aims to recognize the actions of interest presented in videos and localize them in space and time. As shown in Figure 1.1, for the spatio-temporal action localization task, the input is the untrimmed videos while the output is the corresponding action tubes (*i.e.*, a set of bounding boxes). Due to its wide spectrum of applications, it has attracted increasing research interests. The cluttered background, occlusion and large intra-class variance make spatio-temporal action localization a very challenging task.

Existing works [64, 60, 54] have demonstrated the complementarity between appearance and motion information at the feature level in recognizing and localizing human actions. We further observe that these two types of information are also complementary to each other at the region proposal level. That is, either appearance or motion clues alone may succeed or fail to detect region proposals in different scenarios. However, they can help each other by providing region proposals to each

other.

For example, the region proposal detectors based on appearance information may fail when certain human actions exhibit extreme poses (*e.g.*, toothbrushing, where only the movement of the hand changes while the whole scene remains the same), but motion information from human movements could help capture human actions. In another example, when the motion clue is noisy because of subtle movement or cluttered background, it is hard to detect positive region proposals based on motion information while the appearance clue may still provide enough information to successfully find candidate action regions and remove a large amount of background or non-action regions. Therefore, we can use the bounding boxes detected from the motion information as the region proposals to improve action detection results based on the appearance information, and vice versa. This underpins our work in this paper to fully exploit the interactions between appearance and motion information at both region proposal and feature levels to improve spatial action localization, which is our first motivation.

On the other hand, to generate spatio-temporal action localization results, we need to form action tubes by linking the detected bounding boxes for actions in individual frames and temporally segment them out from the entire video clip. Data association methods based on the spatial overlap and action class scores are mainly used in the current works [18, 54, 60, 67]. However, it is difficult for such methods to precisely identify the temporal boundaries of actions. For some cases, due to the subtle difference between the frames near temporal boundaries, it remains an extremely hard challenge to precisely decide the temporal boundary. As a result, it produces a large room for further improvement of the existing temporal refinement methods, which is our second motivation.

Based on the first motivation, in this paper, we propose a progressive framework called Progressive Cross-stream Cooperation (PCSC) to iteratively use both region proposals and features from one stream to progressively help learn better action detection models for another stream. To exploit the information from both streams at the region proposal level, we collect a larger set of training samples by combining the latest region

proposals from both streams. At the feature level, we propose a new message passing module to pass information from one stream to another stream in order to learn better representations by capturing both the motion and the appearance information at the same time. By exploiting the complementality information between appearance and motion clues in this way, we can progressively learn better action detection models and improve action localization results at the frame-level.

Based on the second motivation, we propose a new temporal boundary refinement method consisting of an actionness detector and an action segment detector, in which we can extend our PCSC framework to temporal localization. For the actionness detector, we train a set of class-specific binary classifiers (actionness detectors) to detect the happening of a certain type of actions. These actionness detectors are trained by focusing on “confusing" samples from the action tube of the same class, and therefore can learn critical features that are good at discriminating the subtle changes across the action boundaries. Our approach works better when compared with an alternative approach that learns a general actionness detector for all actions. For the action segment detector, we propose a segment proposal based two-stream detector using a multi-branch architecture to address the large temporal scale variation problem. Our PCSC framework can be readily applied to the action segment detector to take advantage of the complementary information between appearance and motion clues to detect accurate temporal boundaries and further improve spatio-temporal localization results at the video-level.

Our contributions are briefly summarized as follows:

- For spatial localization, we propose the Progressive Cross-stream Cooperation (PCSC) framework to iteratively use both features and region proposals from one stream to help learn better action detection models for another stream, which includes a new message passing approach and a simple region proposal combination strategy.
- We also propose a temporal boundary refinement method to learn

class-specific actionness detectors and an action segment detector that applies the newly proposead PCSC framework from the spatial domain to the temporal domain to improve the temporal localization results.

- Comprehensive experiments on two benchmark datasets UCF-101-24 and J-HMDB demonstrate that our approach outperforms the state-of-the-art methods for localizing human actions both spatially and temporally in realistic scenarios.

1.2 Temporal Action Localization

Temporal action localization aims at simultaneously classifying the actions of interest and localizing the starting and ending frames of every action instance in untrimmed videos. Many real-world applications, such as abnormal event detection, only require to localize the actions of interest in the temporal domain. Unlike the spatio-temporal action localization task, as the temporal action localization task dose not localize the actors in each frame, it can only detect the temporal starting and ending frames based on the whole input image from each frame. As a result, the input is less discriminative, which makes the temporal action localization task more challenging.

Similar as the object detection methods like [58], the prior methods [9, 15, 85, 5] have been proposed to handle this task through a two-stage pipeline, which at first generates a set of 1D temporal segment proposals and then performs classification and temporal boundary regression on each individual proposal. However, this task is still challenging and we are confronted with two major issues: (1) how to generate the proposals with high recall rates and accurate boundaries, and (2) how to effectively make complementary clues cooperate with each other across the whole video localization framework.

For the first issue, as shown in Figure 1.2, we observe that the proposal generation methods with different granularities (*i.e.*, the anchor-based and frame-based methods) are often complementary to each other.

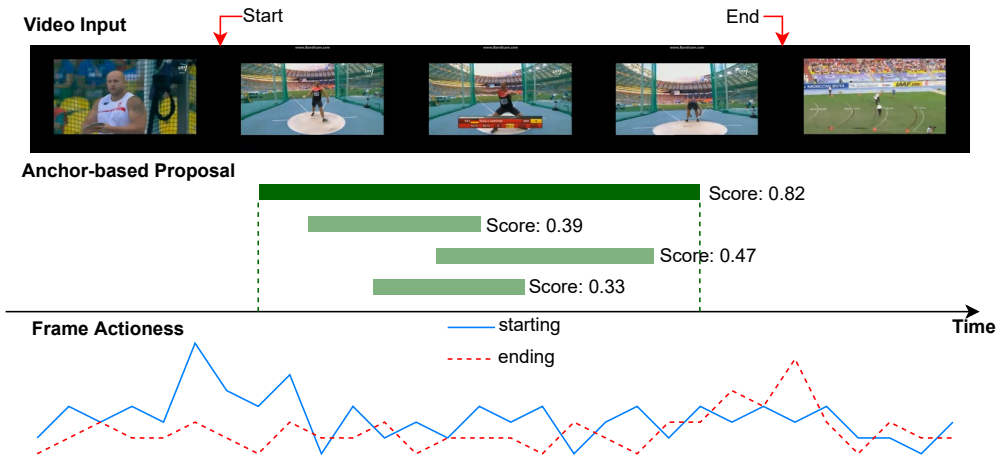


Figure 1.2: Illustration of the temporal action localization task. The starting and the ending frames of the actions within the input videos can be localized by the anchor-based or the frame-based temporal action localization methods.

Specifically, the anchor-based methods generate segment proposals by regressing boundaries for the pre-defined anchors, which generally produce coarse proposals with a high recall rate. Meanwhile, the frame-based methods predict per-frame actionness scores to generate segment proposals, which often detect accurate boundaries but suffer from low recall rates. The prior works [37, 13, 46] have attempted to make these two lines of schemes cooperate with each other by using either bottom-up module stacking strategy or guided proposal filtering. But we argue that the features from the anchor-based methods and the frame-based methods represent different information of the input videos as they are learned based on different proposal generation mechanisms. Consequently, the features and segment proposals from the two lines of proposal generation schemes are considered as complementary to each other for the temporal action localization task, and it is possible to leverage this cross-granularity-level complementary information to generate better action segments with high recall rates and accurate temporal boundaries.

For the second issue, the aforementioned two lines of proposal generation methods can cooperate with each other at both the feature and

segment levels. To be more specific, the feature-level cooperation strategy will improve the frame-based features and filter out false starting/ending point pairs with invalid durations, and thus enhance the frame-based methods to generate better proposals having higher overlapping areas with the ground-truth action segments. Meanwhile, the proposal-level cooperation strategy will collect complete and accurate proposals with the aid of the frame-based approaches, which can help the subsequent action classification and boundary regression process. In order to effectively integrate complementary information between the anchor-based and frame-based methods, one straightforward approach is to progressively leverage the cross-granularity complementarity to improve the temporal action localization performance, and eventually fully exploit the complementary information. However, it is suboptimal by trivially repeating the aforementioned cross-granularity cooperation process as it can easily lead to performance saturation. Building upon the well-known observation that appearance and motion clues are also complementary to each other, our work augments the cross-granularity collaboration process with the cross-stream cooperation strategy, which enhances the representations for the anchor-based methods and better exploits the complementarity between the two-granularity approaches at the feature level by iteratively using the features from one stream (*i.e.*, RGB/flow) to improve the features from another stream (*i.e.*, flow/RGB) for the anchor-based method.

To this end, building upon two-granularity and two-stream pipeline, we propose a unified temporal action localization framework named as PCG-TAL, which enables progressive cross-granularity cooperation. More specifically, our PCG-TAL includes a new Anchor-Frame Cooperation (AFC) module to improve the frame-based features by using a message passing operation to pass complementary information from the anchor-based features and also update the segment proposal set by combining the RGB and flow segment proposals generated from the anchor-based scheme and the improved segment proposals from the frame-based approach based on the updated frame-based features. Moreover, our AFC module incorporates the aforementioned message passing and segment proposal combination strategies into the two-stream anchor-based

segment proposal generation pipeline, in which the RGB-stream AFC module and the flow-stream AFC modules are stacked sequentially in order to exchange cross-stream and cross-granularity information at the feature and segment proposal levels over multiple stages, respectively. Specifically, we iteratively update the RGB-stream AFC module by using the updated features and proposals from the flow-stream AFC module, and vice versa. As a result, our framework can progressively improve the temporal action localization performance by leveraging the cross-granularity and cross-stream complementary information. The entire framework is learned in an end-to-end fashion. Our contributions are three-fold:

(1) We propose an AFC module to exploit the cross-granularity and cross-stream complementarity at both feature and segment proposal levels.

(2) We introduce a new multi-stage framework to effectively integrate the cross-granularity and cross-stream information from the two-stream anchor-based and frame-based methods in a progressive manner.

(3) Comprehensive experiments on three benchmark datasets, such as THUMOS14, ActivityNet v1.3 and UCF-101-24, demonstrate that our approach outperforms the state-of-the-art methods for both temporal and spatial-temporal action localization.

1.3 Spatio-temporal Visual Grounding

Vision and language play important roles for human to understand the world. In recent years, with the remarkable progress of deep neural networks, various vision-language tasks (e.g., image captioning and visual grounding) have attracted increasing attention from researchers.

Spatial-Temporal Video Grounding (STVG), which was introduced in the recent work [102], is a novel and challenging vision-language task. Given an untrimmed video and a textual description of an object, the STVG task aims to produce a spatio-temporal tube (*i.e.*, a sequence of

bounding boxes) for the target object described by the given text description. Different from the existing grounding tasks in images, both spatial and temporal localizations are required in the STVG task. Besides, how to effectively align visual and textual information through cross-modal feature learning in both spatial and temporal domains is also a key issue for accurately localizing the target object, especially in the challenging scenarios where different persons often perform similar actions within one scene.

Spatial localization in images/videos is a related visual grounding task, and spatial localization results have been improved in recent works [42, 96, 45, 82, 87, 88, 41, 86, 7]. In most existing works, a pre-trained object detector is often required to pre-generate object proposals. However, these approaches suffer from two limitations: (1) The localization performance heavily rely on the quality of the pre-generated object proposals. (2) It is difficult for a pre-trained object detector to be well generalized to any new datasets with unseen classes. Although the recent works [92, 34, 48, 91] have attempted to remove the dependency of the pre-generation process in the image grounding tasks, such efforts have not been made for the video grounding tasks.

Beyond spatial localization, temporal localization also plays an important role for the STVG task, in which significant progress has been made in recent years. The existing methods can be mainly divided into anchor-based and frame-based approaches. In the anchor-based approaches [14, 1], the output segments often have relatively high overlap with the ground-truth ones but the temporal boundaries are less accurate because they regress the ground-truth boundaries by using the features extracted from the anchors (i.e., the candidate segments). The frame-based approaches [6] directly determine the temporal boundaries based on the frame-level prediction scores, which often produce more accurate temporal boundaries, but may falsely detect the temporal segments. Intuitively, it is desirable to explore the complementarity of the two lines of methods in order to improve the temporal localization performance.

Motivated by the above observations, in this work, we propose a visual-linguistic transformer based framework called STVGBert for the

STVG task, which includs a Spatial Visual Grounding network (SVG-net) and a Temporal Boundary Refinement network (TBR-net). The SVG-net first takes a pair of video clip and textual query as the input and produces an initial spatio-temporal tube. The initial tube is then fed into the TBR-net, where its temporal boundaries are further refined. Both SVG-net and TBR-net employ a visual-linguistic transformer for cross-modal feature learning given its promising results achieved for various vision-language tasks [47, 70, 33].

Furthermore, both SVG-net and the TBR-net have their special designs. Specifically, the key component of the SVG-net is an improved version of ViLBERT [47], named ST-ViLBERT. In addition to encoding temporal information as in the original ViLBERT, our ST-ViLBERT also preserves spatial information in the visual input feature. As a result, our SVG-net can effectively learn the cross-modal representation and produce the initial spatio-temporal tubes with reasonable quality without requiring any pre-trained object detectors. Besides, we also propose a novel ViLBERT-based temporal localization network referred to as TBR-net, to take advantage of the complementarity between an anchor-based branch and a frame-based branch. Our TBR-net iteratively refines the temporal boundaries by using the output from the frame/anchor-based branch as the input for the anchor/frame-based branch. This multi-stage two-branch design turns out to be effective for progressively refining the temporal boundaries of the initial tube produced by the SVG-net. We evaluate our proposed framework STVGBert on two benchmark datasets, VidSTG [102] and HC-STVG [76], and our framework outperforms the state-of-the-art methods.

Our contributions can be summarized as follows:

(1) We propose a novel two-step visual-linguistic transformer based framework STVGBert for the spatio-temporal video grounding task. To the best of our knowledge, this is the first STVG framework that does not require any pre-trained object detectors.

(2) We introduce a spatial visual grounding network (SVG-net) with a newly proposed cross-modal feature learning module. Moreover,

we propose a temporal boundary refinement network (TBR-net), which progressively refines the temporal boundaries of the object tubes by exploiting the complementarity of the frame-based and anchor-based methods.

(3) Comprehensive experiments conducted on two benchmark datasets, VidSTG and HC-STVG, demonstrat the effectiveness of our framework for the STVG task.

1.4 Thesis Outline

The rest of this thesis is organized into five chapters. We summarize the main content of each chapter as follows:

Chapter 2. Literature Review. In this chapter, we give a comprehensive review on the background and the three video localization tasks to help readers better understand the studies in the following chapters.

Chapter 3. Progressive Cross-stream Cooperation in Spatial and Temporal Domain for Action Localization. In this chapter, we propose a Progressive Cross-stream Cooperation framework in both spatial and temporal domains to solve the spatio-temporal action localization task by leveraging the complementary information between appearance and motion clues at the proposal and feature levels. We also evaluate the effectiveness of our proposed method on two benchmarks, UCF-101-24 [68] and J-HMDB [24].

Chapter 4. PCG-TAL: Progressive Cross-granularity Cooperation for Temporal Action Localization. In this chapter, in order to solve the temporal action localization task, we integrate two lines of previous work (*i.e.*, anchor-based and frame-based methods) and propose a new Anchor-Frame Cooperation (AFC) module to exchange cross-granularity information along with a two-stream cooperation strategy to encourage collaboration between the complementary appearance and motion clues. The effectiveness of our proposed method is validated on three benchmarks, THUMOS14 [25], ActivityNet v1.3 [3] and UCF-101-24 [68].

Chapter 5. STVGBert: A Visual-linguistic Transformer based Framework for Spatio-temporal Video Grounding. In this chapter, we develop a spatio-temporal visual grounding framework based on a visual-linguistic transformer to learn cross-modality representation in order to better align the visual and textual information. we also propose a temporal boundary refinement network to progressively refine the temporal boundaries of the generated results by leveraging the complementarity of the frame-based and the anchor-based methods. Two benchmarks, VidSTG [102] and HC-STVG [76], are used to evaluate our proposed method.

Chapter 6. Conclusions and Future Work. In this chapter, we conclude the contributions of this work and point out the potential directions of video localization in the future.

Chapter 2

Literature Review

In this chapter, we review the related work on the deep neural network and three tasks studied in this thesis. We first introduce the background of the deep neural network, and then summarized related work on the spatio-temporal action localization, temporal action localization and spatio-temporal visual grounding tasks.

2.1 Background

2.1.1 Image Classification

Recently, deep neural network has been widely used in the computer vision task. Krizhevsky *et al.* [27] used a set of convolutional layers with nonlinear functions as activation functions inbetween each layer to build a convolutional neural network named AlexNet for the image classification task. Each convolutional layer contains a set of weights, and the number of weights in each layer is depended on the kernel size used in the convolutional filters. The convolutional neural network can be trained to classify the categories for images by using a set of training samples. Specifically, the loss of ground truth and prediction for each training sample is calculated, and the weights of the convolutional neural network are updated by using back-propagation and gradient descent to minimize the loss. Additionally, due to the gradient vanishing issue cause by sigmoid function, a new nonlinear function named Rectified Linear Units (Relu) is introduced and used as the activation functions to address this issue. However, the large number of parameters

(weights) used in the convolutional neural network can easily lead the network overfit the training samples. In order to ensure the generalization for the network, strategies such as data augmentation and dropout are introduced to avoid the overfitting problem. This network achieved a winning top-5 test error rate of 15.3% in the lagest image classification competition (ILSVRC-2012) at that time, which was a huge improvement from the second best (26.2%). This work has activated the research interest of using deep learning method to solve computer vision problem.

Later, Simonyan and Zisserman [65] proposed a deeper convolutional neural network named VGG-Net to further improve the image classification performance. They observed that convolutional layers with large kernel size were used in AlexNet, which can be replaced by a stack of several convolutional layers with low kernel size. For example, a stack of two 3×3 convolutional layers can achieve the same effective receptive field of a 5×5 convolutional layer and the effective receptive field of a 7×7 convolutional layer can be achieved by stacking three 3×3 convolutional layers. By decomposing the convolutional layers with large kernel size into several 3×3 convolutional layers with activation functions inbetween, the nonlinearity of the convolutional neural network increases and the number of parameters used in the network decreases without losing the effective receptive field of the network. VGG-Net achieved 7.32% top-5 error rate in the ILSVRC-2014 competition.

Similarly, Szegedy *et al.* [74] proposed a deep convolutional neural network named Inception for the image classification task, which achieved 6.67% top-5 error rate and defeated VGG-Net in the ILSVRC-2014 competition. In stead of decomposing convolutional layers with large kernel size into several convolutional layers with low kernel size, Inception employs multiple kernel sizes in each convolutional layer in order to take advantage of the complementary information brought by multiple resolutions. Specifically, they introduced an inception module which takes the output of previous incpeption module as the input of a set of parallel convolutional layers with different kernel sizes followed by pooling layers, and the output of these layers are concatenated to form the final output for the inception module. The deep convolutional

neural network is built by stacking such inception modules to achieve high image classification performance.

VGG-Net and Inception have shown that performance on the image classification problem can be improved by making the convolutional neural network deeper. However, when simply increasing the number of the convolutional layers and stacking them to build the network, the image classification performance easily gets saturated and then degrades rapidly. This is because as the convolutional neural network goes deeper, the error gradients accumulate to a very large value. This results in large update during training process, and leads to an unstable network. In order to address the gradient explosion issue, He *et al.* [20] proposed a shortcut connection architecture to allow the gradients to be easily back-propageted from the deep layers to the shallow layers without gradient accumulation. By using the shortcut connection architecture, the convolutional neural network can be built with extremely high depth to improve the image classification performance.

Instead of increasing the depth of the convolutional neural network, Huang *et al.* [22] introduced another shortcut connection architecture to enlarge the width of the convolutional neural network. The proposed network is built by stacking a set of dense blocks, with each dense block has shortcut connections with other blocks. Specifically, for each dense block, it takes the combination of the output from all the previous dense blocks as the input so that the features from each layers are able to be reused in the following layers. Note that the network becomes wider as it goes deeper.

2.1.2 Object Detection

The object detection task aims to detect the locations and the categories of the objects appearing in the given image. With the promising image classification performance achieved by using deep convolutional neural network, recently, researchers have investigate how to use deep learning methods to solve the object detection problem.

Girshick *et al.* [17] proposed a convolutioal neural network based object detection method called RCNN. They used selective search to generate a set of region proposals, which were then used to crop the given image to obtain the corresponding patches of the input image. All the cropped patches were the candidates, which potentially contains objects. Each of these patches was fed into a convolutional neural network followed by a detection head, which contains a classification head and a regression head to predict the category of the patch and regress the offsets to the coordinates of the corresponding region proposal, respectively. However, one of the drawback of RCNN is that all the generated patches have to be fed into the convolutional neural network, which leads to high computational cost and large training and inference time.

To make the convolutional neural network based object detection frameworks more efficient, Girshick *et al.* [16] proposed Fast RCNN to improve the time cost for detecting objects in a given image. Specifically, similar to RCNN, they first used selective search to generate region proposal set. However, instead of generating patch candidates from the original image to be fed into the convolutional neural network, they only fed the entire image to the network to generate a feature map, and then applied a newly proposed Region-of-Interest (ROI) pooling operation on top of the generated feature map to obtain fixed size feature maps for the corresponding region proposals so that they can be passed to a fully connected layer followed by a detection head to produce the category prediction and the offsets. By doing so, the training and inference time have been largely reduced.

Both RCNN and Fast RCNN use selective search to generating region proposals, which is very time-consuming and affects the object detection performance. As a result, in order to further improve the efficiency of the object detection frameworks, Ren *et al.* [58] introduced a new Region Proposal Network (RPN) that can learn to generate region proposals, and proposed a new object detection framework Faster RCNN. Similar to Fast RCNN, the entire image was fed into a convolutional neural network to generate a feature map. A set of pre-defined

anchors were used at each location of the generated feature map to generate region proposals by a separate network. The final detection results were obtained from a detection head by using the pooled features based on the ROI pooling operation on the generated feature map.

Faster RCNN is considered as a two-stage objected detection framework where a RPN is first used to generate region proposals and then ROI pooling operation and detection head are applied to obtain the final results. Redmon *et al.* [57] proposed a one-stage real-time object detection framework called Yolo. They split the entire image into an $S \times S$ grid, with each location in the gird contained several pre-defined anchors. For each anchor, the network predicted the class probability and the offsets to the anchor. Yolo is faster than Faster RCNN as it does not require region feature extraction by using ROI pooling operation, which is time-consuming.

Similarly, Liu *et al.* [44] proposed a single shot multi-box detector (SSD) to generate bounding boxes in one stage. In [44], a set of boxes with different sizes and aspect ratios are predefined at each locations of feature maps. These predefined boxes were used to directly regress the coordinates of the target objects based on the features at the corresponding locations on the feature maps.

All the aforementioned object detection frameworks are anchor-based methods, which requires pre-defined anchors or generated region proposals. Zhou *et al.* [104] proposed CenterNet for object detection without using any pre-defined anchors, which predicted the locations of the center points of the objects. Specifically, a dense heat map was predicted by the convolutional neural network to represent the probability of each point in the heat map being the center point of an object. Additionally, the network also predicted the bounding box size (height and width) for each point in the heat map.

2.1.3 Action Recognition

Convolutional neural network has been used to solve image computer vision problems such as classification, object detection. Recently, the research interest of using convolutional neural network to address computer vision problems on videos has been also raised. Simonyan and Zisserman [64] introduced Two-Stream Convolutional Networks for the action recognition task in video. To classify the action categories for videos, they designed a spatial network based on the convolutional neural network, which took the RGB frames from the given videos as input and outputted the action class predictions. In addition to the appearance information, they also found that a stack of multi-frame dense optical flow can represent motion information, which also helped the action recognition performance. Specifically, a separate temporal network was designed to take the optical flow maps as input to prediction action classes, which can also achieve good action recognition performance. Moreover, combining spatial and temporal networks can have further improvement on the performance of the action recognition task, which demonstrated that appearance and motion information is complementary to each other.

For the work in [12], Feichtenhofer *et al.* further discussed the effects of several different strategies for combining the spatial and the temporal networks proposed in [64] on the action recognition performance. By setting up experiments, they concluded that among the proposed strategies, by using convolutional layer to fuse the features from the last convolutional layer of the spatial and the temporal networks, the combined network can achieve the best action recognition performance, which provided the insights of how to effectively combine information from different complementary modalities.

Based on the two-stream architecture in [64], Wang *et al.* [81] proposed Temporal Segment Networks to improve the performance for action recognition. They claimed that temporal information is important but consecutive frames from a video are redundant because of their high similarity. As a result of that, they divided a video into segments and

randomly chose a frame from each segment to represent the whole segment. The chosen frames and corresponding optical flow maps were used as input to the two-stream network and then consensus was applied on output for each stream. Finally, the output from each stream were combined to consider the complementary information between two modalities. Additionally, they also took advantage of features extracted from warped flow and RGB difference, which can further benefits the action recognition performance. Besides, it can help the framework from overfitting the training samples by using pretrained model as initialization and applying partial batch normalization.

2D convolutional neural networks has proven its effective usage on the action recognition task. Tran *et al.* [77] also investigated the effect of 3D convolution on feature extraction for videos. They claimed that while 2D convolution only consider spatial information, 3D convolution is able to take advantage of temporal information of the input, which is also important in a video. They proved that 3D convolution with kernel size of 3x3x3 is a reasonable setting among several proposed settings and designed a 3D convolutional network with such convolutional layers named C3D. Compared to the two-stream framework, 3D convolutional neural network can model the spatio-temporal information simultaneously, which is more time-efficient.

Similar to C3D, Carreira *et al.* [4] proposed a deeper 3D convolutional neural network called I3D for the action recognition task. I3D was built by inflating every 2D convolutional layers in Inception [74] to 3D convolutional layers to learn spatio-temporal representations for videos. They trained the proposed network on a large action recognition dataset named Kinetics and demonstrated that it can help significantly improve the action recognition performance on other relatively small dataset such as UCF-101. I3D is widely used to extract spatio-temporal features for videos.

2.2 Spatio-temporal Action Localization

Spatio-temporal action localization aims to localize action instances within untrimmed videos. The input of this task is an untrimmed video while the output is a set of bounding boxes representing the spatial locations in each frame within the temporal durations of the corresponding action instances. Recently, deep convolutional neural network has been widely used to solve this computer vision task.

Weinzaepfel *et al.* [83] proposed a spatio-temporal action localization framework built upon two-stream (RGB and optical flow) features. They used object detection methods to detect action locations for each frame and then tracked the proposals with high scores by using tracking-be-detection methods. A multi-scale slide-window approach was then used as the temporal anchors to detect the temporal segments to perform the temporal localization.

Similar to action recognition, both appearance and motion clues are important for spatio-temporal action localization. Peng *et al.* [54] proposed a multi-region two-stream RCNN framework for spatio-temporal action localization. In addition to a RCNN framework that takes RGB images as input to generate region proposals, they also developed an optical flow based RCNN framework and proved that it can generate high quality proposals by using motion information. Moreover, instead of detecting the human body as a whole, they introduced a multi-region scheme in the RCNN framework to generate proposals for different regions of human body so that complementary information can be added on human body parts. The proposed framework generated frame-level detections and Viterbi algorithm was used to link these detections for temporal localization.

Saha *et al.* [60] proposed a three-stage framework for detecting the spatio-temporal locations of action instance. They first took advantage of both appearance (RGB) and motion (optical flow) information to localize and predict action scores. Then the appearance and motion scores were combined to leverage the complementary information from these

two modalities. Finally, action tubes were constructed by linking the detections generated from the previous stages. The detections were first linked based on their action class scores and spatial overlap, and then the linked action tubes were temporally trimmed via ensuring label consistency.

As computing optical flow from consecutive RGB images is very time-consuming, it is hard to achieve real-time performance for the two-stream based spatio-temporal action localization methods. To address this issue, Singh *et al.* [67] used a real-time one-stage object detection method to replace the RCNN framework used in [54] and [60] in order to reduce the additional time consumption brought by the two-stage object detection framework. Additionally, they also introduced an efficient online algorithm to incrementally link the generated frame-level detections and construct action tubes. By doing so, the proposed method is capable of performing real-time spatio-temporal action localization.

The aforementioned spatio-temporal action localization methods rely on frame-level detections. Kalogeiton *et al.* [26] proposed an ACT-detector to consider the temporal information of videos. Instead of handling one frame at a time, ACT-detector took a sequence of RGB frames as input and outputted a set of bounding boxes to form a tubelet. Specifically, convoulutional neural network was used to extract features for each of these RGB frames, which were stacked to form representation for the whole segment. The representation was then used to regress the coordinates of the output tubelet. The generated tubelets were more robust than the linked action tube from frame-level detections because they were produced by leveraging the temporal continuity of videos.

A progressive learning framework for spatio-temporal action localization named STEP [89] was proposed by Yang *et al.*. They first used several hand-designed coarse anchors to search for action of interests and then progressively refined the coordinates of the anchors through iterations in a coarse-to-fine manner. In each iteration, the anchors were temporally extended to gradually include more temporal context in order to generate high quality proposals. Moreover, by doing so, STEP was able to naturally handle the spatial displacement within action tubes.

2.3 Temporal Action Localization

Unlike spatio-temporal action localization, which requires to obtain the spatial locations of the action instances, temporal action localization focuses on accurately detecting when the action instances start and end in the time domain of the untrimmed videos. Early works used sliding windows to generate segment proposals and then apply the classifiers to classify actions for each proposal. Yuan *et al.* [97] captured motion information at multiple resolutions by using sliding windows to obtain a Pyramid of Score Distribution Feature. They then considered the inter frame consistency by incorporating the pyramid of score distribution feature, which effectively boosted the temporal action localization performance. Shou *et al.* [63] used sliding window and C3D [77] as a binary classification task to generate background and action proposals. All these proposals were fed to a classification network based on C3D to predict labels. Then, they used the trained classification model to initialize a C3D model as a localization model, and took proposals and its overlap score as inputs of this localization model and proposed a new loss function to reduce the classification error.

Most recent temporal action localization methods can be roughly categorized into two types. The first set of works used [49, 10, 29, 98, 103] convolutional neural networks to extract frame-level or snippet-level features for the given videos, which were then used to predict frame-wise or snippet-wise actionness scores. By simply thresholding the actionness scores, the action starting or ending time can be selected, which were then used to determine the temporal boundaries of actions for proposal generation.

Shou *et al.* [62] developed a new 3D convolutional architecture named CDC to predict the actionness scores for each frame in order to determine the temporal boundaries of action instances. CDC consisted of two parts. In the first part 3D convolution layers were used to reduce the spatial and the temporal resolutions simultaneously. In the second part, 2D convolutional layers were applied in the spatial domain to reduce

the resolution while deconvolutional layers were employed in the temporal domain to restore the resolution. As a result, the output of CDC can have the same length as the input in the temporal domain, so that the proposed framework was able to model the spatio-temporal information and predict action label for each frame.

Zhao *et al.* [103] introduced a new effective proposal generation method called temporal actionness grouping to address the temporal action localization problem. The input videos were first divided into snippets and an actionness classifier was used to predict the binary actionness scores for each snippets. Consecutive snippets with high actionness score were grouped to form proposals, which were used to build a feature pyramid to evaluate the action completeness for them. Their method has improvement on the performance on temporal action localization task compared to other state of the art methods.

In the first type of methods, the generated proposals often have relatively accurate temporal boundaries as they are determined based on frame-wise or snippet-wise labels at a finer level with larger aperture of temporal details. However, it is easy for them to wrongly predict low actionness scores for frames within the action instances, which results in low recall rates.

Another set of works [9, 15, 85, 5] applied anchor-based object detection framework, which first generated as set of temporal segment anchors, and then detection heads consisting of boundary regressors and action classifiers were used to refine the locations and predict action classes for the anchors.

Lin *et al.* [36] introduced a single shot detection network to predict the temporal locations of action instances in videos. The idea came from the Single Shot Detection [44] framework used for the object detection task. They first used convolutional neural networks to extract appearance and motion features, which were then used to build a base layer and three anchor layers to generate prediction anchor instances. Offset regression and action class prediction were then performed based on the

prediction anchor instances. Their method can achieve comparable results to those of the state of the art methods, but the temporal boundaries of the detected action segments were less precise.

Gao *et al.* [15] proposed a temporal action localization network named TRUN, which decomposed long videos into short clip units and built anchor pyramid to predict offsets and action class labels for these clip units in order to generate temporal action proposals. By doing so, the time cost of the proposal generation process was significantly reduced and TURN can achieve run time of 880 frames per second on a TITAN X GPU.

Inspired by Faster RCNN [58] object detection framework, Chao *et al.* [5] proposed an improved approach for temporal action localization named TAL-Net. They applied the two-stage Faster RCNN framework on the temporal domain to detect the starting and ending points of action instances. In addition to this, in order to address the reception field alignment issue brought by the temporal localization problem, they used a multi-scale architecture so that extreme variation of action duration can be accommodated. Moreover, they also considered the temporal context of action instances for better proposals generation and action class prediction.

In the second category, as the proposals are generated by the predefined anchors with various time intervals over the untrimmed videos, they are able to cover most ground-truth instances. Unfortunately, they usually has lower performances in terms of temporal boundary accuracy.

Either line of the aforementioned works can take advantage of inherent merits of the methods in the other line, resulting in the trade-off between the boundary accuracy and recall rate. Some recent attempts, such as CTAP [13] and MGG [46], were proposed to take the complementary merits of both lines of works into consideration. They either used several isolated components with the stage-wise training scheme, such as CTAP [13], or simply employed frame-wise or snippet-wise labels to filter the boundaries of anchor-based segment proposals [46] but without fully exploring the complementarity in a more systematic way.

2.4 Spatio-temporal Visual Grounding

Spatio-temporal visual grounding is a newly proposed vision and language task. The goal of spatio-temporal visual grounding is to localize the target objects from the given videos by using the textual queries that describe the target objects.

Zhang *et al.* [102] proposed STGRN to address the spatio-temporal visual grounding task. They first used a pre-trained object detector to detect object proposals from each frame of the input videos. These object proposals were then used to extract region features to build a spatio-temporal graph for relationship modelling. The spatio-temporal graph consisted of the implicit and explicit spatial subgraphs to reason the spatial relationship within each frame and the dynamic temporal subgraph to exploit the temporal relationship across consecutive frames. The proposed graph was used to produce cross-modality spatio-temporal representations, which were used to generate spatio-temporal tubes by a spatial localizer and a temporal localizer.

In [101], Zhang *et al.* proposed a spatial-temporal video grounding framework to model the relationship between the target objects and the related objects while suppressing the relations of unnecessary objects. Specifically, multi-branch architecture was used to handle different types of relations, with each branch focused on its corresponding objects.

Similar to the work in [102], Tang *et al.* [76] also used a pre-trained object detector to generate human proposals. Instead of using these proposals to model the relationship within each frame, they used linking algorithm to link proposals across frames to generate human tubes. Each human tube was then used to extract features, which were then fed into a visual-linguistic transformer to learn cross-modality representations and model temporal information. The cross-modality representation were then used to select target human tubes from a set of human tube proposals. Additionally, temporal trimming process was used to refine the temporal boundaries of the selected human tubes.

One of the drawbacks for the aforementioned spatio-temporal visual grounding methods is that they rely on pre-trained object detectors

to generate a set of object proposals, which require additional training data. Also, when the categories of the target objects do not appeared in the training data of the pre-trained object detectors, it is hard for them to generate proposals with these categories, which degrades the spatio-temporal visual grounding performance.

2.5 Summary

In this chapter, we have summarized the related works on the background vision tasks and three vision problems that are studied in this thesis. Specifically, we first briefly introduce the existing methods on the image classification problem. Then, we review the one-stage, two-stage and anchor-free convlutional neural network based object detection framework. We then summarize the two-stream framework and 3D convolutional neural networks for spatio-temporal representation learning. Besides, we also review the existing spatio-temporal action localization methods. Additionally, we also review the related works on different types of temporal action localization methods including frame-based methods and anchor-based methods. The recent works on combining these two types of works to leverage the complementary information are also discussed. Finally, we give a review on the new vision and language task spatio-temporal visual grounding.

In this thesis, we develop new deep learning approaches on spatio-temporal action localization, temporal action localization, and spatio-temporal visual grounding. Although there are many works related to these vision problems in the literature as summarized above, in this thesis, we propose new frameworks to address these tasks by using deep neural networks and correspondingly design effective algorithms.

Specifically, in Chapter 3, we propose a new algorithm called progressive cross-stream cooperation for spatio-temporal action lcoalization, in which we build a two-stream two-stage object detection framework and exchange complementary information between these two streams at the proposal and feature levels. The newly proposed strategy allows

the information from one stream to help better train another stream and progressively refine the spatio-temporal action localization results.

In Chapter 4, based on the observation of the existing anchor-based and frame-based temporal action localization methods, we propose a novel anchor-frame cooperation module, which can exploit the complementary between the anchor-based and frame-based temporal action localization methods. In addition, we also leverage the complementary information between appearance and motion clues to enhance the anchor-based features, which can further benefit the information fusion between two granularities.

In Chapter 5, we build upon the recent proposed visual-linguistic transformer to learn cross-modality representations for the spatio-temporal visual grounding task. Additionally, by introducing a simple but effective spatio-temporal combination and decomposition module, our proposed framework can remove the dependence on the pre-trained object detectors. We use the anchor-free object detection framework to directly generate object tubes from frames in the input videos.

Chapter 3

Progressive Cross-stream Cooperation in Spatial and Temporal Domain for Action Localization

Spatio-temporal action localization consists of three levels of tasks: spatial localization, action classification, and temporal localization. In this chapter, we propose a new Progressive Cross-stream Cooperation (PCSC) framework that improves all three tasks above. The basic idea is to utilize both spatial region (*resp.*, temporal segment proposals) and features from one stream (*i.e.*, the Flow/RGB stream) to help another stream (*i.e.*, the RGB/Flow stream) to iteratively generate better bounding boxes in the spatial domain (*resp.*, temporal segments in the temporal domain). In this way, not only the actions could be more accurately localized both spatially and temporally, but also the action classes could be predicted more precisely. Specifically, we first combine the latest region proposals (for spatial detection) or segment proposals (for temporal localization) from both streams to form a larger set of labelled training samples to help learn better action detection or segment detection models. Second, to learn better representations, we also propose a new message passing approach to pass information from one stream to another stream, which also leads to better action detection and segment detection models. By first using our newly proposed PCSC framework for spatial localization at the frame-level and then applying our temporal

PCSC framework for temporal localization at the tube-level, the action localization results are progressively improved at both the frame level and the video level. Comprehensive experiments on two benchmark datasets UCF-101-24 and J-HMDB demonstrate the effectiveness of our newly proposed approaches for spatio-temporal action localization in realistic scenarios.

3.1 Background

3.1.1 Spatial temporal localization methods

Spatio-temporal action localization involves three types of tasks: spatial localization, action classification, and temporal localization. A huge amount of efforts have been dedicated to improve the three tasks from different perspectives. First, for spatial localization, the state-of-the-art human detection methods are utilized to obtain precise object proposals, which includes the use of fast and faster R-CNN in [8, 60] as well as Single Shot Multibox Detector (SSD) in [67, 26].

Second, discriminant features are also employed for both spatial localization and action classification. For example, some methods [26, 8] stack neighbouring frames near one key frame to extract more discriminant features in order to remove the ambiguity of actions in each single frame and to better represent this key frame. Other methods [66, 55, 8] utilize recurrent neural networks to link individual frames or use 3D CNNs [4, 77] to exploit temporal information. Several works [31, 93] detect actions in a single frame and predict action locations in the neighbouring frames to exploit the spatio-temporal consistency. Additionally, another work [105] uses a Markov chain model to integrate the appearance, motion and pose clues for action classification and spatial-temporal action localization.

Third, temporal localization is to form action tubes from per-frame

detection results. The methods for this task are largely based on the association of per-frame detection results, such as the overlap, continuity and smoothness of objects, as well as the action class scores. To improve localization accuracies, a variety of temporal refinement methods have been proposed, e.g., the traditional temporal sliding windows [54], dynamic programming [67, 60], tubelets linking [26], and thresholding-based refinement [8], etc.

Finally, several methods were also proposed to improve action detection efficiency. For example, without requiring time-consuming multi-stage training process, the works in [60, 54] proposed to train a single CNN model by simultaneously performing action classification and bounding box regression. More recently, an online real-time spatio-temporal localization method is also proposed in [67].

3.1.2 Two-stream R-CNN

Based on the observation that appearance and motion clues are often complementary to each other, several state-of-the-art action detection methods [60, 26, 54, 8] followed the standard two-stream R-CNN approach. The features extracted from the two streams are fused to improve action detection performance. For example, in [60], the softmax score from each motion bounding box is used to help the appearance bounding boxes with largest overlap. In [8], three types of fusion strategies are discussed: i) simply averaging the softmax outputs of the two streams, ii) learning per-class weights to weigh the original pre-softmax outputs and applying softmax on the weighted sum, and iii) training a fully connected layer on top of the concatenated output from each stream. It is reported in [8] that the third fusion strategy achieves the best performance. In [94], the final detection results are generated by integrating the results from an early-fusion method, which concatenates the appearance and motion clues as the input to generate detection results, and a late-fusion method, which combines the outputs from the two streams.

Please note that most appearance and motion fusion approaches in

the existing works as discussed above are based on the late fusion strategy. They are only trained (if there is any training process) on top of the detection networks of the two streams. In contrast, in this work we iteratively use both types of features and bounding boxes from one stream to progressively help learn better action detection models for another stream, which is intrinsically different with these existing approaches [67, 26] that fuse two-stream information only at the feature level in a late fusion fashion.

3.2 Action Detection Model in Spatial Domain

Building upon the two-stream framework [54], we propose a Progressive Cross-stream Cooperation (PCSC) method for action detection at the frame level. In this method, the RGB (appearance) stream and the flow (motion) stream iteratively help each other at both feature level and region proposal level in order to achieve better localization results.

3.2.1 PCSC Overview

The overview of our PCSC method (*i.e.*, the frame-level detection method) is provided in Fig. 3.1. As shown in Fig. 3.1 (a), our PCSC is composed of a set of "stages". Each stage refers to one round of cross-stream cooperation process, in which the features and region proposals from one stream will help improve action localization performance for another stream. Specifically, each stage comprises of two cooperation modules and a detection head module. Our **detection head** module is a standard one, which consists of several layers for region classification and regression.

The two cooperation modules include region-proposal-level cooperation and feature-level cooperation, which are introduced in details in Section 3.2.2 and Section 3.2.3, respectively. For region-proposal-level cooperation, the detection results from one stream (e.g, the RGB stream) are used as the additional region proposals, which are combined with the region proposals from the other stream (e.g, the flow stream) to refine the region proposals and improve the action localization results. Based on the refined region proposals, we also perform feature-level cooperation

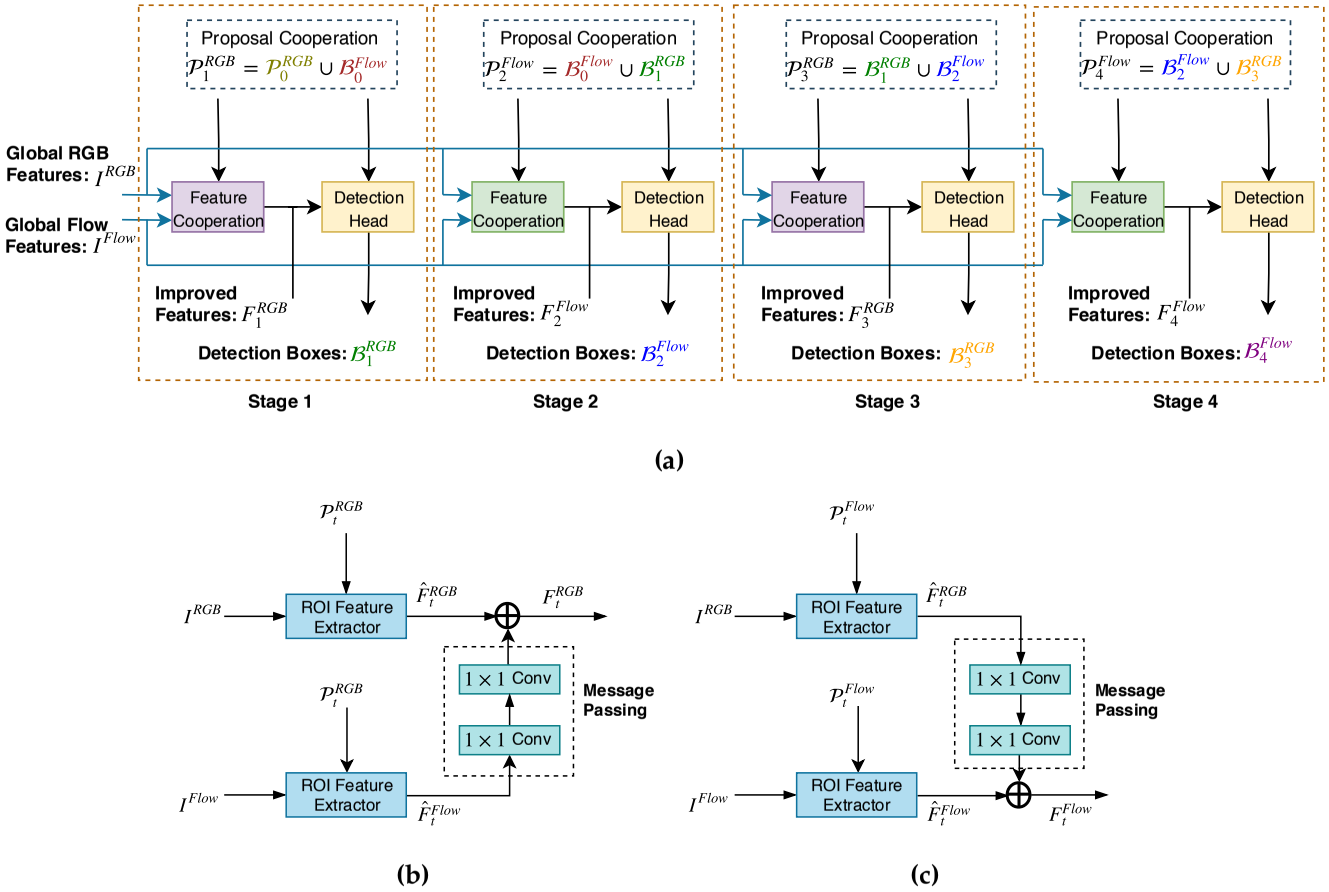


Figure 3.1: (a) Overview of our Cross-stream Cooperation framework (four stages are used as an example for better illustration). The region proposals and features from the flow stream help improve action detection results for the RGB stream at Stage 1 and Stage 3, while the region proposals and features from the RGB stream help improve action detection results for the flow stream at Stage 2 and Stage 4. Each stage comprises of two cooperation modules and the detection head. (b) (c) The details of our feature-level cooperation modules through message passing from one stream to another stream. The flow features are used to help the RGB features in (b), while the RGB features are used to help the flow features in (c).

by first extracting RGB/flow features from these ROIs and then refining these RGB/flow features via a message-passing module shown in Fig. 3.1 (b), and Fig. 3.1 (c), which will be introduced in Section 3.2.3. The refined ROI features in turn lead to better region classification and regression results in the detection head module, which benefits the subsequent action localization process in the next stage. By performing the aforementioned processes for multiple rounds, we can progressively improve the action detection results. The whole network is trained in an end-to-end fashion by minimizing the overall loss, which is the summation of the losses from all stages.

After performing frame-level action detection, our approach links the per-frame detection results to form action tubes, in which the temporal boundary is further refined by using our proposed temporal boundary refinement module. The details are provides in Section 3.3.

3.2.2 Cross-stream Region Proposal Cooperation

We employ the two-stream Faster R-CNN method [58] for frame-level action localization. Each stream has its own Region Proposal Network (RPN) [58] to generate the candidate action regions, and these candidates are then used as the training samples to train a bounding box regression network for action localization. Based on our observation, the region proposals generated by each stream can only partially cover the true action regions, which degrades the detection performance. Therefore, in our method, we use the region proposals from one stream to help another stream.

In this paper, the bounding boxes from RPN are called **region proposals**, while the bounding boxes from the detection head are called **detection boxes**.

The region proposals from the RPNs of the two streams are first used to train their own detection head separately in order to obtain their own corresponding detection boxes. The set of region proposals for each stream is then refined by combining two subsets. The first subset is from the detection boxes of its own stream (e.g, RGB) at the previous stage.

The second subset is from the detection boxes of another stream (e.g, flow) at the current stage. To remove more redundant boxes, a lower NMS threshold is used when the detection boxes in another stream (e.g. flow) are used for the current stream (e.g. RGB).

Mathematically, let $\mathcal{P}_t^{(i)}$ and $\mathcal{B}_t^{(i)}$ denote the set of region proposals and the set of the detection boxes, respectively, where i indicates the ith stream, t denotes the tth stage. The region proposal $\mathcal{P}_t^{(i)}$ is updated as $\mathcal{P}_t^{(i)} = \mathcal{B}_{t-2}^{(i)} \cup \mathcal{B}_{t-1}^{(j)}$, and the detection box $\mathcal{B}_t^{(j)}$ is updated as $\mathcal{B}_t^{(j)} = \mathcal{G}(\mathcal{P}_t^{(j)})$, where $\mathcal{G}(\cdot)$ denotes the mapping function from the detection head module. Initially, when $t = 0$, $\mathcal{P}_0^{(i)}$ is the region proposal from RPN for the ith stream. This process is repeated between the two streams for several stages, which will progressively improve the detection performance of both streams.

With our approach, the diversity of region proposals from one stream will be enhanced by using the complementary boxes from another stream. This could help reduce the missing bounding boxes. Moreover, only the bounding boxes with high confidence are utilized in our approach, which increases the chances to add more precisely detected bounding boxes and thus further help improve the detection performance. These aforementioned strategies, together with the cross-modality feature cooperation strategy that will be described in Section 3.2.3, effectively improve the frame-level action detection results.

Our cross-stream cooperation strategy at the region proposal level shares similar high-level ideas with the two-view learning method co-training [2], as both methods make use of predictions from one stream/view to generate more training samples (*i.e.*, the additional region proposals in our approach) to improve the prediction results for another stream/view. However, our approach is intrinsically different with co-training in the following two aspects. As a semi-supervised learning method, the co-training approach [2] selects unlabelled testing samples and assigns pseudo-labels to those selected samples to enlarge the training set. In contrast, the additional region proposals in our work still come from the training videos instead of testing videos, so we know the labels of the

new training samples by simply comparing these additional region proposals with the ground-truth bounding boxes. Also, in co-training [2], the learnt classifiers will be directly used for predicting the labels of testing data in the testing stage so the complementary information from the two views for the testing data is not exploited. In contrast, in our work, the same pipeline used in the training stage will also be adopted for the testing samples in the testing stage, and thus we can further exploit the complementary information from the two streams for the testing data.

3.2.3 Cross-stream Feature Cooperation

To extract spatio-temporal features for action detection, similar to [19], we use the I3D network as the backbone network for each stream. Moreover, we follow Feature Pyramid Network (FPN) [38] to build feature pyramid with high-level semantics, which has been demonstrated to be useful to improve bounding box proposals and object detection. This involves a bottom-up pathway and a top-down pathway and lateral connections. The bottom-up pathway uses the feed-forward computation along the backbone I3D network, which produces a feature hierarchy with increasing semantic levels but decreasing spatial resolutions. Then these features are upsampled by using the top-down pathway, which are merged with the corresponding features in the bottom-up pathway through lateral connections.

Following [38], we use the feature maps at the layers of Conv2c, Mixed3d, Mixed4f, Mixed5c in I3D to construct the feature hierarchy in the bottom-up pathway, and denote these feature maps as $\{C_2^i, C_3^i, C_4^i, C_5^i\}$, where $i \in \{\text{RGB}, \text{Flow}\}$ indicates the RGB and the flow streams, respectively. Accordingly, the corresponding feature maps in the top-down pathway are denoted as $\{U_2^i, U_3^i, U_4^i, U_5^i\}$.

Most two-stream action detection frameworks [67, 54, 26] only exploit the complementary RGB and flow information by fusing softmax scores or concatenating the features before the final classifiers, which are

insufficient for the features from the two streams to exchange information from one stream to another and benefit from such information exchange. Based on this observation, we develop a message-passing module to bridge these two streams, so that they help each other for feature refinement.

We pass the messages between the feature maps in the bottom-up pathway of the two streams. Denote l as the index for the set of feature maps in $\{C_2^i, C_3^i, C_4^i, C_5^i\}$, $l \in \{2,3,4,5\}$. Let us use the improvement of the RGB features as an example (the same method is also applicable to the improvement of the flow features). Our message-passing module improves the features C_l^{RGB} with the help of the features C_l^{Flow} as follows:

$$C_l^{\text{RGB}} = f_\theta(C_l^{\text{Flow}}) \oplus C_l^{\text{RGB}}, \tag{3.1}$$

where $\oplus$ denotes the element-wise addition of the feature maps, $f_\theta(\cdot)$ is the mapping function (parameterized by θ) of our message-passing module. The function $f_\theta(C_l^{\text{Flow}})$ nonlinearly extracts the message from the feature C_l^{Flow}, and then uses the extracted message for improving the features C_l^{RGB}.

The output of $f_\theta(\cdot)$ has to produce the feature maps with the same number of channels and resolution as C_l^{RGB} and C_l^{Flow}. To this end, we design our message-passing module by stacking two 1×1 convolutional layers with relu as the activation function. The first 1×1 convolutional layer reduces the channel dimension and the second convolutional layer restores the channel dimension back to its original number. This design saves the number of parameters to be learnt in the module and exchanges message by using two-layer non-linear transform. Once the feature maps C_l^{RGB} in the bottom-up pathway are refined, the corresponding features maps U_l^{RGB} in the top-down pathway are generated accordingly.

The above process is for image-level messaging passing only. The image-level message passing is only performed from the Flow stream to the RGB stream once. This message passing provides good features to initialize the message-passing stages.

Denote the image-level feature map sets for the RGB and flow streams by I^{RGB} and I^{Flow} respectively. They are used to extract features $\hat{F}_t^{\text{RGB}}$ and $\hat{F}_t^{\text{Flow}}$ by ROI pooling at each stage t, as shown in Fig. 3.1. At Stage 1 and Stage 3, the ROI-feature $\hat{F}_t^{\text{Flow}}$ of the flow stream is used to help improve the ROI-feature $\hat{F}_t^{\text{RGB}}$ of the RGB stream, as illustrated in Fig. 3.1 (b). Specifically, the improved RGB feature F_t^{RGB} is obtained by applying the same method in Eqn. (3.1). Similarly, at Stage 2 and Stage 4, the ROI-feature $\hat{F}_t^{\text{RGB}}$ of the RGB stream is also used to help improve the ROI-feature $\hat{F}_t^{\text{Flow}}$ of the flow stream, as illustrated in Fig. 3.1 (c). The message passing process between ROI-features aims to provide better features for action box detection and regression, which benefits the next cross-stream cooperation stage.

3.2.4 Training Details

For better spatial localization at the frame-level, we follow [26] to stack neighbouring frames to exploit temporal context and improve action detection performance for key frames. A key frame is a frame containing the ground-truth actions. Each training sample, which is used to train the RPN in our PCSC method, is composed of k neighbouring frames with the key frame in the middle. The region proposals generated from the RPN are assigned with positive labels when they have an intersection-over-union (IoU) overlap higher than 0.5 with any ground-truth bounding box, or negative labels if their IoU overlap is lower than 0.5 with all ground-truth boxes. This label assignment strategy also applies to the additional bounding boxes from the assistant stream during the region proposal-level cooperation process.

3.3 Temporal Boundary Refinement for Action Tubes

After the frame-level detection results are generated, we then build the action tubes by linking them. Here we use the same linking strategy as in [67], except that we do not apply temporal labeling. Although this

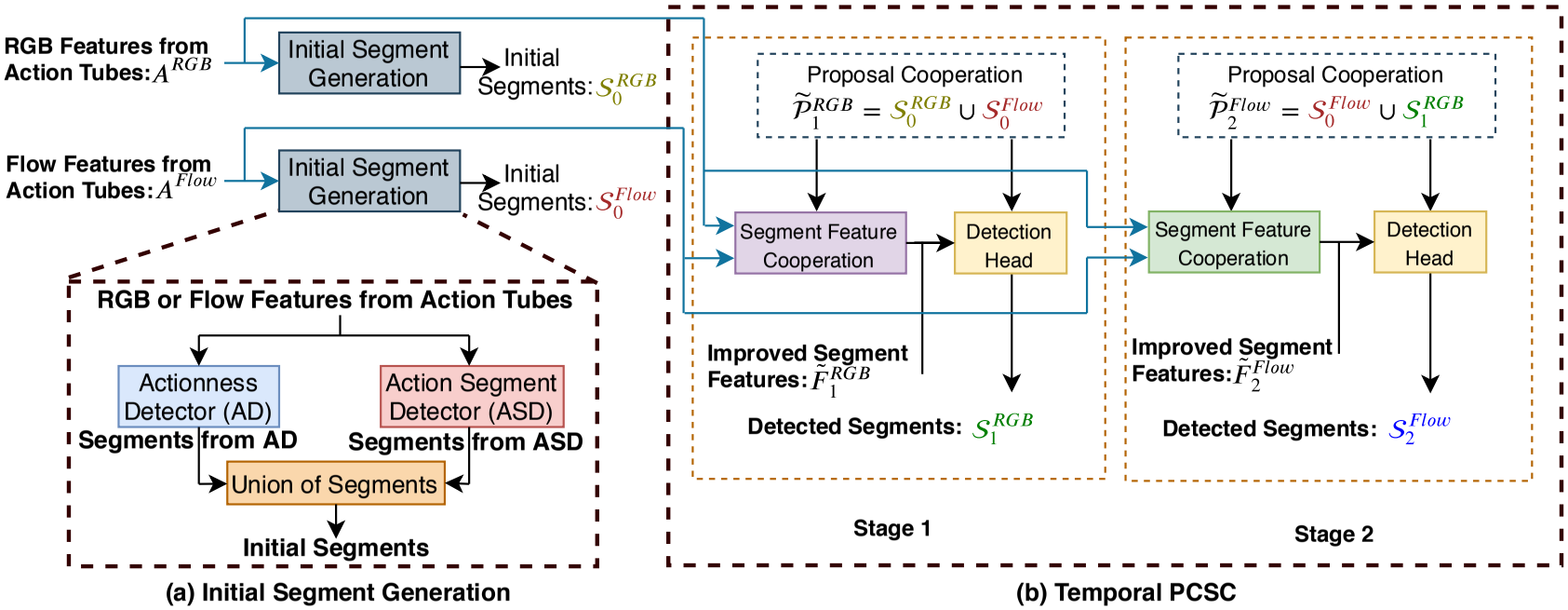


Figure 3.2: Overview of two modules in our temporal boundary refinement framework: (a) the Initial Segment Generation Module and (b) the Temporal Progressive Cross-stream Cooperation (T-PCSC) Module. The Initial Segment Generation module combines the outputs from Actionness Detector and Action Segment Detector to generate the initial segments for both RGB and flow streams. Our T-PCSC framework takes these initial segments as the segment proposals and iteratively improves the temporal action detection results by leveraging complementary information of two streams at both feature and segment proposal levels (two stages are used as an example for better illustration). The segment proposals and features from the flow stream help improve action segment detection results for the RGB stream at Stage 1, while the segment proposals and features from the RGB stream help improve action detection results for the flow stream at Stage 2. Each stage comprises of the cooperation module and the detection head. The segment proposal level cooperation module refines the segment proposals $\widetilde{\mathcal{P}}_t^i$ by combining the most recent segment proposals from the two streams, while the segment feature level cooperation module improves the features $\tilde{\mathbf{F}}_t^i$, where the superscript $i \in \{RGB, Flow\}$ denotes the RGB/Flow stream and the subscript t denotes the stage number. The detection head is used to estimate the action segment location and the actionness probability.

linking strategy is robust to missing detection, it is still difficult to accurately determine the start and the end of each action tube, which is a key factor that degrades video-level performance of our action detection framework.

To address this problem, the temporal boundaries of the action tubes need to be refined. Our temporal boundary refinement method uses the features extracted from action tubes to refine the temporal boundaries of the action tubes. It is built upon two-stream framework and consists of four modules: Pre-processing Module, Initial Segment Generation module, Temporal Progressive Cross-stream Cooperation (T-PCSC) module, and Post-processing Module. An overview of the two major modules, i.e., the Initial Segment Generation Module and the T-PCSC Module, is provided in Fig. 3.2. Specifically, in the Pre-processing module, we use each bounding box in the action tubes to extract the 7×7 feature maps for the detected region on each frame by ROI-Pooling and then temporally stack these feature maps to construct the features for the action tubes. These features are then used as the input to the Initial Segment Generation module. As shown in Fig. 3.2, we develop two segment generation methods in the Initial Segment Generation module: (1) a class-specific actionness detector to evaluate actionness (*i.e.* the happening of an action) for each frame, and (2) an action segment detector based on two-stream Faster-RCNN framework to detect action segments. The outputs of the actionness detector and the action segment detector are combined to generate the initial segments for both the RGB and flow streams. Our T-PCSC Module extends our PCSC framework from the spatial domain to the temporal domain for temporal localization, and it uses features from the action tubes and the initial segments generated by the Initial Segment Generation module to encourage the two streams to help each other at both feature level and segment proposal level to improve action segment detection. After multiple stages of cooperation, the output action segments from each stage are combined in the Post-processing module, in which NMS is also applied to remove redundancy and generate the final action segments.

3.3.1 Actionness Detector (AD)

We first develop a class-specific actionness detector to detect the actionness at each given location (spatially and temporally). Taking advantage of the predicted action class labels produced by frame-level detection, we construct our actionness detector by using N binary classifiers, where N is the number of action classes. Each classifier addresses the actionness of a specific class (*i.e.*, whether an action happens or not at a given location in a frame). This strategy is more robust than learning a general actionness classifier for all action classes. Specifically, after frame-level detection, each bounding box has a class label, based on which, the bounding boxes from the same class are traced and linked to form action tubes [67]. To train the binary actionness classifier for each action class i, the bounding boxes that are within the action tubes and predicted as class i by the frame-level detector are used as the training samples. Each bounding box is labeled either as 1 when its overlap with the ground-truth box is greater than 0.5, or as 0 otherwise. Note that the training samples may contain bounding boxes falsely detected near the temporal boundary of the action tubes. Therefore, these samples are the "confusing" samples and are useful for the actionness classifier to learn the subtle but critical features, which better determine the beginning and the end of this action. The output of the actionness classifier is a probability of actionness belonging to class i.

At the testing stage, given an action tube formed by using [67], we apply the class-specific actionness detector to every frame-level bounding box in this action tube to predict its actionness probability (called actionness score). Then a median filter over multiple frames is employed to smooth the actionness scores of all bounding boxes in this tube. If a bounding box has a smoothed score lower than a predefined threshold, it will be filtered out from this action tube, and then we can refine the action tubes so that they have more accurate temporal boundaries. Note that when a non-action region near a temporal boundary is falsely detected, it is included in the training set to train our class-specific actionness detector. Therefore, our approach takes advantage of the confusing samples across temporal boundary to obtain better action tubes at the

testing stage.

3.3.2 Action Segment Detector (ASD)

Motivated by the two-stream temporal Faster-RCNN framework [5], we also build an action segment detector, which directly utilizes segment proposals to detect action segments. Our action segment detector takes both RGB and optical flow features from the action tubes as the input, and follows the fast-RCNN framework [5] to include a Segment Proposal Network (SPN) for temporal segment proposal generation and detection heads for segment proposal regression. Specifically, we extract the segment features by using these two-stream features from each action tube and the segment proposals produced by SPN. The segment features are then sent to the corresponding detection head for regression to generate the initial detection of segments. This process is similar to the action detection method described in Section 3.2. A comparison between the single-stream action detection method and the single-stream action segment detector is shown in Figure 3.3, in which the modules within each blue dashed box share similar functions at the spatial domain and the temporal domain, respectively. Our implementation details for SPN, feature extraction and detection head are introduced below.

(a) SPN In Faster-RCNN, the anchors of different object sizes are pre-defined. Similary, in our SPN, K types of action anchors with different duration (time lengths) are pre-defined. The boundaries of the action anchors are regressed by two layers of temporal convolution (called ConvNet in [5]) using RGB or flow features from the action tube [5]. Note that the object sizes in an image only vary in a relatively small range, but the temporal span of an action can dramatically change in a range from less than one second to more than one minute. Consequently, it is inappropriate for anchors with different scales to share the temporal features from the same receptive field. To address this issue, we use the multi-branch architecture and dilated temporal convolutions in [5] for each anchor prediction based on the anchor-specific features. Specifically, for each scale of anchors, a sub-network with the same architecture

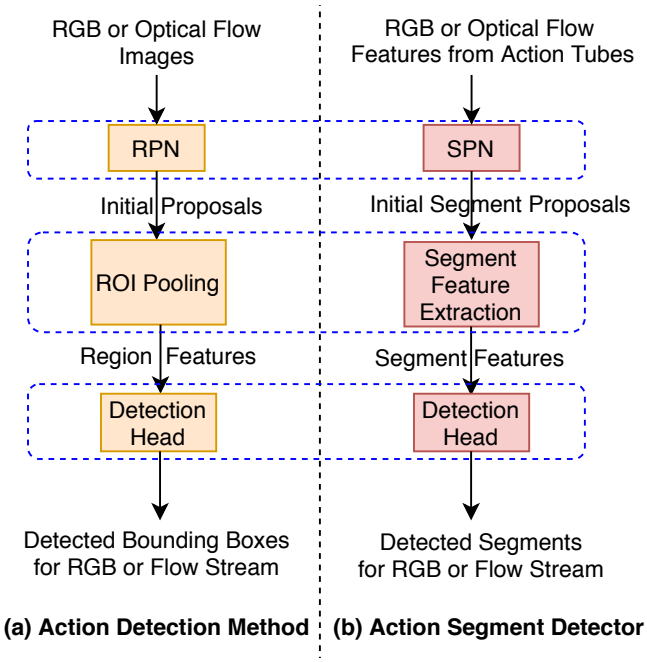


Figure 3.3: Comparison of the single-stream frameworks of the action detection method (see Section 3) and our action segment detector (see Section 4.2). The modules within each blue dashed box share similar functions at the spatial domain and the temporal domain, respectively.

but different dilation rate is employed to generate features with different receptive fields and decide the temporal boundary. In this way, our SPN consists of K sub-networks targeting at different time lengths. In order to consider the context information, for each anchor with the scale s, we require the receptive field to cover the context information with temporal span $s/2$ before and after the start and the end of the anchor. We then regress temporal boundaries and predict actionness probabilities for the anchors by applying two parallel temporal convolutional layers with the kernel size of 1. The output segment proposals are ranked according to their proposal actionness probabilities and NMS is applied to remove the redundant proposals. Please refer to [5] for more details.

(b) Segment feature extraction After generating segment proposals, we construct segment features to refine the temporal boundaries and predict the segment actionness probabilities. For each segment proposal with the length l_s, the start point t_{start} and the end point t_{end}, we extract three

types of features, F_{center}, F_{start}, and F_{end}. We use temporal ROI-Pooling on top of the segment proposal regions to pool action tube features and obtain the features $F_{center} \in R^{D \times l_t}$, where l_t is the temporal length and D is the dimension of the features at each temporal location. As the context information immediately before and after the segment proposal is helpful to decide the temporal boundaries of an action, this information should be represented by our segment features. Therefore, we additionally use temporal ROI-Pooling to pool features F_{start} (*resp.*, F_{end}), from the segments centered at t_{start} (*resp.*, t_{end}) with the length of $2l_s/5$ to represent the starting (reps., ending) information. We then concatenate F_{start}, F_{center} and F_{end} to construct the segment features F_{seg}.

(c) Detection head The extracted segment features are passed to the detection head to obtain the refined segment boundaries and the segment actionness probabilities. As the temporal length of the extracted temporal features is not fixed, similar to SPN, the detection head module consists of M sub-networks corresponding to the M groups of the temporal length of the segment proposals resulting from SPN. For each sub-network, a spatial fully-connected layer is first used to reduce the spatial resolution, and then two temporal convolution layers with the kernel size of 3 are applied to non-linearly combine temporal information. The output features are used to predict the actionness probabilities and regress the segment boundaries. For actionness classification, in addition to minimize the segment-level actionness loss, we also minimize the frame-level actionness loss. We observe that it is beneficial to extract discriminative features by predicting actionness for each frame, which will eventually help refine the boundaries of segments.

3.3.3 Temporal PCSC (T-PCSC)

Our PCSC framework can leverage the complementary information at the feature level and the region proposal level to iteratively improve action detection performance at the spatial domain. Similarly, we propose a Temporal PCSC (T-PCSC) module that extends our PCSC framework

from the spatial domain to the temporal domain for temporal localization. Specifically, we combine the outputs generated by the actionness detector and the action segment detector to obtain the initial segments for each stream. These initial segments are then used by our T-PCSC module as the initial segment proposals to generate action segments.

Similar to our PCSC method described in Section 3.2, our T-PCSC method consists of the feature cooperation module and proposal cooperation module, as shown in Fig. 3.2. Note that for the message-passing block in the feature cooperation module, the two 1×1 convolution layers are replaced by two temporal convolution layers with the kernel size of 1. We only apply T-PCSC for two stages in our action segment detector method due to memory constraints and the observation that we already achieve good results after using two stages. Similar to our PCSC framework shown in Figure 3.1, in the first stage of our T-PCSC framework, the initial segments from the RGB stream are combined with those from the flow stream to form a new segment proposal set by using the proposal cooperation module. This new segment proposal set is used to construct the segment features as described in Section 3.3.2 (b) for both the RGB and the flow streams, and then the feature cooperation module passes the messages from the flow features to the RGB features to refine the RGB features. The refined RGB features are then used to generate the output segments by the detection head described in Section 3.3.2 (c). A similar process takes place in the second stage. But this time, the output segments from the flow stream and the first stage are used to form the segment proposal set and the message passing process is conducted from the RGB features to the flow features to help refine the flow features. We then generate the final action segments by combining the output action segments of each stage and applying NMS to remove redundancy and obtain the final results.

3.4 Experimental results

We introduce our experimental setup and datasets in Section 3.4.1, and then compare our method with the state-of-the-art methods in Section 3.4.2, and conduct ablation study in Section 3.4.3.

3.4.1 Experimental Setup

Datasets. We evaluate our PCSC method on two benchmarks: UCF-101-24 [68] and J-HMDB-21 [24]. **UCF-101-24** contains 3207 untrimmed videos from 24 sports classes, which is a subset of the UCF-101 dataset, with spatio-temporal annotations provided by [67]. Following the common practice, we use the predefined "split 1" protocol to split the training and test sets, and report the results based on this split. **J-HMDB-21** contains 928 videos from 21 action classes. All the videos are trimmed to contain the actions only. We perform the experiments on three predefined training-test splits, and report the average results on this dataset.

Metrics. We evaluate the action detection performance at both frame-level and video-level by mean Average precision (mAP). To calculate mAP, a detection box is considered as a correct one when its overlap with a ground-truth box or tube is greater than a threshold δ. The overlap between our detection results and the ground truth is measured by the intersection-over-union (IoU) score at the frame-level and the spatio-temporal tube overlap at the video-level, respectively. In addition, we also report the results based on the COCO evaluation metric [40], which averages the mAPs over 10 different IoU thresholds from 0.5 to 0.95 with the interval of 0.05.

To evaluate the localization accuracy of our method, we also calculate Mean Average Best Overlap (MABO). We compute the IoU scores between the bounding boxes from our method and the ground-truth boxes, and then average the IoU scores from the bounding boxes with largest overlap to the ground-truth boxes for each class. The MABO is calculated by averaging the averaged IoU scores over all classes.

To evaluate the classification performance of our method, we report the average confident score over different IoU threshold θ. The average confident score is computed by averaging the confident scores of the bounding boxes from our method that have an IoU with the ground-truth boxes greater than θ.

Implementation Details. We use the I3D features [4] for both streams, and the I3D model is pretrained with Kinetics. The optical flow images are extracted by using FlowNet v2 [23]. The mini-batch size used to train the RPN and the detection heads in our action detection method is 256 and 512, respectively. Our PCSC based action detection method is trained for 6 epochs by using three 1080Ti GPUs. The initial learning rate is set as 0.01, which drops 10% at the 5th epoch and another 10% at the 6th epoch. For our actionness detector, we set the window size as 17 when using median filter to smooth the actionness scores. we also use a predefined threshold to decide the starting and ending locations for action tubes, which is empirically set as 0.01. For segment generation, we apply the anchors with the following K scales ($K = 11$ in this work): {6, 12, 24, 42, 48, 60, 84, 92, 144, 180, 228}. The length ranges R, which are used to group the segment proposals, are set as follows: $\{[20, 80], [80, 160], [160, 320], [320, \infty]\}$, *i.e.*, we have $M = 4$, and the corresponding feature lengths l_t for the length ranges are set to $\{15, 30, 60, 120\}$. The numbers of the scales and the lengths are empirically decided based on the frame numbers. In this work, we apply NMS to remove redundancy and combine the detection results from different stages with an overlap threshold of 0.5. The mini-batch sizes used to train the SPN and the detection heads in our temporal refinement method are set as 128 and 32, respectively. Our PCSC based temporal refinement method is trained for 5 epochs and the initial learning rate is set as 0.001, which drops 10% at the 4th epoch and another 10% at the 5th epoch.

3.4.2 Comparison with the State-of-the-art methods

We compare our PCSC method with the state-of-the-art methods. The results of the existing methods are quoted directly from their original papers. In addition, we also evaluate the object detection method (denoted

as “Faster R-CNN + FPN") in [38] with the I3D network as its backbone network, in which we report the results by only using the single stream (*i.e.*, RGB or Flow) features as well as the result by applying NMS to the union set of the two stream bounding boxes. Specifically, the only difference between the method in [38] and our PCSC is that our PCSC method has the cross-stream cooperation modules while the work in [38] does not have them. Therefore, the comparison between the two approaches can better demonstrate the benefit of our cross-stream cooperation strategy.

Results on the UCF-101-24 Dataset

The results on the UCF-101-24 dataset are reported in Table 3.1 and Table 3.2.

Table 3.1: Comparison (mAPs % at the frame level) of different methods on the UCF-101-24 dataset when using the IoU threshold δ at 0.5.

	Streams	mAPs
Weinzaepfel, Harchaoui, and Schmid [83]	RGB+Flow	35.8
Peng and Schmid [54]	RGB+Flow	65.7
Ye, Yang, and Tian [94]	RGB+Flow	67.0
Kalogeiton et al. [26]	RGB+Flow	69.5
Gu et al. [19]	RGB+Flow	76.3
Faster R-CNN + FPN (RGB) [38]	RGB	73.2
Faster R-CNN + FPN (Flow) [38]	Flow	64.7
Faster R-CNN + FPN [38]	RGB+Flow	75.5
PCSC (Ours)	RGB+Flow	**79.2**

Table 3.1 shows the mAPs from different methods at the frame-level on the UCF-101-24 dataset. All mAPs are calculated based on the IoU threshold $\delta = 0.5$. From Table 3.1, we observe that our PCSC method achieves an mAP of 79.2%, outperforming all the existing methods by a large margin. Especially, our PCSC performs better than Faster R-CNN + FPN [38] by an improvement of 3.7%. This improvement is fully due to the proposed cross-stream cooperation framework, which is the only difference between [38] and our PCSC. It is interesting to observe that both methods [26] and [19] additionally utilize the temporal context for

Table 3.2: Comparison (mAPs % at the video level) of different methods on the UCF-101-24 dataset when using different IoU thresholds. Here "AD" is for actionness detector, and "ASD" is for action segment detector.

IoU threshold δ	Streams	0.2	0.5	0.75	0.5:0.95
Weinzaepfel et al. [83]	RGB+Flow	46.8	-	-	-
Zolfaghari et al. [105]	RGB+Flow +Pose	47.6	-	-	-
Peng and Schmid [54]	RGB+Flow	73.5	32.1	02.7	07.3
Saha et al. [60]	RGB+Flow	66.6	36.4	0.8	14.4
Yang, Gao, and Nevatia [93]	RGB+Flow	73.5	37.8	-	-
Singh et al. [67]	RGB+Flow	73.5	46.3	15.0	20.4
Ye, Yang, and Tian [94]	RGB+Flow	76.2	-	-	-
Kalogeiton et al. [26]	RGB+Flow	76.5	49.2	19.7	23.4
Li et al. [31]	RGB+Flow	77.9	-	-	-
Gu et al. [19]	RGB+Flow	-	59.9	-	-
Faster R-CNN + FPN (RGB) [38]	RGB	77.5	51.3	14.2	21.9
Faster R-CNN + FPN (Flow) [38]	Flow	72.7	44.2	7.4	17.2
Faster R-CNN + FPN [38]	RGB+Flow	80.1	53.2	15.9	23.7
PCSC + AD [69]	RGB+Flow	84.3	61.0	23.0	27.8
PCSC + ASD+ AD + T-PCSC (Ours)	RGB+Flow	**84.7**	**63.1**	**23.6**	**28.9**

per frame-level action detection. However, both methods do not fully exploit the complementary information between the appearance and motion clues, and therefore they are worse than our method. Moreover, our method also outperforms [26] and [19] by 9.7% and 2.9%, respectively, in terms of frame-level mAPs.

Table 3.2 reports the video-level mAPs at various IoU thresholds (0.2, 0.5, and 0.75) on the UCF-101-24 dataset. The results based on the COCO evaluation metrics [40] are reported in the last column of Table 3.2. Our method is denoted as "PCSC + ASD + AD + T-PCSC" in Table 3.2, where the proposed temporal boundary refinement method, consisting of three modules (*i.e.*, the actionness detector module, the action segment detector module, and the Temporal PCSC), is applied to refine the action tubes generated from our PCSC method. Our preliminary

work [69], which only uses the actionness detector module as the temporal boundary refinement method, is denoted as "PCSC + AD". Consistent with the observations at the frame-level, our method outperforms all the state-of-the-art methods under all evaluation metrics. When using the IoU threshold $\delta = 0.5$, we achieve an mAP of 63.1% on the UCF-101-24 dataset. This result beats [26] and [19], which only achieve the mAPs of 49.2% and 59.9%, respectively. Moreover, as the IoU threshold increases, the performance of our method drops less when compared with other state-of-the-art methods. The results demonstrate that our detection method achieves higher localization accuracy than other competitive methods.

Results on the J-HMDB Dataset

For the J-HMDB dataset, the frame-level mAPs and the video-level mAPs from different mothods are reported in Table 3.3 and Table 3.4, respectively. Since the videos in the J-HMDB dataset are trimmed to only contain actions, the temporal refinement process is not required, so we do not apply our temporal boundary refinement module when generating action tubes on the J-HMDB dataset.

Table 3.3: Comparison (mAPs % at the frame level) of different methods on the J-HMDB dataset when using the IoU threshold δ at 0.5.

	Streams	mAPs
Peng and Schmid [54]	RGB+Flow	58.5
Ye, Yang, and Tian [94]	RGB+Flow	63.2
Kalogeiton et al. [26]	RGB+Flow	65.7
Hou, Chen, and Shah [21]	RGB	61.3
Gu et al. [19]	RGB+Flow	73.3
Sun et al. [72]	RGB+Flow	**77.9**
Faster R-CNN + FPN (RGB) [38]	RGB	64.7
Faster R-CNN + FPN (Flow) [38]	Flow	68.4
Faster R-CNN + FPN [38]	RGB+Flow	70.2
PCSC (Ours)	RGB+Flow	74.8

We have similar observations as in the UCF-101-24 dataset. At the video-level, our method is again the best performer under all evaluation metrics on the J-HMDB dataset (see Table 3.4). When using the IoU

Table 3.4: Comparison (mAPs % at the video level) of different methods on the J-HMDB dataset when using different IoU thresholds.

IoU threshold δ	Streams	0.2	0.5	0.75	0.5:0.95
Gkioxari and Malik [18]	RGB+Flow	-	53.3	-	-
Wang et al. [79]	RGB+Flow	-	56.4	-	-
Weinzaepfel et al. [83]	RGB+Flow	63.1	60.7	-	-
Saha et al. [60]	RGB+Flow	72.6	71.5	43.3	40.0
Peng and Schmid [54]	RGB+Flow	74.1	73.1	-	-
Singh et al. [67]	RGB+Flow	73.8	72.0	44.5	41.6
Kalogeiton et al. [26]	RGB+Flow	74.2	73.7	52.1	44.8
Ye, Yang, and Tian [94]	RGB+Flow	75.8	73.8	-	-
Zolfaghari et al. [105]	RGB+Flow +Pose	78.2	73.4	-	-
Hou, Chen, and Shah [21]	RGB	78.4	76.9	-	-
Gu et al. [19]	RGB+Flow	-	78.6	-	-
Sun et al. [72]	RGB+Flow	-	80.1	-	-
Faster R-CNN + FPN (RGB) [38]	RGB	74.5	73.9	54.1	44.2
Faster R-CNN + FPN (Flow) [38]	Flow	77.9	76.1	56.1	46.3
Faster R-CNN + FPN [38]	RGB+Flow	79.1	78.5	57.2	47.6
PCSC (Ours)	RGB+Flow	**82.6**	**82.2**	**63.1**	**52.8**

threshold $\delta = 0.5$, our PCSC method outperforms [19] and [72] by 3.6% and 2.1%, respectively.

At the frame-level (see Table 3.3), our PCSC method performs the second best, which is only worse than a very recent work [72]. However, the work in [72] uses S3D-G as the backbone network, which provides much stronger features when compared with the I3D features used in our method. In addition, please note that, our method outperforms [72] in terms of mAPs at the video level (see Table 3.4), which demonstrates promising performance of our PCSC method. Moreover, as a general framework, our PCSC method could also take advantage of strong features provided by the S3D-G method to further improve the results, which will be explored in our future work.

3.4.3 Ablation Study

In this section, we take the UCF-101-24 dataset as an example to investigate the contributions of different components in our proposed method.

Table 3.5: Ablation study for our PCSC method at different training stages on the UCF-101-24 dataset.

Stage	PCSC w/o feature cooperation	PCSC
0	75.5	**78.1**
1	76.1	**78.6**
2	76.4	**78.8**
3	76.7	**79.1**
4	76.7	**79.2**

Table 3.6: Ablation study for our temporal boundary refinement method on the UCF-101-24 dataset. Here "AD" is for the actionness detector while "ASD" is for the action segment detector.

	Video mAP (%)
Faster R-CNN + FPN [38]	53.2
PCSC	56.4
PCSC + AD	61.0
PCSC + ASD (single branch)	58.4
PCSC + ASD	60.7
PCSC + ASD + AD	61.9
PCSC + ASD + AD + T-PCSC	**63.1**

Table 3.7: Ablation study for our Temporal PCSC (T-PCSC) method at different training stages on the UCF-101-24 dataset.

Stage	Video mAP (%)
0	61.9
1	62.8
2	**63.1**

Progressive cross-stream cooperation. In Table 3.5, we report the results of an alternative approach of our PCSC (called PCSC w/o feature cooperation). In the second column of Table 3.5, we remove the feature-level cooperation module from Fig. 3.1, and only use the region-proposal-level cooperation module. As our PCSC is conducted in a progressive manner, we also report the performance at different stages to

verify the benefit of this progressive strategy. It is worth mentioning that the output at each stage is obtained by combining the detected bounding boxes from both the current stage and all previous stages, and then we apply non-maximum suppression (NMS) to obtain the final bounding boxes. For example, the output at Stage 4 is obtained by applying NMS to the union set of the detected bounding boxes from Stages 0, 1, 2, 3 and 4. At Stage 0, the detection results from both RGB and the Flow streams are simply combined. From Table 3.5, we observe that the detection performance of our PCSC method with or without the feature-level cooperation module is improved as the number of stages increases. However, such improvement seems to become saturated when reaching Stage 4, as indicated by the marginal performance gain from Stage 3 to Stage 4. Meanwhile, when comparing our PCSC with the alternative approach PCSC w/o feature cooperation at every stage, we observe that both region-proposal-level and feature-level cooperation contributes to performance improvement.

Action tubes refinement. In Table 3.6, we investigate the effectiveness of our temporal boundary refinement method by switching on and off the key components in our temporal boundary refinement module, in which video-level mAPs at the IoU threshold $\delta = 0.5$ are reported. In Table 3.6, we investigate five configurations for temporal boundary refinement. Specifically, in 'PCSC + AD", we use the PCSC method for spatial localization together with the actionness detector (AD) for temporal localization. In "PCSC + ASD", we use the PCSC method for spatial localization, while we use the action segment detection method for temporal localization. In "PCSC + ASD (single branch)", we use the "PCSC + ASD" configuration, but replace the multi-branch architecture in the action segment detection method with the single-branch architecture. In "PCSC + ASD + AD", the PCSC method is used for spatial localization together with both AD and ASD for temporal localization. In "PCSC + ASD + AD + T-PCSC", the PCSC method is used for spatial localization and both actionness detector and action segment detector are used to generate the initial segments, and our T-PCSC is finally used for temporal localization. By generating higher quality detection results at each frame, our PCSC method in the spatial domain outperforms the work

Table 3.8: The large overlap ratio (LoR) of the region proposals from the latest RGB stream with respective to the region proposals from the latest flow stream at each stage in our PCSC action detector method on the UCF-101-24 dataset.

Stage	LoR(%)
1	61.1
2	71.9
3	95.2
4	**95.4**

in [38] that does not use the two-stream cooperation strategy in the spatial domain by 3.2%. After applying our actionness detector module to further boost the video-level performance, we arrive at the video-level mAP of 61.0%. Meanwhile, our action segment detector without using T-PCSC has already been able to achieve the video-level mAP of 60.7%, which is comparable to the performance from the actionness detector module in terms of video-level mAPs. On the other hand, the video-level mAP drops 2.3% after replacing the multi-branch architecture in the action segment detector with the single-branch architecture, which demonstrates the effectiveness of the multi-branch architecture. Note that the video-level mAP is further improved when combining the results from actionness detector and the action segment detector without using the temporal PCSC module, which indicates that the frame-level and the segment-level actionness results could be complementary. When we further apply our temporal PCSC framework, the video-level mAP of our proposed method increases 1.2%, and arrives at the best performance of 63.1%, which demonstrates the effectiveness of our PCSC framework for temporal refinement. In total, we achieve about 7.0% improvement after applying our temporal boundary refinement module on top of our PCSC method, which indicates the necessity and benefits of the temporal boundary refinement process in determining the temporal boundaries of action tubes.

Temporal progressive cross-stream cooperation. In Table 3.7, we report the video-level mAPs at the IoU threshold $\delta = 0.5$ by using our temporal PCSC method at different stages to verify the benefit of this progressive strategy in the temporal domain. Stage 0 is our "PCSC +

Table 3.9: The Mean Average Best Overlap (MABO) of the bounding boxes with the ground-truth boxes at each stage in our PCSC action detector method on the UCF-101-24 dataset.

Stage	MABO(%)
1	84.1
2	84.3
3	84.5
4	**84.6**

Table 3.10: The average confident scores (%) of the bounding boxes with respect to the IoU threshold θ at different stages on the UCF-101-24 dataset.

θ	Stage 1	Stage 2	Stage 3	Stage 4
0.5	55.3	55.7	56.3	**57.0**
0.6	62.1	63.2	64.3	**65.5**
0.7	68	68.5	69.4	**69.5**
0.8	73.9	74.5	74.8	**75.1**
0.9	78.4	78.4	79.5	**80.1**

Table 3.11: Detection speed in frame per second (FPS) for our action detection module in the spatial domain on UCF-101-24 when using different number of stages.

Stage	0	1	2	3	4
Detection speed (FPS)	3.1	2.9	2.8	2.6	2.4

Table 3.12: Detection speed in video per second (VPS) for our temporal boundary refinement method for action tubes on UCF-101-24 when using different number of stages.

Stage	0	1	2
Detection speed (VPS)	2.1	1.7	1.5

ASD + AD" method. Stage 1 and Stage 2 refer to our “PCSC + ASD + AD + T-PCSC" method using one-stage and two-stage temproal PCSC, respectively. The video-level performance after using our temporal PCSC method is improved as the number of stages increases, which demonstrates the effectiveness of the progressive strategy in the temporal domain.

3.4.4 Analysis of Our PCSC at Different Stages

In this section, we take the UCF-101-24 dataset as an example to further investigate our PCSC framework for spatial localization at different stages.

In order to investigate the similarity change of proposals generated from two streams over multiple stages, for each proposal from the RGB stream, we compute its maximum IoU (mIoU) with respect to all the proposals from the flow stream. We define the large overlap ratio (LoR) as the percentage of the RGB-stream proposals whose mIoU with all the flow-stream proposals is larger than 0.7 over the total number of RGB proposals. Table 3.8 shows the LoR of the latest RGB stream with respect to the latest flow stream at every stage on the UCF-101-24 dataset. Specifically, at stage 1 or stage 3, LoR indicates the overlap degree of the RGB-stream proposals generated at the current stage (i.e., stage 1 or stage 3, respectively) with respect to the flow-stream proposals generated at the previous stage (i.e., stage 0 or stage 2, respectively); at stage 2 or stage 4, LoR indicates the overlap degree of the RGB-stream proposals generated at the previous stage (i.e., stage 1 or stage 3, respectively) with respect to the flow-stream proposals generated at the current stage (i.e., stage 2 or stage 4, respectively). We observe that the LoR increases from 61.1% at stage 1 to 95.4% at stage 4, which demonstrates that the similarity of the region proposals from these two streams increases over stages. At stage 4, the region proposals from different streams are very similar to each other, and therefore, these two types of proposals are lack of complementary information. As a result, there is no further improvement after stage 4.

To further evaluate the improvement of our PCSC method over stages, we report MABO and the average confident score to measure the localization and classification performance of our PCSC method over stages. Table 3.9 and Table 3.10 show the MABO and the average confident score of our PCSC method at every stages on the UCF-101-24 dataset. In terms of both metrics, improvement after each stage can be observed clearly, which demonstrates the effectiveness of our PCSC method for both localization and classification.

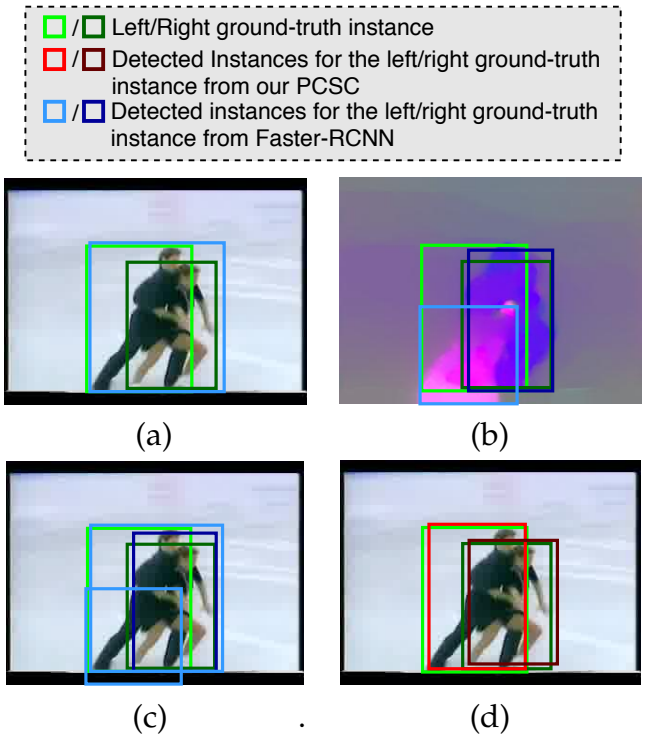


Figure 3.4: An example of visualization results from different methods for one frame. (a) single-stream Faster-RCNN (RGB stream), (b) single-stream Faster-RCNN (flow stream), (c) a simple combination of two-stream results from Faster-RCNN [38], and (d) our PCSC. (Best viewed on screen.)

3.4.5 Runtime Analysis

We report the detection speed of our action detection method in spatial domain and our temporal boundary refinement method for detecting action tubes when using different number of stages and the results are provided in Table 3.11 and Table 3.12, respectively. We observe that the detection speed of both our action detection method in the spatial domain and the temporal boundary refinement method for detecting action tubes slightly decreases as the number of stages increases (note that Stage 0 in Table 3.11 corresponds to the baseline method Faster-RCNN [38] and Stage 0 in Table 3.12 corresponds to the results from our Initial Segment Generation module). Therefore, there is a trade-off between the localization performance and the speed by using the optimal number of stages.

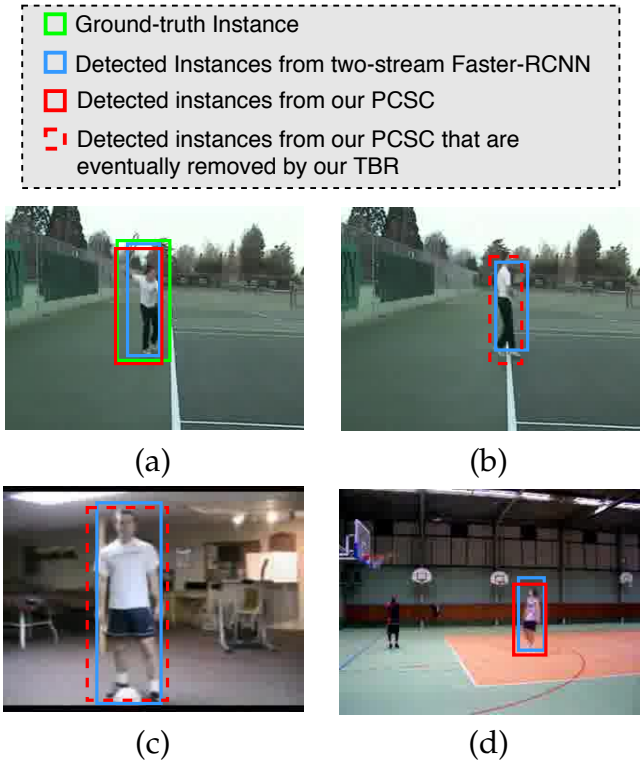


Figure 3.5: Example results from our PCSC with temporal boundary refinement (TBR) and the two-stream Faster-RCNN method. There is one ground-truth instance in (a) and no ground-truth instance in (b-d). In (a), both of our PCSC with TBR method and the two-stream Faster-RCNN method [38] successfully capture the ground-truth action instance "Tennis Swing". In (b) and (c), both of our PCSC method and the two-stream Faster-RCNN method falsely detect the bounding boxes with the class label "Tennis Swing" for (b) and "Soccer Juggling" for (c). However, our TBR method can remove the falsely detected bounding boxes from our PCSC in both (b) and (c). In (d), both of our PCSC method and the two-stream Faster-RCNN method falsely detect the action bounding boxes with the class label "Basketball" and our TBR method cannot remove the falsely detected bounding box from our PCSC in this failure case. (Best viewed on screen.)

3.4.6 Qualitative Results

In addition to the quantitative performance comparison, we also provide some visualization results to further demonstrate the performance of our method. In Fig. 3.4, we show the visualization results generated by using either single RGB or flow stream, simple combination of two streams

and our two-stream cooperation method. The detection method using only the RGB or flow stream leads to either missing detection (see Fig. 3.4 (a)) or false detection (see Fig. 3.4 (b)). As shown in Fig. 3.4 (c), a simple combination of these two results can solve the missing detection problem but cannot avoid the false detection. In contrast, our action detection method can refine the detected bounding boxes produced by either stream and remove some falsely detected bounding boxes (see Fig. 3.4 (d)).

Moreover, the effectiveness of our temporal refinement method is also demonstrated in Figure 3.5. We compare the results from our method and the two-stream Faster-RCNN method [38]. Although both our PCSC method and the two-stream Faster-RCNN method can correctly detect the ground-truth action instance (see Fig. 3.5 (a)), after the refinement process by using our temporal boundary refinement module (TBR), the actionness score for the correctly detected bounding box from our PCSC is further increased. On the other hand, because of the similar scene and motion patterns in the foreground frame (see Fig. 3.5 (a)) and the background frame (see Fig. 3.5 (b)), both our PCSC method and the two-stream Faster-RCNN method generate bounding boxes in the background frame, although there is no ground-truth action instance. However, for the falsely detected bounding box from our PCSC method, our temporal boundary refinement (TBR) module can significantly decrease the actionness score from 0.74 to 0.04, which eventually removes the falsely detected bounding box. The similar observation can be found in Fig. 3.5 (c).

While our method can remove some falsely detected bounding boxes as shown in Fig. 3.5 (b) and Fig. 3.5 (c), there are still some challenging cases that our method cannot work (see Fig. 3.5 (d)). Although the actionness score of the falsely detected action instance could be reduced after the temporal refinement process by using our TBR module, it could not be completely removed due to the relatively high score. We also observe that the state-of-the-art methods (e.g., two-stream Faster-RCNN [38]) cannot work well for this challenging case, either. Therefore, how to reduce false detection from the background frames is still an open issue,

which will be studied in our future work.

3.5 Summary

In this chapter, we have proposed the Progressive Cross-stream Cooperation (PCSC) framework to improve the spatio-temporal action localization results, in which we first use our PCSC framework for spatial localization at the frame level and then apply our temporal PCSC framework for temporal localization at the action tube level. Our PCSC framework consists of several iterative stages. At each stage, we progressively improve action localization results for one stream (*i.e.*, RGB/flow) by leveraging the information from another stream (*i.e.*, RGB/flow) at both region proposal level and feature level. The effectiveness of our newly proposed approaches is demonstrated by extensive experiments on both UCF-101-24 and J-HMDB datasets.

Chapter 4

PCG-TAL: Progressive Cross-granularity Cooperation for Temporal Action Localization

There are two major lines of works, *i.e.*, anchor-based and frame-based approaches, in the field of temporal action localization. But each line of works is inherently limited to a certain detection granularity and cannot simultaneously achieve high recall rates with accurate action boundaries. In chapter work, we propose a progressive cross-granularity cooperation (PCG-TAL) framework to effectively take advantage of complementarity between the anchor-based and frame-based paradigms, as well as between two-view clues (*i.e.*, appearance and motion). Specifically, our new Anchor-Frame Cooperation (AFC) module can effectively integrate both two-granularity and two-stream knowledge at the feature and proposal levels, as well as within each AFC module and across adjacent AFC modules. Specifically, the RGB-stream AFC module and the flow-stream AFC module are stacked sequentially to form a progressive localization framework. The whole framework can be learned in an end-to-end fashion, whilst the temporal action localization performance can be gradually boosted in a progressive manner. Our newly proposed framework outperforms the state-of-the-art methods on three benchmark datasets THUMOS14, ActivityNet v1.3 and UCF-101-24, which clearly demonstrate the effectiveness of our framework.

4.1 Background

4.1.1 Action Recognition

Video action recognition is one important research topic in the community of video analysis. With the prominent development of deep learning in recent years, a large amount of works [84, 12, 64, 81, 4, 77, 56] were proposed by applying the convolutional neural networks to solve this classification problem. It is well-known that both the appearance and motion clues are complementary to each other and discriminant for video action recognition. The two-stream networks [12, 64, 81] exploited the RGB frames and the optical flow maps to extract appearance and motion features, and then combined these two streams for complementary information exchange from both modalities. Additionally, the works in [84, 4, 77, 56] employed 3D convolutional neural layers to instantaneously model spatio-temporal information for videos. However, action recognition is limited to the trimmed input videos, where one action is spanned over the entire video. In addition, the action recognition models can also be applied to extract frame-level or snippet-level features from untrimmed videos. For example, the temporal action localization method just employ the I3D model [4] for base feature extraction.

4.1.2 Spatio-temporal Action Localization

The spatio-temporal action localization task aims to generate action tubes for the predicted action categories with their spatial locations and temporal durations. Most works [67, 66, 55] used the object detection framework to spatially localize actions from individual frames and then temporally linked the detected boxes to form an action tube. Several state-of-the-arts works [26, 54, 69] took advantage of the two-stream RCNN framework to improve the spat/io-temporal action localization performances by exploiting the complementary information between the appearance and motion clues.

4.1.3 Temporal Action Localization

Unlike spatio-temporal action localization, temporal action localization focuses on how to accurately detect the starting and ending points of a predicted action instance in the time domain of an untrimmed video. Early works [97, 50] used slide windows to generate segment proposals and then applied the classifiers to classify actions for each proposal. Most recent works can be roughly categorized into two types: (1) [49, 10, 29, 98, 103] used frame-level or snippet-level features to evaluate frame-wise or snippet-wise actionness by predicting the action starting or ending time, which were then used to determine the temporal boundaries of actions for proposal generation; (2) another set of works [9, 15, 85, 5] applied two-stage object detection framework to generate temporal segment proposals, and then utilized boundary regressor for segment refinement and action classifier for action prediction on a given proposal. In the first class, the segments generated based on frame-wise or snippet-wise labels have relatively accurate temporal boundaries as they are determined at a finer level with larger aperture of temporal details. However, these segments often have low recall rate. In the second category, as the proposals are generated by the predefined anchors with various time intervals over the untrimmed videos, they are able to cover most ground-truth instances. Unfortunately, they usually has lower performances in terms of temporal boundary accuracy.

Either line of the aforementioned works can take advantage of inherent merits of the methods in the other line, resulting in the trade-off between the boundary accuracy and recall rate. Some recent attempts, such as CTAP [13] and MGG [46], were proposed to take the complementary merits of both lines of works into consideration. They either used several isolated components with the stage-wise training scheme, such as CTAP [13], or simply employed frame-wise or snippet-wise labels to filter the boundaries of anchor-based segment proposals [46] but without fully exploring the complementarity in a more systematic way. In our work, our Anchor-Frame Cooperation (AFC) module can be end-to-end optimized and also allows rich feature-level and segment-level

interactions between action semantics extracted from these two complementary lines of works, which offers a significant performance gain toward our temporal action localization framework. Two-stream fusion is another cue to enhance the performance. Recent state-of-the-art methods [5, 99] used late fusion as the only operation to combine complementary information from two modalities, where each stream is trained independently. In our work, each stream progressively and explicitly helps the other stream at the feature level and the segment proposal level. Two streams can also be trained collaboratively in an end-to-end fashion, which is thus beneficial to fully exploit the complementarity between the appearance and motion clues.

4.2 Our Approach

In the following section, we will at first briefly introduce our overall temporal action localization framework (in Sec. 4.2.1), and then present how to perform progressive cross-granularity cooperation (in Sec. 4.2.2), as well as the newly proposed Anchor-Frame Cooperation (AFC) module (in Sec. 4.2.3). The training details and discussions will be provided in Sec. 4.2.4 and Sec. 4.2.5, respectively.

4.2.1 Overall Framework

We denote an untrimmed video as $\mathbf{V} = \{\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{T}$, where $\mathbf{I}_t$ is an RGB frame at time t, with the height of H and the width of W. We seek to detect the action categories $\{\hat{y}_i\}_{i=1}^{N}$ and their temporal durations $\{(\hat{t}_{i,s}, \hat{t}_{i,e})\}_{i=1}^{N}$ in this untrimmed video, where N is the total number of detected action instances. $\hat{t}_{i,s}$ and $\hat{t}_{i,e}$ indicate the starting and ending time of each detected action $\hat{y}_i$.

The proposed progressive cross-granularity cooperation framework builds upon the effective cooperation strategy between anchor-based localization methods and frame-based schemes. We briefly introduce each component as follows:

Base Feature Extraction. For each video $\mathbf{V}$, we extract its RGB and flow features $\mathbf{B}^{\text{RGB}}$ and $\mathbf{B}^{\text{Flow}}$ from a pre-trained and fixed feature extractor (*e.g.*, the I3D feature extractor [4]). The size of each feature is $C \times T$, where C indicates the number of channels for the feature.

Anchor-based Segment Proposal Generation. This module provides the anchor-based knowledge for the subsequent cooperation stage, which adopts the structure as that in TAL-Net [5]. This module is based on the base features $\mathbf{B}^{\text{RGB}}$ and $\mathbf{B}^{\text{Flow}}$, and each stream generates a set of temporal segment proposals of interest (called proposals in later text), such as $\mathcal{P}^{\text{RGB}} = \{\mathbf{p}_i^{\text{RGB}} | \mathbf{p}_i^{\text{RGB}} = (t_{i,s}^{\text{RGB}}, t_{i,e}^{\text{RGB}})\}_{i=1}^{N^{\text{RGB}}}$ and $\mathcal{P}^{\text{Flow}} = \{\mathbf{p}_i^{\text{Flow}} | \mathbf{p}_i^{\text{Flow}} = (t_{i,s}^{\text{Flow}}, t_{i,e}^{\text{Flow}})\}_{i=1}^{N^{\text{Flow}}}$. N^{RGB} and N^{Flow} are the number of proposals for the two streams. We then extract the anchor-based features $\mathbf{P}_0^{\text{RGB}}/\mathbf{P}_0^{\text{Flow}}$ for each stream by using two temporal convolutional layers on top of $\mathbf{B}^{\text{RGB}}/\mathbf{B}^{\text{Flow}}$, which can be readily used as the input to a standalone "Detection Head" (described in Section 4.2.3) after a segment-of-interest pooling operation [5]. The outputs from the detection head are the refined action segments $\mathcal{S}_0^{\text{RGB}}/\mathcal{S}_0^{\text{Flow}}$ and the predicted action categories.

Frame-based Segment Proposal Generation. The frame-based knowledge come from a frame-based proposal generation module. Based on the base features $\mathbf{B}^{\text{RGB}}$ and $\mathbf{B}^{\text{Flow}}$, we apply two 1D convolutional layers (similar as [46]) along the temporal dimension to extract the frame-based features for both streams, termed as $\mathbf{Q}_0^{\text{RGB}}$ and $\mathbf{Q}_0^{\text{Flow}}$, respectively. The frame-based feature in each stream is then processed by two parallel 1×1 convolutions together with the `sigmoid` activation functions, which outputs the probability distributions of temporal starting and ending positions of an action, respectively. We then gather a set of temporal segment proposals for each stream, *i.e.*, $\mathcal{Q}_0^{\text{RGB}}$ and $\mathcal{Q}_0^{\text{Flow}}$, respectively, after pairing highly confident starting and ending indices with valid durations, and removing redundant pairs by non-maxium suppression (NMS).

Progressive Cross-granularity Cooperation Framework. Given the two-stream features and proposals by the preceding two proposal generation

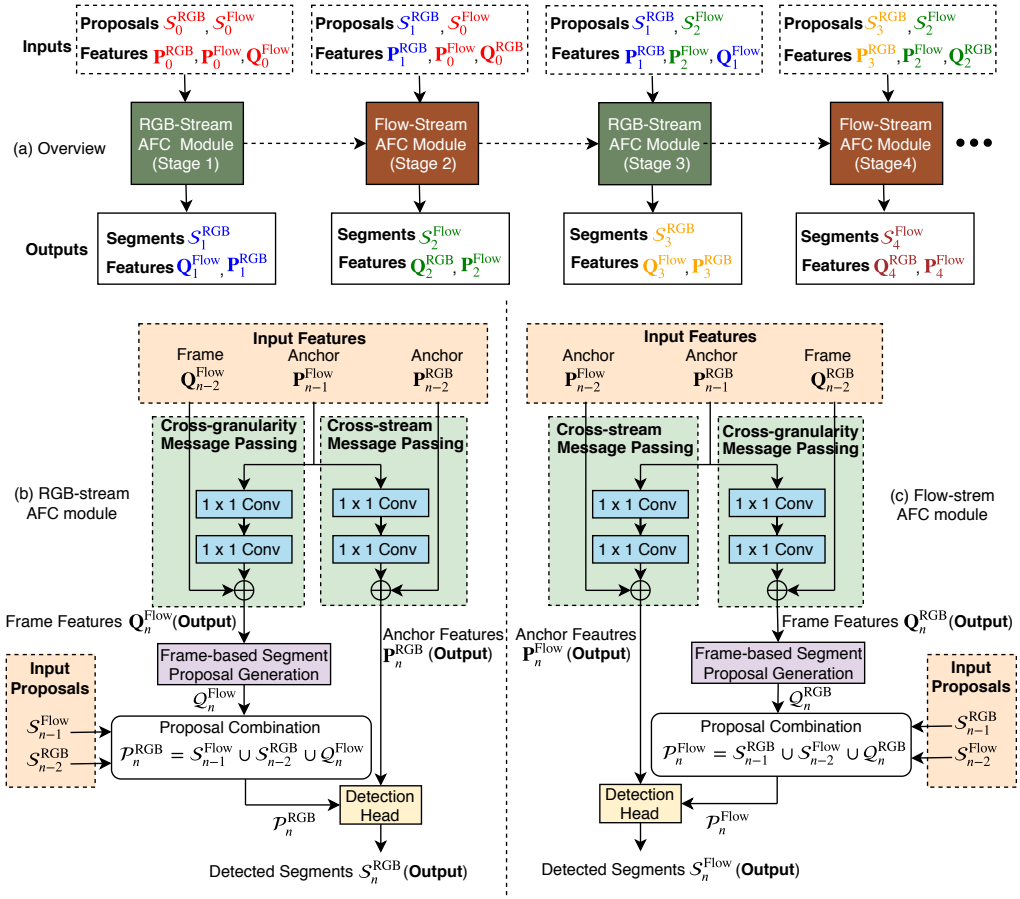


Figure 4.1: (a) Overview of our Progressive Cross-granularity Cooperation framework. At each stage, our Anchor-Frame Cooperation (AFC) module focuses on only one stream (*i.e.*, RGB/Flow). The RGB-stream AFC modules at Stage 1 and 3(*resp.*, the flow-stream AFC modules at Stage 2 and 4) generate anchor-based features and segments as well as frame-based features for the RGB stream (*resp.*, the Flow stream). The inputs of each AFC module are the proposals and features generated by the previous stages. (b) (c) Details of our RGB-stream and Flow stream AFC modules at different stages n. Note that we use the RGB-stream AFC module in (b) when n is an odd number and the Flow-stream AFC module in (c) when n is an even number. For better representation, when $n = 1$, the initial proposals $\mathcal{S}_{n-2}^{\text{RGB}}$ and features $\mathbf{P}_{n-2}^{\text{RGB}}, \mathbf{Q}_{n-2}^{\text{Flow}}$ are denoted as $\mathcal{S}_0^{\text{RGB}}, \mathbf{P}_0^{\text{RGB}}$ and $\mathbf{Q}_0^{\text{Flow}}$. The initial proposals and features from anchor-based and frame-based branches are obtained as described in Section 4.2.3. In both (b) and (c), the frame-based and anchor-based features are improved by the cross-granularity and cross-stream message passing modules, respectively. The improved frame-based features are used to generate better frame-based proposals, which are then combined with anchor-based proposals from both RGB and flow streams. The newly combined proposals together with the improved anchor-based features are used as the input to "Detection Head" to generate the updated segments.

methods, our method progressively uses the newly proposed Anchor-Frame Cooperation (AFC) module (described in Sec. 4.2.3) to output a series of gradually improved localization results. Note that while each AFC module focuses on knowledge exchange within one stream, there are sufficient cross-granularity and cross-stream cooperations within each AFC module and between two adjacent AFC modules. After stacking a series of AFC modules, the final action localization results are obtained after performing the NMS operation on the union set of the output action segments across all stages.

4.2.2 Progressive Cross-Granularity Cooperation

The proposed progressive cross-granularity cooperation framework aims at encouraging rich feature-level and proposal-level collaboration simultaneously from two-granularity scheme (*i.e.*, the anchor-based and frame-based temporal localization methods), and from two-stream appearance and motion clues (*i.e.*, RGB and flow).

To be specific, this progressive cooperation process is performed by iteratively stacking multiple AFC modules with each focusing on one individual stream. As shown in Fig. 4.1(a), the first AFC stage focuses the on RGB stream, and the second AFC stage on flow stream, which are respectively referred to as RGB-stream AFC and flwo-stream AFC for better presentation. For example, the RGB-stream AFC at stage n (n is an odd number) employs the preceding two-stream detected action segments $\mathcal{S}_{n-1}^{\text{Flow}}$ and $\mathcal{S}_{n-2}^{\text{RGB}}$ as the input two-stream proposals, and uses preceding two-stream anchor-based features $\mathbf{P}_{n-1}^{\text{Flow}}$ and $\mathbf{P}_{n-2}^{\text{RGB}}$ and flow-stream frame-based feature $\mathbf{Q}_{n-2}^{\text{Flow}}$ as the input features. This AFC stage will output the RGB-stream action segments $\mathcal{S}_{n}^{\text{RGB}}$ and anchor-based feature $\mathbf{P}_{n}^{\text{RGB}}$, as well as update the flow-stream frame-based features as $\mathcal{Q}_{n}^{\text{Flow}}$. These outputs will be used as the inputs to the subsequent AFC stages, thus enabling cross-granularity and cross-stream cooperation over multiple AFC stages.

In each AFC stage, which will be discussed in Sec. 4.2.3, we introduce a set of feature-level message passing and proposal-level cooperation strategies to effectively integrate the complementary knowledge from the anchor-based branch to the frame-based branch through "Cross-granularity Message Passing", and then back to the anchor-based branch. We argue that, within each AFC stage, this particular structure will seamlessly encourage collaboration through "Proposal Combination" (see Sec. 4.2.3 for more details) by exploiting cross-granularity and cross-stream complementarty, at both the feature and proposal levels, as well as between two temporal action localization schemes.

Since the rich cooperations can be performed within and across adjacent AFC modules, the successes in detecting temporal segments at the earlier AFC stages will be faithfully propagated to the subsequent AFC stages, thus the proposed progressive localization strategy can gradually improve the localization performances.

4.2.3 Anchor-frame Cooperation Module

The Anchor-frame Cooperation (AFC) module is based on the intuition that complementary knowledge from two-stream anchor-based and frame-based temporal localization methods would not only improve the action boundary localization accuracy but also increase the recall rate of the detected action segments. Therefore, the AFC module is designed to mine the rich cross-granularity and cross-stream knowledge at both the feature and proposal levels, also between the aforementioned two lines of action localization methods. To be specific, there are RGB-stream and flow-stream AFC modules, according to the main stream that an AFC module would like to focus on. In this section, we will present the detailed structures of two types of AFC modules, as shown in Fig. 4.1(b-c).

Initialization. As the AFC module is applied in a progressive manner, it uses the inputs from the preceding AFC stages. However, for the first stage (*i.e.*, $n = 1$), some inputs do not have valid predecessors. Instead, we use the initial anchor-based and frame-based two-stream features

$\mathbf{P}_0^{\text{RGB}}/\mathbf{P}_0^{\text{Flow}}$ and $\mathbf{Q}_0^{\text{RGB}}/\mathbf{Q}_0^{\text{Flow}}$, as well as proposals $\mathcal{S}_0^{\text{RGB}}/\mathcal{S}_0^{\text{Flow}}$ as the input, as shown in Sec. 4.2.1 and Fig. 4.1(a).

Single-stream Anchor-to-Frame Message Passing (*i.e.*, "Cross-granularity Message Passing in Fig. 4.1(b-c)). Taking the RGB-stream AFC module as an example. The network structure is shown in Fig. 4.1(b). About the feature-level message passing, we use a simple message passing structure to propagate the feature-level message from the anchor-based features $\mathbf{P}_{n-1}^{\text{Flow}}$ to the frame-based features $\mathbf{Q}_{n-2}^{\text{Flow}}$, both from the flow-stream. The message passing operation is defined as

$$\mathbf{Q}_n^{\text{Flow}} = \mathcal{M}_{anchor \to frame}(\mathbf{P}_{n-2}^{\text{Flow}}) \oplus \mathbf{Q}_{n-2}^{\text{Flow}}. \tag{4.1}$$

$\oplus$ is the element-wise addition operation, and $\mathcal{M}_{anchor \to frame}(\bullet)$ is the flow-stream message passing function from the anchor-based branch to the frame-based branch, which is formulated as two stacked 1×1 convolutional layers with `ReLU` non-linear activation functions. This improved feature $\mathbf{Q}_n^{\text{Flow}}$ can generate more accurate action starting and ending probability distributions (a.k.a., more accurate action durations) as they absorb the temporal duration-aware anchor-based features, which can generate better frame-based proposals $\mathcal{Q}_n^{\text{Flow}}$, as described in Sec. 4.2.1. For the flow-stream AFC module (see Fig. 4.1(c)), the message passing process is similarly defined as $\mathbf{Q}_n^{\text{RGB}} = \mathcal{M}_{anchor \to frame}(\mathbf{P}_{n-2}^{\text{RGB}}) \oplus \mathbf{Q}_{n-2}^{\text{RGB}}$, which flips the superscripts from Flow to RGB. $\mathbf{Q}_n^{\text{RGB}}$ also leads to better frame-based RGB-stream proposals $\mathcal{Q}_n^{\text{RGB}}$.

Cross-stream Frame-to-Anchor Proposal Cooperation (*i.e.*, "Proposal Combination" in Fig. 4.1(b-c)). We also use the RGB-stream AFC module for instance, which is shown in Fig. 4.1(b), to enable cross-granularity two-stream proposal-level cooperation, we gather sufficient valid frame-based proposals $\mathcal{Q}_n^{\text{Flow}}$ by using the "Frame-based Segment Proposal Generation" method described in Sec. 4.2.1 based on the improved frame-based features $\mathbf{Q}_n^{\text{Flow}}$. And then we combine $\mathcal{Q}_n^{\text{Flow}}$ together with the two-stream anchor-based proposals $\mathcal{S}_{n-1}^{\text{Flow}}$ and $\mathcal{S}_{n-2}^{\text{RGB}}$ to form an updated set of anchor-based proposals at stage n, *i.e.*, $\mathcal{P}_n^{\text{RGB}} = \mathcal{S}_{n-2}^{\text{RGB}} \cup \mathcal{S}_{n-1}^{\text{Flow}} \cup$

$\mathcal{Q}_n^{\text{Flow}}$. This set of proposals not only provides more comprehensive proposals but also guides our approach to pool more accurate proposal features, both of which would significantly benefit the subsequent boundary refinement and action class prediction procedures in the detection head. The flow-stream AFC module also perform the similar process as in the RGB-stream AFC module, *i.e.*, the anchor-based flow-stream proposals are updated as $\mathcal{P}_n^{\text{Flow}} = \mathcal{S}_{n-2}^{\text{Flow}} \cup \mathcal{S}_{n-1}^{\text{RGB}} \cup \mathcal{Q}_n^{\text{RGB}}$.

Cross-stream Anchor-to-Anchor Message Passing (*i.e.*, "Cross-stream Message Passing" in Fig. 4.1(b-c)). In addition to anchor-frame cooperation, we also include two-stream feature cooperation module. The two-stream anchor-based features are updated similarly as equation (4.1) by a message passing operation, *i.e.*, $\mathbf{P}_n^{\text{RGB}} = \mathcal{M}_{flow \to rgb}(\mathbf{P}_{n-1}^{\text{Flow}}) \oplus \mathbf{P}_{n-2}^{\text{RGB}}$ for the RGB stream, and $\mathbf{P}_n^{\text{Flow}} = \mathcal{M}_{rgb \to flow}(\mathbf{P}_{n-1}^{\text{RGB}}) \oplus \mathbf{P}_{n-2}^{\text{Flow}}$ for flow stream.

Detection Head. The detection head aims at refining the segment proposals and predicting the action categories for each proposal. Its inputs are the segment proposal features generated by segment-of-interest (SOI) pooling [5] with respect to the updated proposals $\mathcal{P}_n^{\text{RGB}}/\mathcal{P}_n^{\text{Flow}}$ and the updated anchor-based features $\mathbf{P}_n^{\text{RGB}}/\mathbf{P}_n^{\text{Flow}}$. To increase the perceptual aperture of action boundaries, we add two more SOI pooling operations around the starting and ending indices of the proposals with a given interval (in this paper, it is fixed to 10), and then concatenate these pooled features after the segment proposal features. The concatenated segment proposal features are fed to the segment regressor for segment refinement and the action classifier for action prediction. We also use GCN [99] to explore the proposal-to-proposal relations, which additionally provides performance gains independent to our cooperation schemes.

4.2.4 Training Details

At each AFC stage, the training losses are used at both anchor-based and frame-based branches. To be specific, the training loss in each anchor-based branch has two components: one is the cross-entropy loss for the action classification task and the other is the smooth-L_1 loss for the

boundary regression task. On the other hand, the training loss in each frame-based branch is also an cross-entropy loss, which indicates the frame-wise action starting and ending positions. We sum up all losses and jointly train our model. We follow the strategy in [46] to assign the ground-truth labels for supervision at all AFC modules.

4.2.5 Discussions

Variants of the AFC Module. Besides using the aforementioned AFC module, as illustrated in Fig. 4.1(b-c), one can also simplify the structure by either removing the entire frame-based branch or the cross-stream cooperation module (termed as "`w/o CG`" and "`w/o CS`" in Sec. 4.3.3, respectively). In the first case ("w/o CG"), the variant looks like we incorporate the progressive feature and proposal cooperation strategy within the two-stream anchor-based localization framework, thus is to some extent similar to the method proposed in [69]. But this variant is totally applied in the temporal domain for temporal action localization, while the work in [69] is conducted in spatial domain for spatial action localization. We argue that our framework after incorporating the newly proposed cross-granularity cooperation strategy significantly improves this variant. Since it enhances the boundary-sensitive frame-based features by leveraging anchor-based features with more duration-aware characteristics, and in turn help generate high-quality anchor-based proposals with better action boundaries and high recall rates. Thus it is more likely that the final localization results can capture accurate action segments. In the second case ("w/o CS"), the variant applies a simple two-stream fusion strategy by aggregating RGB and flow features as lengthy feature vectors before the first stage instead of using the "Cross-stream Message Passing" block for fusing two-stream features. But this variant suffers from early performance saturation (namely the localization results become stable after the second stage as shown in Table 4.5), which is also inferior to our AFC. We compare our method with these two variants in our experiments, and show that our AFC framework outperforms these two variants.

Similarity of Results Between Adjacent Stages. Note that even though

each AFC module focuses on one stream, the rich cross-stream interactions eventually make the two-stream features and proposals be similar to each other. By applying more AFC stages, it can encourage more accurate proposals and more representative action features from gradually absorbed cross-stream information and cross-granularitry knowledges, and thus significantly boosts the performance of temporal action localization. But the performance gain will be less significant after a certain number of stages. Empirically, we observe a three-stage design achieves the best localization results.

Number of Proposals. The number of proposals will not rapidly increase when the number of stages increases, because the NMS operations applied in previous detection heads remarkably reduce redundant segments but still preserve highly confident temporal segments. But the number of proposals will not decrease as well, as the proposed proposal cooperation strategy still absorbs sufficient number of proposals from complementary sources.

4.3 Experiment

4.3.1 Datasets and Setup

Datasets. We evaluate our temporal action localization framework on three datasets: THUMOS14 [25], ActivityNet v1.3 [3] and UCF-101-24 [68].

- *THUMOS14* contains 1,010 and 1,574 untrimmed videos from 20 action classes for validation and testing, respectively. We use 200 temporally annotated videos in the validation set as the training samples, and 212 temporally annotated videos in the test set as the testing samples.

- *ActivityNet v1.3* contains 19,994 videos from 200 different activities. The whole video set is divided into training, validation and testing sets with the ratio of 2:1:1. We evaluate our model on the validation set as the annotation for the testing set is not publicly available.

- *UCF-101-24* consists of 3,207 untrimmed videos from 24 action classes with spatio-temporal annotations provided by [67]. We use an

existing spatial action detection method [69] to spatially locate actions at each frame and generate action tubes by linking the detection results across frames. We consider these action tubes as untrimmed videos and use our model to identify the temporal action boundaries. We report and compare the results based on split 1.

Implementation Details. We use a two-stream I3D [4] model pretrained on Kinetics as the feature extractor to extract base features. For training, we set the size of minibatch as 256 for the anchor-based proposal generation network and 64 for detection head, respectively. The initial learning rate is 0.001 for both the THUMOS14 and the UCF-101-24 datasets, and 0.005 for the ActivityNet v1.3. During training, we divide the learning rate by 10 for every 12 epoches.

Evaluation Metrics. We evaluate our model by mean average precision (mAP). A detected action segment or action tube is considered as correct if its overlapping with the ground-truth instances is larger than a threshold δ, and its action label is correctly predicted. To measure the overlapping area, we use temporal intersection-over-union (tIoU) for temporal action localization on THUMOS14 and ActivityNet v1.3 and spatio-temporal intersection-over-union (st-IoU) for spatio-temporal action localization on UCF-101-24.

4.3.2 Comparison with the State-of-the-art Methods

We compare our PCG-TAL framework with the state-of-the-art methods on THUMOS14, ActivityNet v1.3 and UCF-101-24. All results of the existing methods are quoted from their original papers.

THUMOS14. We report the results on the THUMOS14 in Table 4.1. Our method PCG-TAL outperforms the existing methods at all tIoU thresholds. Similar to our method, both TAL-Net [5] and P-GCN [99] use I3D features as their base features and apply a two-stream framework and late fusion strategy to exploit the complementary information between appearance and motion clues. When using the tIoU threshold $\delta = 0.5$,

Table 4.1: Comparison (mAP %) on the THUMOS14 dataset using different tIoU thresholds

tIoU thresh. δ	0.1	0.2	0.3	0.4	0.5
SCNN [63]	47.7	43.5	36.3	28.7	19.0
REINFORCE[95]	48.9	44.0	36.0	26.4	17.1
CDC [62]	-	-	40.1	29.4	23.3
SMS[98]	51.0	45.2	36.5	27.8	17.8
TRUN [15]	54.0	50.9	44.1	34.9	25.6
R-C3D [85]	54.5	51.5	44.8	35.6	28.9
SSN [103]	66.0	59.4	51.9	41.0	29.8
BSN [37]	-	-	53.5	45.0	36.9
MGG [46]	-	-	53.9	46.8	37.4
BMN [35]	-	-	56.0	47.4	38.8
TAL-Net [5]	59.8	57.1	53.2	48.5	42.8
P-GCN [99]	69.5	67.8	63.6	57.8	49.1
ours	**71.2**	**68.9**	**65.1**	**59.5**	**51.2**

our method outperforms TAL-Net and P-GCN by 8.4% and 2.1%, respectively. The improvement largely comes from our well designed progressive cooperation framework that fully exploits the cross-granularity information between anchor-based and frame-based methods. However, the mAP of our method is 13.8% higher than that of MGG, possibly because MGG can not well capture the complementary information between these two methods (*i.e.*, the frame actionness producer and the segment proposal producer). It is worth mentioning that, following P-GCN [99], we also apply the GCN module in [99] to consider proposal-to-proposal relations in our detection head, and our method can still achieve mAP of 48.37% without using the GCN module, which is comparable to the performance of P-GCN.

ActivityNet v1.3. We also compare the mAPs at different tIoU thresholds $\delta = \{0.5, 0.75, 0.95\}$ on the validation set of the ActivityNet v1.3 dataset. Besides, we report average mAP as well, which is calculated by averaging all mAPs at multiple tIoU thresholds δ from 0.5 to 0.95 with an interval of 0.05. All the results are shown in Table 4.2. In terms of the average mAP, our method can achieve mAP of 28.85%, which outperforms P-GCN [99], SSN [103] and TAL-Net [5] by 1.86%, 6.87% and 8.63%, respectively. In Table 4.3, it is observed that both BSN [37] and

Table 4.2: Comparison (mAPs %) on the ActivityNet v1.3 (val) dataset using different tIoU thresholds. No extra external video labels are used.

tIoU threshold δ	0.5	0.75	0.95	Average
CDC [62]	**45.30**	26.00	0.20	23.80
TCN [9]	36.44	21.15	3.90	-
R-C3D [85]	26.80	-	-	-
SSN [103]	39.12	23.48	**5.49**	23.98
TAL-Net [5]	38.23	18.30	1.30	20.22
P-GCN[99]	42.90	28.14	2.47	26.99
ours	44.31	**29.85**	5.47	**28.85**

Table 4.3: Comparison (mAPs %) on the ActivityNet v1.3 (val) dataset using different tIoU thresholds. The external video labels from UntrimmedNet [80] are applied.

tIoU threshold δ	0.5	0.75	0.95	Average
BSN [37]	46.45	29.96	8.02	30.03
P-GCN [99]	48.26	33.16	3.27	31.11
BMN [35]	50.07	34.78	**8.29**	33.85
ours	**52.04**	**35.92**	7.97	**34.91**

BMN [35] are superior than the best result shown in Table 4.2. However, BSN and BMN have to rely on the extra external video-level action labels from untrimmedNet [80]. Similarly, the performance of our method can be boosted by using extra external labels as in [35] and proposals from BMN, whose average mAP (34.91%) is better than BSN and BMN.

UCF-101-24. To further evaluate our method, we conduct another comparison for the spatio-temporal action localization task, on the UCF-101-24 dataset [68]. We report mAPs at st-IoU threshold $\delta = \{0.2, 0.5, 0.75\}$ and average mAP when δ is set between 0.5 and 0.95 with an interval of 0.05. The input action tubes before temporal refinement were provided by the authors in [69]. The results are reported in Table 4.4. As can be seen, our method achieves better results then the state-of-the-art methods over all st-IoU thresholds, which shows that our proposed method can also help improve the spatio-temporal action localization results. Note that our method and the temporal refinement module in [69] use the same action tubes and the mAP of our method is 2.5% higher at the st-IoU threshold $\delta = 0.5$.

Table 4.4: Comparison (mAPs %) on the UCF-101-24 dataset using different st-IoU thresholds.

st-IoU thresh. δ	0.2	0.5	0.75	0.5:0.95
LTT[83]	46.8	-	-	-
MRTSR[54]	73.5	32.1	02.7	07.3
MSAT[60]	66.6	36.4	0.79	14.4
ROAD[67]	73.5	46.3	15.0	20.4
ACT [26]	76.5	49.2	19.7	23.4
TAD[19]	-	59.9	-	-
PCSC[69]	84.3	61.0	23.0	27.8
ours	**84.9**	**63.5**	**23.7**	**29.2**

4.3.3 Ablation study

We perform a set of ablation studies to investigate the contribution of each component in our proposed PCG-TAL framework.

Anchor-Frame Cooperation. We compare the performances among five AFC-related variants and our complete method: (1) `w/o CS-MP`: we remove the "Cross-stream Message Passing" block from our AFC module; (2) `w/o CG-MP`: we remove the "Cross-granularity Message Passing" block from our AFC module; (3) `w/o PC`: we remove the "Proposal Combination" block from our AFC module in which $\mathcal{Q}_n^{\text{Flow}}$ and $\mathcal{Q}_n^{RGB}$ are directly used as the input to"Detection Head" in the RGB-stream AFC module and the flow-stream AFC module, respectively; (4) `w/o CS`: we aggregate two-stream features as lengthy feature vectors for both frame-based and anchor-based branches before the first stage, in which the aggregated features are directly used as the input to the "Cross-granularity Message Passing" block and "Detection Head" in Fig. 4.1(b-c); (5) `w/o CG`: we remove the entire frame-based branch and in this case only two proposal sets based on the RGB and flow features generated from the anchor-based scheme are combined in the "Proposal Combination" block in Fig. 4.1(b-c); (6) `AFC`: our complete method. Note that two alternative methods "`w/o CS`" and "`w/o CS-MP`" have the same network structure in each AFC module except that the input anchor-based features to the "Cross-granularity Message Passing" block and "Detection Head" are respectively $\mathbf{P}_{n-1}^{\text{Flow}}$ and $\mathbf{P}_{n-2}^{\text{RGB}}$ (for the RGB-stream AFC module) and

Table 4.5: Ablation study among different AFC-related variants at different training stages on the THUMOS14 dataset.

Stage	0	1	2	3	4
AFC	44.75	46.92	48.14	**48.37**	48.22
w/o CS	44.21	44.96	45.47	45.48	45.37
w/o CG	43.55	45.95	46.71	46.65	46.67
w/o CS-MP	44.75	45.34	45.51	45.53	45.49
w/o CG-MP	44.75	46.32	46.97	46.85	46.79
w/o PC	42.14	43.21	45.14	45.74	45.77

$\mathbf{P}_{n-1}^{\text{RGB}}$ and $\mathbf{P}_{n-2}^{\text{Flow}}$ (for the flow-stream AFC module) in "`w/o CS-MP`", while the inputs are the same (*i.e.*, the aggregated features) in "`w/o CS`". Also, the discussions for the alternative methods "`w/o CS`" and "`w/o GS`" are provided in Sec. 4.2.5. We take the THUMOS14 dataset at tIoU threshold $\delta = 0.5$ as an example to report the mAPs of different variants at different stages in Table 4.5. Note that we remove the GCN module from our method for fair comparison.

In Table 4.5, the results for stage 0 are obtained based on the initial proposals generated by our complete method and five alternative methods before applying our AFC module. The initial proposals for "`w/o CS-MP`" and "`w/o CG-MP`" are the same as our complete method, so the results of these three methods at stage 0 are the same. The initial proposals are $\mathcal{P}_0^{\text{RGB}} \cup \mathcal{P}_0^{\text{Flow}}$ for "`w/o CG`", and $\mathcal{Q}_0^{\text{RGB}} \cup \mathcal{Q}_0^{\text{Flow}}$ for "`w/o PC`". We have the following observations, (1) the two alternative methods "`w/o CS`" and"`w/o CS-MP`" achieve comparable results when the number of stages increase as these two methods share the same network structure. (2) our complete method outperforms "`w/o CS-MP`", "`w/o CG-MP`" and"`w/o PC`", which demonstrates that it is useful to include three cross-stream and/or cross-granularity operations at both feature and proposal levels in each AFC module within our framework. (3) our complete method also outperforms the alternative method "`w/o CG`", which demonstrates that it is beneficial to encourage cross-granularity collaboration by passing the features from the anchor-based approach to the frame-based scheme and simultaneously providing the segment proposals from the frame-based scheme back to "Detection Head" of the anchor-based approach. (4) at the first few stages, the results of all methods are generally improved

Table 4.6: LoR (%) scores of proposals from adjacent stages. This score is calculated for each video and averaged for all videos in the test set of the THUMOS14 dataset.

Stage	Comparisons	LoR (%)
1	$\mathcal{P}_1^{\text{RGB}}$ *vs.* $\mathcal{S}_0^{\text{Flow}}$	67.1
2	$\mathcal{P}_2^{\text{Flow}}$ *vs.* $\mathcal{P}_1^{\text{RGB}}$	79.2
3	$\mathcal{P}_3^{\text{RGB}}$ *vs.* $\mathcal{P}_2^{\text{Flow}}$	89.5
4	$\mathcal{P}_4^{\text{Flow}}$ *vs.* $\mathcal{P}_3^{\text{RGB}}$	97.1

when the number of stages increases and the results will become stable after 2 or 3 stages.

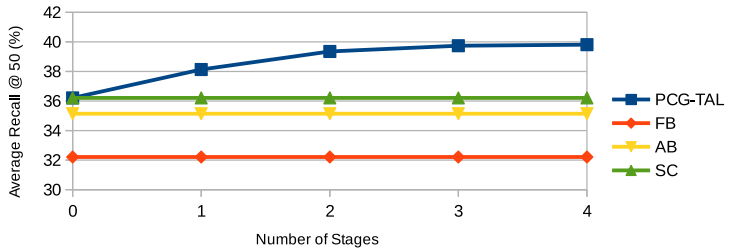
Similarity of Proposals. To evaluate the similarity of the proposals from different stages for our method, we define a new metric named as **Large-over-Ratio (LoR)** score. Taking the proposal sets $\mathcal{P}_1^{\text{RGB}}$ at Stage 1 and $\mathcal{P}_2^{\text{Flow}}$ at Stage 2 as the examples. The LoR score at Stage 2 is the ratio of the **similar** proposals in $\mathcal{P}_2^{\text{Flow}}$ among all proposals in $\mathcal{P}_2^{\text{Flow}}$. A proposal is defined as a similar proposal if its maximum temporal intersection-over-union (max-tIoU) score is higher than a pre-defined threshold 0.7 with the preceding proposals $\mathcal{P}_1^{\text{RGB}}$. The max-tIoU score for one proposal in the corresponding proposal set (say $\mathcal{P}_2^{\text{Flow}}$) is calculated by comparing it with all proposals in the other proposal set (say $\mathcal{P}_1^{\text{RGB}}$), returning the maximum temporal intersection-over-union score. Therefore, a higher LoR score indicates that two sets of proposals from different streams are more similar to each other.

Table 4.6 shows the LoR scores at different stages, where LoR scores at Stage 1 and Stage 3 examine the similarities between RGB-stream proposals against the preceding flow-stream proposals, and those at Stage 2 and Stage 4 indicate the similarities between flow-stream proposals against the preceding RGB-stream proposals. In Table 4.6, we observe that the LoR score gradually increases from 67.1% at Stage 1 to 97.1% at Stage 4. This large LoR score at Stage 4 indicates that the proposals in $\mathcal{P}_4^{\text{Flow}}$ and those in $\mathcal{P}_3^{\text{RGB}}$ are very similar even they are gathered for different streams. Thus little complementary information can be further exploited between these two types of proposals. In this case, we cannot achieve further performance improvements by using our temporal

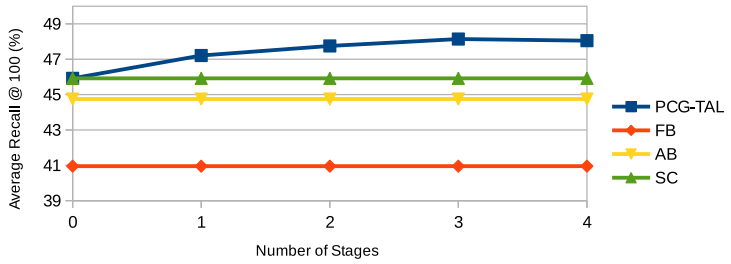
action localization method after Stage 3, which is consistent with the results in Table 4.5.

Quality of the Generated Action Segments. To further analyze our PCG-TAL framework, we evaluate the quality of the action segments generated from our proposed method. We follow the work in [37] to calculate average recall (AR) when using different average number of proposals (AN), which is referred to as AR@AN. On the THUMOS14 dataset [25], we compute the average recall by averaging eleven recall rates based on the tIoU thresholds between 0.5 and 1.0 with an interval of 0.05.

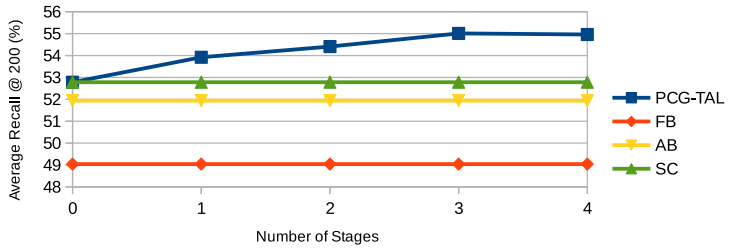
In Fig. 4.2, we compare the AR@AN results of action segments generated from the following baseline methods and our PCG-TAL method at different number of stages: (1) FB: we use only the frame-based method to generate action segments; (2) AB: we use only the anchor-based method to generate action segments; (3) SC: we simply combine the segment proposals from the "FB" method and the "AB" method to generate action segments. Note that the baseline methods "FB", "AB" and "SC" are not based on the multi-stage framework, for better presentation, their average recall rates at stage 0 are the same as their results at stages 1-4. The results from our PGC-TAL method at stage 0 are the results from the "SC" method. We observe that the action segments generated from the "AB' method are better than those from the "FB" method as its average recall rates are higher. Additionally, a simple combination of action segments as used in the "SC" method can lead to higher average recall rates, which demonstrates the complementarity between the frame-based and anchor-based temporal action localization methods. We also observe that the average recall rates from our PCG-TAL method improve as the number of stages increases. The results together with the quantitative results shown in Table 4.5 demonstrate that our newly proposed PCG-TAL framework, which takes advantage of cross-granularity and cross-stream complementary information, can help progressively generate better action segments with higher recall rates and more accurate boundaries.



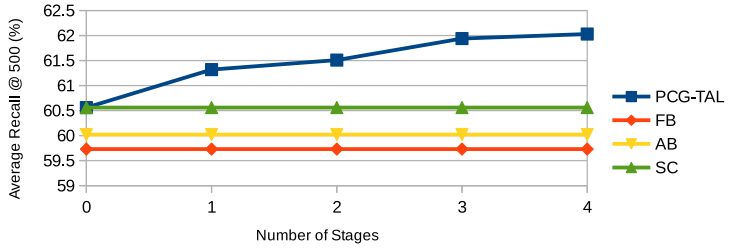
(a) AR@50 (%) for various methods at different number of stages



(b) AR@100 (%) for various methods at different number of stages



(c) AR@200 (%) for various methods at different number of stages



(d) AR@500 (%) for various methods at different number of stages

Figure 4.2: AR@AN (%) for the action segments generated from various methods at different number of stages on the THUMOS14 dataset.

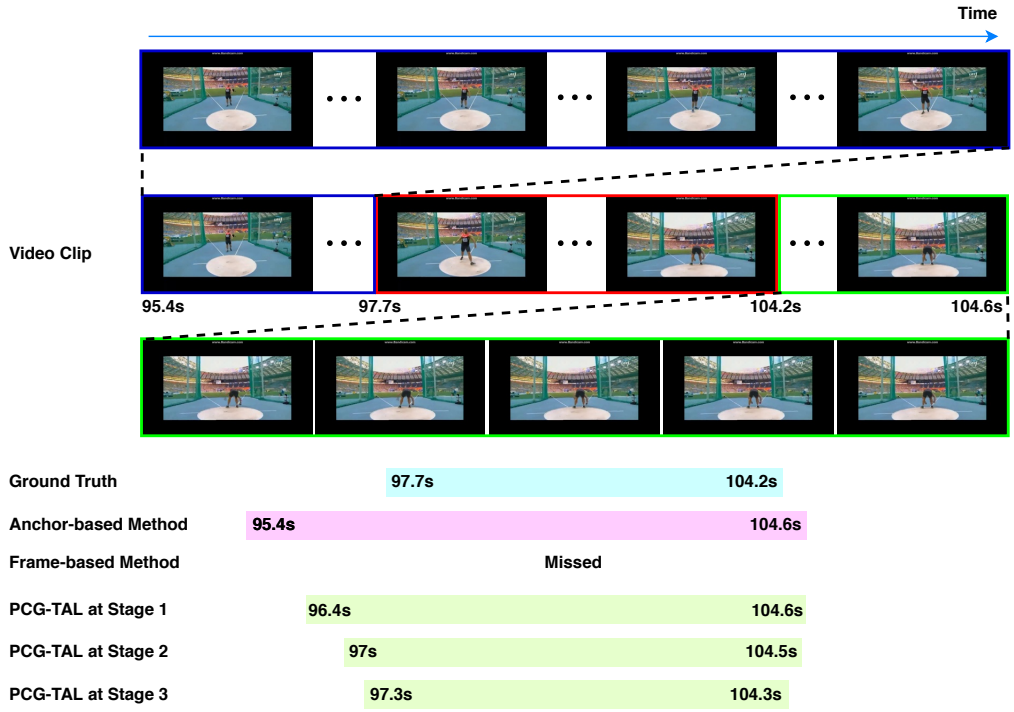


Figure 4.3: Qualitative comparisons of the results from two baseline methods (*i.e.*, the anchor-based method [5] and the frame-based method [37]) and our PCG-TAL method at different number of stages on the THUMOS14 dataset. More frames during the period between the 95.4-th second to the 97.7-th second of the video clip are shown in the blue boxes, where the actor slowly walks into the circle spot to start the action. The true action instance of the action class "Throw Discus" with the starting frame and the ending frame from the 97.7-th second to the 104.2-th second of the video clip are marked in the red box. The frames during the period from the 104.2-th second to the 104.6-th second of the video clip are shown in the green boxes, where the actor just finishes the action and starts to straighten himself up. In this example, the anchor-based method detects the action segment with less accurate boundaries while the frame-based method misses the true action instance. Our PCG-TAL method can progressively improve the temporal boundaries of the action segment.

4.3.4 Qualitative Results

In addition to the quantitative performance comparison, we also provide some visualization results to further demonstrate the performance of our method. In Fig. 4.3, we show the visualization results generated by using either anchor-based method [5] or frame-based method [37], and our

PCG-TAL method at different number of stages. Although the anchor-based method can successfully capture the action instance, it generates action segments with less accurate temporal boundaries. Meanwhile, the frame-based method fails to detect the action instance due to the low prediction probabilities as the starting or ending positions during that period of time. On the other hand, our PCG-TAL method can take advantage of both the anchor-based and frame-based methods to progressively improve the temporal boundaries of the detected action segment, and eventually generate the action segment with more accurate temporal boundaries, which demonstrates the effectiveness of our proposed AFC module.

4.4 Summary

In this chapter, we have proposed a progressive cross-granularity cooperation (PCG-TAL) framework, to gradually improve temporal action localization performance. Our PCG-TAL framework consists of an Anchor-Frame Cooperation (AFC) module to take advantage of both frame-based and anchor-based proposal generation schemes, along with a two-stream cooperation strategy to encourage collaboration between the complementary appearance and motion clues, In our framework, the cooperation mechanisms are conducted in a progressive fashion at both feature level and segment proposal level by stacking multiple AFC modules over different stages. Comprehensive experiments and ablation studies on THUMOS14, ActivityNet v1.3, and UCF-101-24 datasets show the effectiveness of our proposed framework.

Chapter 5

STVGBert: A Visual-linguistic Transformer based Framework for Spatio-temporal Video Grounding

Spatio-temporal video grounding (STVG) aims to localize a spatio-temporal tube of a target object in an untrimmed video based on a query sentence. In this work, we propose a two-step visual-linguistic transformer based framework called STVGBert for the STVG task, which consists of a Spatial Visual Grounding network (SVG-net) and a Temporal Boundary Refinement network (TBR-net). In the first step, the SVG-net directly takes a video and a query sentence as the input and produces the initial tube prediction. In the second step, the TBR-net refines the temporal boundaries of the initial tube to further improve the spatio-temporal video grounding performance. As the key component in our SVG-net, the newly introduced cross-modal feature learning module improves ViLBERT by additionally preserving spatial information when extracting text-guided visual features. Besides, the TBR-net is a novel ViLBERT-based multi-stage temporal localization network, which exploits the complementarity of the anchor-based and the frame-based temporal boundary refinement methods. Different from the existing works for the video grounding tasks, our proposed framework does not rely on any pre-trained object detector. Comprehensive experiments demonstrate our

newly proposed framework outperforms the state-of-the-art methods on two benchmark datasets, VidSTG and HC-STVG.

5.1 Background

5.1.1 Vision-language Modelling

Transformer-based neural networks have been widely explored for various vision-language tasks [75, 32, 33, 47, 70], such as visual question answering, image captioning, and image-text retrieval. Apart from these works designed for image-based visual-linguistic tasks, Sun *et al.* [71] proposed VideoBERT for the video captioning task by modeling temporal variations across multiple video frames. However, this work does not model spatial information within each frame, so that it cannot be applied for the spatio-temporal video grounding task discusseed in this work.

The aforementioned transformer-based neural networks take the features extracted from either Region of Interests (RoIs) in images or video frames as the input features, but spatial information in the feature space cannot be preserved when transforming them to visual tokens. In this work, we propose an improved version of ViLBERT [47] to better model spatio-temporal information in videos and learn better cross-modal representations.

5.1.2 Visual Grounding in Images/Videos

Visual grounding in images/videos aims to localize the object of interest in an image/video based on a query sentence. In most existing methods [42, 96, 45, 82, 87, 88, 41, 86, 7, 102], a pre-trained object detector is often required to pre-generate object proposals. The proposal that best matches the given input description is then selected as the final result. For the visual grounding tasks in images, some recent works [92, 34, 48, 91] proposed new one-stage grounding frameworks without using the pre-trained object detector. For example, Liao *et al.* [34] used the anchor-free object detection method [104] to localize the target objects based on the cross-modal representations. For the video grounding task, Zhang *et*

al. [102] proposed a new method (referred to as STGRN) that does not rely on the pre-generated tube proposals. Unfortunately, this work [102] still requires a pre-trained object detector to first generate object proposals since the output bounding boxes are retrieved from these candidate bounding boxes. In the concurrent work STGVT [76], similar as in our proposed framework, it also adopted a visual-linguistic transformer to learn cross-modal representations for the spatio-temporal video grounding task. But this work [76] also needs to first generate the tube proposals as in most existing methods [7, 86]. Additionally, in the works [59, 30, 101, 90], the pre-trained object detectors are also required to generate object proposals for object relationship modelling.

In contrast to these works [102, 76, 59, 30, 101, 90], in this work, we introduce a new framework with an improved ViLBERT module to generate spatio-temporal object tubes without requiring any pre-trained object detectors. We also propose a temporal boundary refinement network to further refine the temporal boundaries, which has not been explored in the existing methods.

5.2 Methodology

5.2.1 Overview

We denote an untrimmed video $\mathbf{V}$ with KT frames as a set of non-overlapping video clips, namely we have $\mathbf{V} = \{\mathbf{V}_k^{\text{clip}}\}_{k=1}^{K}$, where $\mathbf{V}_k^{\text{clip}}$ indicates the k-th video clip consisting of T frames, and K is the number of video clips. We also denote a textual description as $\mathbf{S} = \{s_n\}_{n=1}^{N}$, where s_n indicates the n-th word in the description $\mathbf{S}$, and N is the total number of words. The STVG task aims to output the spatio-temporal tube $\mathbf{B} = \{\mathbf{b}_t\}_{t=t_s}^{t_e}$ containing the object of interest (*i.e.*, the target object) between the t_s-th frame and the t_e-th frame, where $\mathbf{b}_t$ is a 4-d vector indicating the top-left and bottom-right spatial coordinates of the target bounding box in the t-th frame. t_s and t_e are the temporal starting and ending frame indexes of the object tube $\mathbf{B}$.

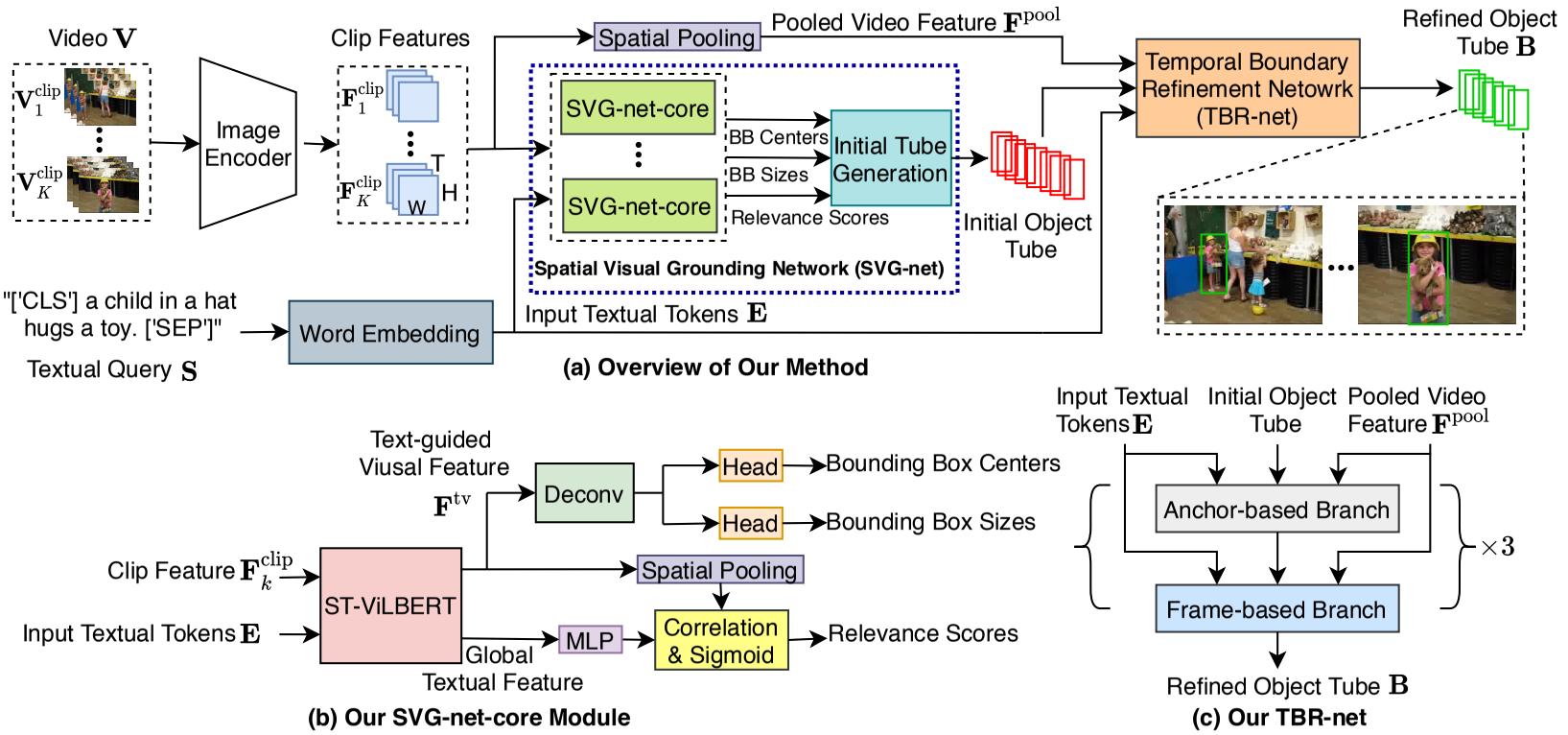


Figure 5.1: (a) The overview of our spatio-temporal video grounding framework STVGBert. Our two-step framework consists of two sub-networks, Spatial Visual Grounding network (SVG-net) and Temporal Boundary Refinement network (TBR-net), which takes video and textual query pairs as the input to generate spatio-temporal object tubes containing the object of interest. In (b), the SVG-net generates the initial object tubes. In (c), the TBR-net progressively refines the temporal boundaries of the initial object tubes and generates the final spatio-temporal object tubes.

In this task, it is required to perform both spatial localization and temporal localization based on the query sentence. Therefore, we propose a two-step framework which consists of two sub-networks, *Spatial Visual Grounding* network (SVG-net) and *Temporal Boundary Refinement* network (TBR-net). As shown in Fig. 5.1, the SVG-net first takes the untrimmed video and the textual description as the input, and predicts the initial object tube, based on which the TBR-net progressively updates the temporal boundaries by using a newly proposed ViLBERT-based multi-stage framework to produce the final spatio-temporal object tube. We will describe each step in details in the following sections.

5.2.2 Spatial Visual Grounding

Unlike the previous methods [86, 7], which first output a set of tube proposals by linking the pre-detected object bounding boxes, our SVG-net does not require any pre-trained object detector. In general, our SVG-net extracts the visual features from the video frames, and then produces the text-guided visual feature by using our newly introduced cross-modal feature learning module and finally generates an initial object tube, including the bounding boxes $\mathbf{b}_t$ for the target object in each frame and the indexes of the starting and ending frames.

Visual Feature and Textual Feature Encoding. We first use ResNet-101 [20] as the image encoder to extract the visual feature. The output from the 4th residual block is reshaped to the size of $HW \times C$ with H, W and C indicating the height, width, and the number of channels of the feature map, respectively, which is then employed as the extracted visual feature. For the k-th video clip, we stack the extracted visual features from each frame in this video clip to construct the clip feature $\mathbf{F}_k^{\text{clip}} \in R^{T \times HW \times C}$, which is then fed into our cross-modal feature learning module to produce the multi-modal visual feature.

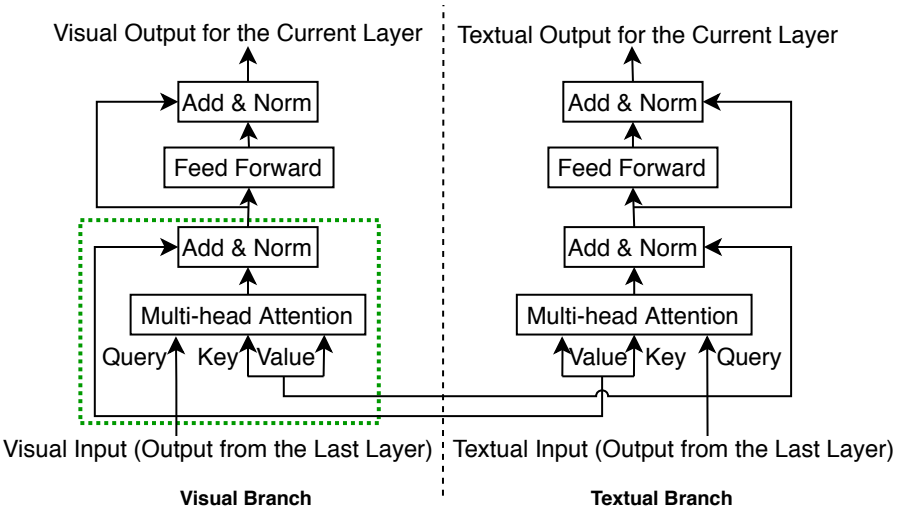
For the textual descriptions, we use a word embedding module to map each word in the description as a word vector, and each word vector is considered as one textual input token. Additionally, we add two

special tokens, ['CLS'] and ['SEP'], before and after the textual input tokens of the description to construct the complete textual input tokens $\mathbf{E} = \{e_n\}_{n=1}^{N+2}$, where e_n is the i-th textual input token.

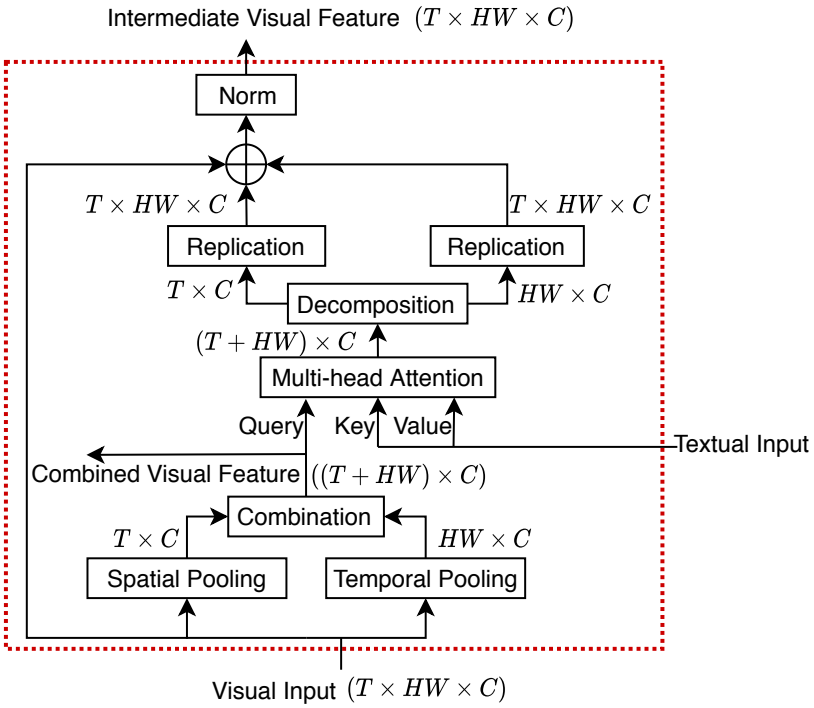
Multi-modal Feature Learning. Given the visual input feature $\mathbf{F}_k^{\text{clip}}$ and the textual input tokens $\mathbf{E}$, we develop an improved ViLBERT module called ST-ViLBERT, to learn the visual-linguistic representation. Following the structure of ViLBERT [47], our ST-ViLBERT module consists of a visual branch and a textual branch where both branches adopt the multi-layer transformer encoder [78] structure. As shown in Figure 5.2(a), the visual branch interacts with the textual branch via a set of co-attention layers, which exchanges information between the key-value pairs to generate the text-guided visual feature (or vice versa). Please refer to the work [47] for further details.

The work ViLBERT [47] takes the visual features extracted from all pre-generated proposals within an image as the visual input to learn the visual-linguistic representation (see Figure 5.2(a)). However, since these visual features are spatially pooled by using the average pooling operation, spatial information in the visual input feature space will be lost. While such information is important for bounding boxes prediction, this is not an issue for ViLBERT as it assumes the bounding boxes are generated by using the pre-trained object detectors.

Our ST-ViLBERT module extends ViLBERT for the spatial localization task without requiring any pre-generated bounding boxes, so we need to preserve spatial information when performing cross-modal feature learning. Specifically, we introduce a Spatio-temporal Combination and Decomposition (STCD) module to replace the Multi-head Attention and Add & Norm modules for the visual branch in ViLBERT. As shown in Figure 5.2(b), our STCD module respectively applies the *spatial* and *temporal* average pooling operations on the input visual feature (*i.e.*, the visual output from the last layer) to produce the initial temporal feature with the size of $T \times C$ and the initial spatial feature with the size of $HW \times C$, which are then concatenated to construct the combined visual feature with the size of $(T + HW) \times C$. We then pass the combined visual feature to the textual branch, which is used as key and value in



(a) One Co-attention Layer in ViLBERT



(b) Our Spatio-temporal Combination and Decomposition Module

Figure 5.2: (a) The overview of one co-attention layer in ViLBERT [47]. The co-attention block, consisting of a visual branch and a textual branch, generates the visual-linguistic representations by exchanging the key-value pairs for the multi-head attention blocks. (b) The structure of our Spatio-temporal Combination and Decomposition (STCD) module, which replaces the “Multi-head attention" and “Add & Norm" blocks in the visual branch of ViLBERT (marked in the green dotted box in (a)).

the multi-head attention block of the textual branch. Additionally, the combined visual feature is also fed to the multi-attention block of the visual branch together with the textual input (*i.e.*, the textual output from the last layer) to generate the initial text-guided visual feature with the size of $(T + HW) \times C$, which is then decomposed into the text-guided *temporal* feature with the size of $T \times C$ and the text-guided *spatial* feature with the size of $HW \times C$. These two features are then respectively replicated HW and T times to match the dimension of the input visual feature. The replicated features and the input visual feature are added up and normalized to generate the intermediate visual feature with the size of $T \times HW \times C$. The remaining part (including the textual branch) are the same as that in ViLBERT [47]. In our ST-ViLBERT, we take the output from the visual and textual branches in the last co-attention layer as the **t**ext-guided **v**isual feature $\mathbf{F}^{\text{tv}} \in R^{T \times HW \times C}$ and the visual-guided textual feature, respectively.

Spatial Localization Given the text-guided visual feature $\mathbf{F}^{\text{tv}}$, our SVG-net predicts the bounding box $\mathbf{b}_t$ at each frame. We first reshape $\mathbf{F}^{\text{tv}}$ to the size of $T \times H \times W \times C$. Taking the feature from each individual frame (with the size of $H \times W \times C$) as the input of three deconvolution layers, we then upsample the spatial resolution by a factor of 8. Similar as in CenterNet [104], the upsampled feature is used as the input of two parallel detection heads, with each head consisting of a 3×3 convolution layer for feature extraction and a 1×1 convolution layer for dimension reduction. The first detection head outputs a heatmap $\mathbf{A} \in R^{8H \times 8W}$, where the value at each spatial location indicates the probability of this position being the bounding box center of the target object, and the second head regresses the size (*i.e.*, the height and width) of the target bounding box at each position. In the heatmap, we select the spatial location with the highest probability as the predicted bounding box center, and use the corresponding predicted height and width at this selected position to calculate the top-left and the bottom-right coordinates of the predicted bounding box.

Initial Tube Generation Sub-module In addition to bounding box prediction, the SVG-net also predicts the temporal starting and ending

frame positions. As shown in Fig. 5.1(b), we apply spatial average pooling on the text-guided visual feature $\mathbf{F}^{\text{tv}}$ to produce the **g**lobal **t**ext-guided **v**isual feature $\mathbf{F}^{\text{gtv}} \in R^{T\times C}$ for each input video clip with T frames. We also feed the global textual feature (*i.e.*, the visual-guided textual feature corresponding to the token ['CLS']) into a MLP layer to produce the intermediate textual feature in the C-dim common feature space such that we can compute the initial relevance score for each frame between the corresponding visual feature from $\mathbf{F}^{\text{gtv}}$ and the intermediate textual feature in the common feature space by using the correlation operation. After applying the Sigmoid activation function, we obtain the final relevance score o^{obj} for each frame in one video clip, which indicates how each frame is relevant to the textual description in the common feature space. We then combine the relevance scores of all frames from all video clips to construct a sequence of relevance scores for all KT frames in the whole video and then apply a median filter operation to smooth this score sequence. After that, the positions of the starting and ending frames can be determined by using a pre-defined threshold. Namely, the bounding boxes with the smoothed relevance scores less than the threshold will be removed from the initial tube. If more than one starting and ending frame pairs are detected, we select the pair with the longest duration as our initial starting and ending frame prediction result $(\bar{t}_s^0, \bar{t}_e^0)$. The initial temporal boundaries $(\bar{t}_s^0, \bar{t}_e^0)$ and the predicted bounding boxes $\mathbf{b}_t$ form the initial tube prediction result $\mathbf{B}^{\text{init}} = \{\mathbf{b}_t\}_{t=\bar{t}_s^0}^{\bar{t}_e^0}$.

Loss Functions We use a combination of a focal loss, a L1 loss, and a cross-entropy loss to train the SVG-net. At the training stage, we randomly sample a set of video clips with T consecutive frames from each input videos, and then we select the video clips which contain at least one frame having a ground-truth bounding box as the training samples. Below we take one frame as an example to introduce the losses. Specifically, for each frame in a training video clip, we denote the center, width and height, as well as the relevance label of the ground-truth bounding box as $(\hat{x}, \hat{y})$, $\hat{w}$, $\hat{h}$, and $\hat{o}$, respectively. When the frame contains a ground-truth bounding box, we have $\hat{o} = 1$; otherwise, we have $\hat{o} = 0$. Based on the ground-truth bounding box, we follow [104] to generate a center heatmap $\hat{\mathbf{A}}$ for each frame by using the Gaussian kernel

$\hat{a}^{x,y} = \exp(-\frac{(x-\hat{x})^2+(y-\hat{y})^2}{2\sigma^2})$, where $\hat{a}^{x,y}$ is the value of $\hat{\mathbf{A}} \in R^{8H\times 8W}$ at the spatial location (x, y), and σ is the bandwidth parameter, which is adaptively determined based on the object size [28]. For training our SVG-net, the objective function L_{SVG} for each frame in the training video clip is defined as follows:

$$\begin{aligned} L_{\text{SVG}} =& \lambda_1 L_c(\mathbf{A}; \hat{\mathbf{A}}) + \lambda_2 L_{\text{rel}}(o^{\text{obj}}; \hat{o}) \\ &+ \lambda_3 L_{\text{size}}(w^{\hat{x},\hat{y}}; h^{\hat{x},\hat{y}}; \hat{w}; \hat{h}), \end{aligned} \tag{5.1}$$

where $\mathbf{A} \in R^{8H\times 8W}$ is the predicted heatmap, o^{obj} is the predicted relevance score, $w^{\hat{x},\hat{y}}$ and $h^{\hat{x},\hat{y}}$ are the predicted width and height of the bounding box centered at the position $(\hat{x},\hat{y})$. L_c is the focal loss [39] for predicting the bounding box center; L_{size} is a L1 loss for regressing the size of the bounding box; L_{rel} is a cross-entropy loss for relevance score prediction. We empirically set the loss weights $\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = 0.1$. We then average L_{SVG} over all frames in each training video clip to produce the total loss for this training video clip. Please refer to Supplementary for further details.

5.2.3 Temporal Boundary Refinement

The accurate temporal boundaries of the generated tube play an important role in the spatio-temporal video grounding task. For the temporal localization task, we observe that the anchor-based methods can often produce temporal segments covering the entire ground-truth segments, but their boundaries may not be accurate. While the frame-based methods can accurately determine boundaries in local starting/ending regions, it may falsely detect the temporal segments because overall segment-level information is missing in these methods. To this end, we develop a multi-stage ViLBERT-based Temporal Boundary Refinement network (TBR-net), consisting of an anchor-based branch and a frame-based branch, to progressively refine the temporal boundaries of the initial tube.

As shown in Fig. 5.3, at the first stage, the output from the SVG-net (i.e., $\mathbf{B}^{init}$) is used as an anchor, and its temporal boundaries are first refined by using the anchor-based branch, which takes advantage of the overall feature of the anchor to produce the refined temporal boundaries,

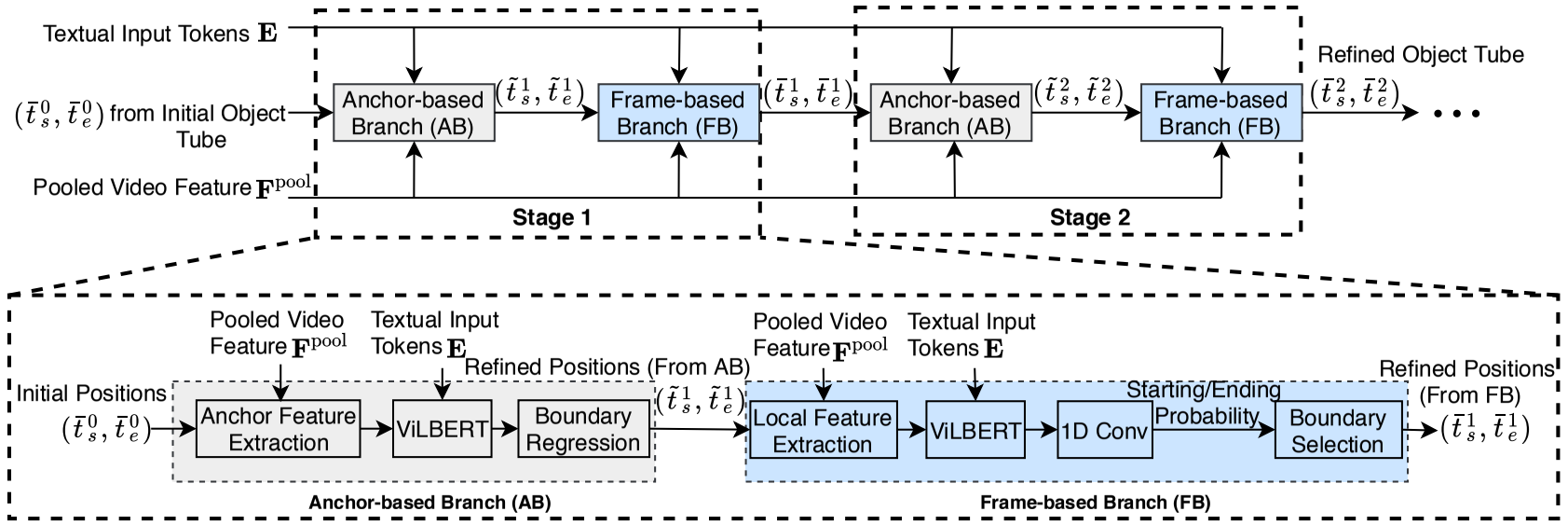


Figure 5.3: Overview of our Temporal Boundary Refinement network (TBR-net), which is a multi-stage ViLBERT-based framework with an anchor-based branch and a frame-based branch (two stages are used for better illustration). The input starting and ending frame positions at each stage are those of the output object tube from the previous stage. At stage 1, the input object tube is generated by our SVG-net.

$\tilde{t}_s$ and $\tilde{t}_e$. The frame-based branch then continues to refine the boundaries $\tilde{t}_s$ and $\tilde{t}_e$ within the local regions by leveraging the frame-level features. The improved temporal boundaries from the frame-based branch, (*i.e.*, $\bar{t}_s$ and $\bar{t}_e$), can be used as a better anchor for the anchor-based branch at the next stage. The iterative process will repeat until satisfactory results are obtained. We take the first stage of our SVG-net as an example to introduce the details of each branch, and the details in other stages are similar to those in the first stage.

Anchor-based branch For the anchor-based branch, the core idea is to take advantage of the feature extracted from the whole anchor in order to regress the target boundaries and predict the offsets.

Given an anchor with the temporal boundaries ($\bar{t}_s^0$, $\bar{t}_e^0$) and the length $l = \bar{t}_e^0 - \bar{t}_s^0$, we temporally extend the anchor before and after the starting and the ending frames by $l/4$, to include more context information. We then uniformly sample 20 frames within the temporal duration $[\bar{t}_s^0 - l/4, \bar{t}_e^0 + l/4]$ and generate the anchor feature $\tilde{\mathbf{F}}$ by stacking the corresponding features along the temporal dimension from the pooled video feature $\mathbf{F}^{\text{pool}}$ at the 20 sampled frames, where the pooled video feature $\mathbf{F}^{\text{pool}}$ is generated by stacking the spatially pooled features from all frames within each video along the temporal dimension.

The anchor feature $\tilde{\mathbf{F}}$ together with the textual input token $\mathbf{E}$ are used as the input to the co-attention based transformer module of ViLBERT to generate the text-guided anchor feature, based on which the average pooling operation is used to produce the pooled text-guided anchor feature. This feature is then fed into an MLP layer to regress the relative positions with respect to $\bar{t}_s^0$ and $\bar{t}_e^0$, and generate the new starting and ending frame positions $\tilde{t}_s^1$ and $\tilde{t}_e^1$.

Frame-based branch For the frame-based branch, we extract the feature from each frame within the starting/ending duration and predicts the possibility of this frame being a starting/ending frame. Below we take the starting position as an example. Given the refined starting position $\tilde{t}_s^1$ generated from the previous anchor-based branch, we define the starting duration as $D_s = [\tilde{t}_s^1 - D/2 + 1, \tilde{t}_s^1 + D/2]$, where D is the length

of the starting duration. We then construct the frame-level feature $\bar{\mathbf{F}}^s$ by stacking the corresponding features of all frames within the starting duration from the pooled video feature $\mathbf{F}^{\text{pool}}$. We then take the frame-level feature $\bar{\mathbf{F}}^s$ and the textual input token $\mathbf{E}$ as the input and adopt the ViLBERT module to produce the text-guided frame-level feature, which is then used as the input to a 1D convolution operation to predict the probability of each frame within the starting duration being the starting frame. We then select the frame with the highest probability as the starting frame position $\bar{t}_s^1$. Accordingly, we can produce the ending frame position $\bar{t}_e^1$. Note that the kernel size of the 1D convolution operations is 1 and the parameters related to the operations for generating the starting and ending frame positions are not shared. Once the new starting and ending frame positions $\bar{t}_s^1$ and $\bar{t}_e^1$ are generated, they can be used as the input for the anchor-based branch at the next stage.

Loss Functions To train our TBR-net, we define the training objective function L_{TBR} as follow:

$$L_{\text{TBR}} = \lambda_4 L_{\text{anc}}(\Delta\mathbf{t}; \Delta\hat{\mathbf{t}}) + \lambda_5 L_s(p^s; \hat{p}^s) + \lambda_6 L_e(p^e; \hat{p}^e), \tag{5.2}$$

where L_{anc} is the regression loss used in [14] for regressing the temporal offsets; L_s and L_e are the cross-entropy loss for predicting the starting and the ending frames, respectively. In L_{anc}, $\Delta\mathbf{t}$ and $\Delta\hat{\mathbf{t}}$ are the predicted offsets and the ground-truth offsets, respectively; In L_s, p^s is the predicted probability of being the starting frame for the frames within the *starting* duration, while $\hat{p}^s$ is the ground-truth label for *starting* frame position prediction. Similarly, we use p^e and $\hat{p}^e$ for the loss L_e. In this work, we empirically set the loss weights $\lambda_4 = \lambda_5 = \lambda_6 = 1$.

5.3 Experiment

5.3.1 Experiment Setup

Datasets We evaluate our proposed framework on the VidSTG [102] dataset and the HC-STVG [76] dataset.

-VidSTG. This dataset consists of 99,943 sentence descriptions with 44,808 declarative sentences and 55,135 interrogative sentences describing 79 types of objects appearing in the untrimmed videos. Following [102], we divide the sentence descriptions into the training set, the validation set, and the testing set with 36,202 (*resp.*, 44,482), 3,996 (*resp.*, 4,960), and 4,610 (*resp.*, 5,693) declarative (*resp.*, interrogative) sentences. The described objects in the untrimmed videos are annotated with the spatio-temporal tubes.

-HC-STVG. This dataset consists of 5,660 video-description pairs and all videos are untrimmed. This dataset is human-centric since all videos are captured in multi-person scenes and the descriptions contain rich expressions related to human attributes and actions. This dataset is divided into the training set and the testing set with 4,500 and 1,160 video-sentence pairs, respectively. All target persons are annotated with spatio-temporal tubes.

Implementation details We use the ResNet-101 [20] network pretrained on ImageNet [11] as our image encoder to extract the visual features from the RGB frames in the input videos. For the ST-ViLBERT module in our SVG-net and the ViLBERT module in our TBR-net, we employ the ViLBERT model pretained on the Conceptual Caption dataset [61] for initialization. Following [102], we sample the input videos at the frame rate of 5fps. The batch size and the initial learning rate are set to be 6 and 0.00001, respectively. After training our models (for both SVG-net and TBR-net) for 50 epochs, we decrease the learning rate by a factor of 10 and then train the models for another 10 epochs. The window size of the median filter, the pre-defined threshold θ_p, and the temporal length of each clip K are set as 9, 0.3, and 4, respectively. We apply *three* stages in the temporal boundary refinement step as the results are already saturated. Our method is implemented by using PyTorch on the machine with a single GTX 1080Ti GPU.

Evaluation metrics We follow [102] to use m_vIoU and vIoU@R as our evaluation criteria. The vIoU is calculated as $\text{vIoU} = \frac{1}{|S_U|}\sum_{t\in S_I} r_t$, where r_t is the IoU between the detected bounding box and the ground-truth bounding box at frame t, the set S_I contains the intersected frames

between the detected tubes and the ground-truth tubes (*i.e.*, the intersection set between the frames from both tubes), and S_U is the set of frames from either the detected tubes or the ground-truth tubes. The m_vIoU score is defined as the average vIoU score over all testing samples, and vIoU@R refers to the ratio of the testing samples with vIoU $>$ R over all the testing samples.

Baseline Methods

-STGRN [102] is the state-of-the-art method on the VidSTC dataset. Although this method does not require pre-generate tube proposals, it still requires the pre-trained detector to produce the bounding box proposals in each frame, which are then used to build the spatial relation graph and the temporal dynamic graph. And the final bounding boxes are *selected* from these proposals. Therefore, its performance is highly dependent on the quality of the pre-generated proposals.

-STGVT [76] is the state-of-the-art method on the HC-STVG dataset. Similar to our SVG-net, it adopts a visual-linguistic transformer module based on ViLBERT [47] to learn the cross-modal representations. However, STGVT relies on a pre-trained object detector and a linking algorithm to generate the tube proposals, while our framework does not require any pre-generated tubes.

-Vanilla (ours) In our proposed SVG-net, the key part is ST-ViLBERT. Different from the original ViLBERT [47], which does not model spatial information for the input visual features, the ST-ViLBERT in our SVG-net can preserve spatial information from the input visual features and model spatio-temporal information. To evaluate the effectiveness of the ST-ViLBERT, we introduce this baseline by replacing the ST-ViLBERT with the original ViLBERT.

-SVG-net (ours) As the first step of our proposed framework, the SVG-net can predict the initial tube. Here we treat it as a baseline method. By comparing the performance of this baseline method and our two-step framework, we can evaluate the effectiveness of our TBR-net.

5.3.2 Comparison with the State-of-the-art Methods

We compare our proposed method with the state-of-the-art methods on both VidSTG and HC-STVG datasets. The results on these two datasets are shown in Table 5.1 and Table 5.2. From the results, we observe that: 1) Our proposed method outperforms the state-of-the-art methods by a large margin on both datasets in terms of all evaluation metrics; 2) SVG-net (ours) achieves better performance when compared with the Vanilla model, indicating that it is more effective to use our ST-ViLBERT module to learn cross-modal representations than the original ViLBERT module; 3) Our proposed Temporal Boundary Refinement method can further boost the performance. When compared with the initial tube generation results from our SVG-net, we can further achieve more than 2% improvement in terms of all evaluation metrics after applying our TBR-net to refine the tmeporal boundaries.

5.3.3 Ablation Study

In this section, we take the VidSTG dataset as an example to conduct the ablation study and investigate the contributions of different components in our proposed method.

Effectiveness of the ST-ViLBERT Module As shown in Table 5.1 and Table 5.2, the video visual grounding results after using our SVG-net are much higher than our baseline method Vanilla, which demonstrates it is useful to additionally preserve spatial information when modeling spatio-temporal information in the spatial visual grounding step.

It is also interesting to observe that the performance improvement between our SVG-net and our baseline method Vanilla on the HC-STVG dataset is larger than that on the VidSTG dataset, which indicates it is more useful to employ the proposed ST-ViLBERT module on the HC-STVG dataset. A possible explanation is that the HC-STVG dataset has many complex multi-person scenes, which makes the spatial localization task more challenging than the VidSTG dataset. Without pre-generating

Table 5.1: Results of different methods on the VidSTG dataset. "*" indicates the results are quoted from its original work.

Method	Declarative Sentence Grounding			Interrogative Sentence Grounding		
	m_vIoU(%)	vIoU@0.3(%)	vIoU@0.5(%)	m_vIoU(%)	vIoU@0.3(%)	vIoU@0.5(%)
STGRN* [102]	19.75	25.77	14.60	18.32	21.10	12.83
Vanilla (ours)	21.49	27.69	15.77	20.22	23.17	13.82
SVG-net (ours)	22.87	28.31	16.31	21.57	24.09	14.33
SVG-net + TBR-net (ours)	**25.21**	**30.99**	**18.21**	**23.67**	**26.26**	**16.01**

Table 5.2: Results of different methods on the HC-STVG dataset. "*" indicates the results are quoted from the original work.

Method	m_vIoU	vIoU@0.3	vIoU@0.5
STGVT* [76]	18.15	26.81	9.48
Vanilla (ours)	17.94	25.59	8.75
SVG-net (ours)	19.45	27.78	10.12
SVG-net + TBR-net (ours)	**22.41**	**30.31**	**12.07**

the tube proposals, our ST-ViLBERT can learn a more representative cross-modal feature than the original ViLBERT because spatial information is preserved from the input visual feature in our ST-ViLBERT.

Effectiveness of the TBR-net Our temporal boundary refinement method takes advantage of complementary information between the anchor-based and frame-based temporal boundary refinement methods to progressively refine the temporal boundary locations of the initial object tubes from our SVG-net. To verify the effectiveness of combining these two lines of temporal localization methods and the design of our multi-stage structure, we conduct the experiments by using different variants for the temporal boundary refinement module. The results are reported in Table 5.3. For all methods, the initial tubes are generated by using our SVG-net, so these methods are referred to as SVG-net, SVG-net + IFB-net, SVG-net + IAB-net, and SVG-net + TBR-net, respectively.

Through the results, we have the following observations: (1) For the *iterative frame-based* method (IFB-net) and the *iterative anchor-based* method (IAB-net), the spatio-temporal video grounding performance can be slightly improved after increasing the number of stages. (2) The performance of both SVG-net + IFB-net and SVG-net + IAB-net at any stages are generally better than that of the baseline method SVG-net, indicating that either the frame-based or the anchor-based method can refine the temporal boundaries of the initial object tubes. (3) Also, the results show that the IFB-net can bring more improvement in terms of $vIoU$@0.5, while the IAB-net performs better in terms of $vIoU$@0.3. (4)

Table 5.3: Results of our method and its variants when using different number of stages on the VidSTG dataset.

Methods	#Stages	Declarative Sentence Grounding			Interrogative Sentence Grounding		
		m_vIoU	vIoU@0.3	vIoU@0.5	m_vIoU	vIoU@0.3	vIoU@0.5
SVG-net	-	22.87	28.31	16.31	21.57	24.09	14.33
SVG-net + IFB-net	1	23.41	28.84	16.69	22.12	24.48	14.57
	2	23.82	28.97	16.85	22.55	24.69	14.71
	3	23.77	28.91	16.87	22.61	24.72	14.79
SVG-net + IAB-net	1	23.27	29.57	16.39	21.78	24.93	14.31
	2	23.74	29.98	16.52	22.21	25.42	14.59
	3	23.85	30.15	16.57	22.43	25.71	14.47
SVG-net + TBR-net	1	24.32	29.92	17.22	22.69	25.37	15.29
	2	24.95	30.75	17.02	23.34	26.03	15.83
	3	**25.21**	**30.99**	**18.21**	**23.67**	**26.26**	**16.01**

Our full model SVG-net + TBR-net outperforms the two alternative methods SVG-net + IFB-net and SVG-net + IAB-net and achieves the best results in terms of all evaluation metrics, which demonstrates our TBR-net can take advantage of the merits of these two alternative methods.

5.4 Summary

In this chapter, we have proposed a novel two-step spatio-temporal video grounding framework STVGBert based on the visual-linguistic transformer to produce spatio-temporal object tubes for a given query sentence, which consists of SVG-net and TBR-net. Besides, we have introduced a novel cross-modal feature learning module ST-ViLBERT in our SVG-net. Our SVG-net can produce bounding boxes without requiring any pre-trained object detector. We have also designed a new Temporal Boundary Refinement network, which leverages the complementarity of the frame-based and the anchor-based methods to iteratively refine the temporal boundaries. Comprehensive experiments on two benchmark datasets VidSTG and HC-STVG demonstrate the effectiveness of our newly proposed framework for spatio-temporal video grounding.

Chapter 6

Conclusions and Future Work

As the increasing accessibility of personal camera devices and the rapid development of network technologies, we have witnessed an increasing number of videos on Internet. It can be foreseen that the number of videos will continually grow in the future. The learning techniques for video localization will attract more and more attentions from both academies and industries. In this thesis, we have investigated three major vision tasks for video localization, spatio-temporal action localization, temporal action localization, and spatio-temporal visual grounding, and we have also developed new algorithms for each of these problems. In this chapter, we conclude the contributions of our works and discuss the potential directions of video localization in the future.

6.1 Conclusions

The contributions of this thesis are summarized as follows,

- We have proposed the Progressive Cross-stream Cooperation (PCSC) framework to improve the spatio-temporal action localization results, in which we first use our PCSC framework for spatial localization at the frame level and then apply our temporal PCSC framework for temporal localization at the action tube level. Our PCSC framework consists of several iterative stages. At each stage, we progressively improve action localization results for one stream (*i.e.*, RGB/flow) by leveraging the information from another stream (*i.e.*, RGB/flow) at both region proposal level and feature level.

The effectiveness of our newly proposed approaches is demonstrated by extensive experiments on both UCF-101-24 and J-HMDB datasets.

- We have proposed a progressive cross-granularity cooperation (PCG-TAL) framework, to gradually improve temporal action localization performance. Our PCG-TAL framework consists of an Anchor-Frame Cooperation (AFC) module to take advantage of both frame-based and anchor-based proposal generation schemes, along with a two-stream cooperation strategy to encourage collaboration between the complementary appearance and motion clues, In our framework, the cooperation mechanisms are conducted in a progressive fashion at both feature level and segment proposal level by stacking multiple AFC modules over different stages. Comprehensive experiments and ablation studies on THUMOS14, ActivityNet v1.3, and UCF-101-24 datasets show the effectiveness of our proposed framework.
- We have proposed a novel two-step spatio-temporal video grounding framework STVGBert based on the visual-linguistic transformer to produce spatio-temporal object tubes for a given query sentence, which consists of SVG-net and TBR-net. Besides, we have introduced a novel cross-modal feature learning module ST-ViLBERT in our SVG-net. Our SVG-net can produce bounding boxes without requiring any pre-trained object detector. We have also designed a new Temporal Boundary Refinement network, which leverages the complementarity of the frame-based and the anchor-based methods to iteratively refine the temporal boundaries. Comprehensive experiments on two benchmark datasets VidSTG and HC-STVG demonstrate the effectiveness of our newly proposed framework for spatio-temporal video grounding.

6.2 Future Work

The potential directions of video localization in the future could be unifying the spatial video localization and the temporal video localization into

one framework. As described in Chapter 3, most of the existing works generate the spatial and temporal action localization results by two separate networks, respectively. It is a non-trivial task to integrate these two networks into one framework and optimize it as a whole. Similarly, while we have made progresses on handling the spatial visual grounding task and the temporal visual grounding task with our STVGBert introduced in Chapter 5, these two tasks are still performed by two separate networks. We believe that it still needs more efforts in the future to combine these two branches to generate the spatial and the temporal visual grounding results simultaneously.

Bibliography

[1] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. "Localizing moments in video with natural language". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5803–5812.

[2] A. Blum and T. Mitchell. "Combining Labeled and Unlabeled Data with Co-training". In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. COLT' 98. Madison, Wisconsin, USA: ACM, 1998, pp. 92–100.

[3] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. "Activitynet: A large-scale video benchmark for human activity understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.

[4] 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*. 2017, pp. 6299–6308.

[5] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. "Rethinking the faster r-cnn architecture for temporal action localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1130–1139.

[6] S. Chen and Y.-G. Jiang. "Semantic proposal for activity localization in videos via sentence query". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 8199–8206.

[7] Z. Chen, L. Ma, W. Luo, and K.-Y. Wong. "Weakly-Supervised Spatio-Temporally Grounding Natural Sentence in Video". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 1884–1894.

[8] G. Chéron, A. Osokin, I. Laptev, and C. Schmid. "Modeling Spatio-Temporal Human Track Structure for Action Localization". In: *CoRR* abs/1806.11008 (2018). arXiv: 1806.11008. URL: http://arxiv.org/abs/1806.11008.

[9] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen. "Temporal context network for activity localization in videos". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5793–5802.

[10] A. Dave, O. Russakovsky, and D. Ramanan. "Predictive-corrective networks for action detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 981–990.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[12] 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*. 2016, pp. 1933–1941.

[13] J. Gao, K. Chen, and R. Nevatia. "Ctap: Complementary temporal action proposal generation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 68–83.

[14] J. Gao, C. Sun, Z. Yang, and R. Nevatia. "Tall: Temporal activity localization via language query". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5267–5275.

[15] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia. "Turn tap: Temporal unit regression network for temporal action proposals". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3628–3636.

[16] R. Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[18] G. Gkioxari and J. Malik. "Finding action tubes". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[19] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. "AVA: A video dataset of spatio-temporally localized atomic visual actions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6047–6056.

[20] 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*. 2016, pp. 770–778.

[21] R. Hou, C. Chen, and M. Shah. "Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2017.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[23] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *IEEE conference on computer vision and pattern recognition (CVPR)*. Vol. 2. 2017, p. 6.

[24] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. "Towards understanding action recognition". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3192–3199.

[25] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. *THUMOS challenge: Action recognition with a large number of classes*. 2014.

[26] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. "Action tubelet detector for spatio-temporal action localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4405–4413.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[28] H. Law and J. Deng. "Cornernet: Detecting objects as paired keypoints". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 734–750.

[29] 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*. 2017, pp. 156–165.

[30] J. Lei, L. Yu, T. L. Berg, and M. Bansal. "Tvqa+: Spatio-temporal grounding for video question answering". In: *arXiv preprint arXiv:1904.11574* (2019).

[31] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei. "Recurrent tubelet proposal and recognition networks for action detection". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 303–318.

[32] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou. "Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training." In: *AAAI*. 2020, pp. 11336–11344.

[33] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. "VisualBERT: A Simple and Performant Baseline for Vision and Language". In: *Arxiv*. 2019.

[34] Y. Liao, S. Liu, G. Li, F. Wang, Y. Chen, C. Qian, and B. Li. "A Real-Time Cross-modality Correlation Filtering Method for Referring Expression Comprehension". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10880–10889.

[35] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. "BMN: Boundary-Matching Network for Temporal Action Proposal Generation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3889–3898.

[36] T. Lin, X. Zhao, and Z. Shou. "Single shot temporal action detection". In: *Proceedings of the 25th ACM international conference on Multimedia*. 2017, pp. 988–996.

[37] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. "Bsn: Boundary sensitive network for temporal action proposal generation". In:

Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 3–19.

[38] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature Pyramid Networks for Object Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[41] D. Liu, H. Zhang, F. Wu, and Z.-J. Zha. "Learning to assemble neural module tree networks for visual grounding". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4673–4682.

[42] J. Liu, L. Wang, and M.-H. Yang. "Referring expression generation and comprehension via attributes". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4856–4864.

[43] L. Liu, H. Wang, G. Li, W. Ouyang, and L. Lin. "Crowd counting using deep recurrent spatial-aware network". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press. 2018, pp. 849–855.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[45] X. Liu, Z. Wang, J. Shao, X. Wang, and H. Li. "Improving referring expression grounding with cross-modal attention-guided erasing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1950–1959.

[46] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang. "Multi-granularity Generator for Temporal Action Proposal". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3604–3613.

[47] J. Lu, D. Batra, D. Parikh, and S. Lee. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13–23.

[48] G. Luo, Y. Zhou, X. Sun, L. Cao, C. Wu, C. Deng, and R. Ji. "Multi-task Collaborative Network for Joint Referring Expression Comprehension and Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10034–10043.

[49] S. Ma, L. Sigal, and S. Sclaroff. "Learning activity progression in lstms for activity detection and early detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1942–1950.

[50] B. Ni, X. Yang, and S. Gao. "Progressively parsing interactional objects for fine grained action detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1020–1028.

[51] W. Ouyang, K. Wang, X. Zhu, and X. Wang. "Chained cascade network for object detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1938–1946.

[52] W. Ouyang and X. Wang. "Single-pedestrian detection aided by multi-pedestrian detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3198–3205.

[53] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, et al. "DeepID-Net: Object detection with deformable part based convolutional neural networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.7 (2017), pp. 1320–1334.

[54] X. Peng and C. Schmid. "Multi-region two-stream R-CNN for action detection". In: *European Conference on Computer Vision*. Springer. 2016, pp. 744–759.

[55] L. Pigou, A. van den Oord, S. Dieleman, M. V. Herreweghe, and J. Dambre. "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video". In: *International Journal of Computer Vision* 126.2-4 (2018), pp. 430–439.

[56] Z. Qiu, T. Yao, and T. Mei. "Learning spatio-temporal representation with pseudo-3d residual networks". In: *proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5533–5541.

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[58] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[59] A. Sadhu, K. Chen, and R. Nevatia. "Video Object Grounding using Semantic Roles in Language Description". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10417–10427.

[60] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin. "Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos". In: *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. 2016.

[61] P. Sharma, N. Ding, S. Goodman, and R. Soricut. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2556–2565.

[62] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5734–5743.

[63] Z. Shou, D. Wang, and S.-F. Chang. "Temporal action localization in untrimmed videos via multi-stage cnns". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1049–1058.

[64] K. Simonyan and A. Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Advances in Neural Information Processing Systems 27*. 2014, pp. 568–576.

[65] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[66] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. "A multi-stream bi-directional recurrent neural network for fine-grained action detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1961–1970.

[67] G. Singh, S. Saha, M. Sapienza, P. H. Torr, and F. Cuzzolin. "Online real-time multiple spatiotemporal action localisation and prediction". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3637–3646.

[68] K. Soomro, A. R. Zamir, and M. Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[69] R. Su, W. Ouyang, L. Zhou, and D. Xu. "Improving Action Localization by Progressive Cross-stream Cooperation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12016–12025.

[70] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. "VL-BERT: Pre-training of Generic Visual-Linguistic Representations". In: *International Conference on Learning Representations*. 2020.

[71] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. "Videobert: A joint model for video and language representation learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7464–7473.

[72] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. "Actor-centric Relation Network". In: *The European Conference on Computer Vision (ECCV)*. 2018.

[73] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang. "Fishnet: A versatile backbone for image, region, and pixel level prediction". In: *Advances in Neural Information Processing Systems*. 2018, pp. 762–772.

[74] 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*. 2015, pp. 1–9.

[75] H. Tan and M. Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019.

[76] Z. Tang, Y. Liao, S. Liu, G. Li, X. Jin, H. Jiang, Q. Yu, and D. Xu. "Human-centric Spatio-Temporal Video Grounding With Visual Transformers". In: *arXiv preprint arXiv:2011.05049* (2020).

[77] 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*. 2015, pp. 4489–4497.

[78] 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*. 2017, pp. 5998–6008.

[79] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. "Actionness Estimation Using Hybrid Fully Convolutional Networks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2708–2717.

[80] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. "Untrimmednets for weakly supervised action recognition and detection". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 4325–4334.

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

[82] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel. "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks". In: *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition. 2019, pp. 1960–1968.

[83] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. "Learning to track for spatio-temporal action localization". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3164–3172.

[84] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 305–321.

[85] H. Xu, A. Das, and K. Saenko. "R-c3d: Region convolutional 3d network for temporal activity detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5783–5792.

[86] M. Yamaguchi, K. Saito, Y. Ushiku, and T. Harada. "Spatio-temporal person retrieval via natural language queries". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1453–1462.

[87] S. Yang, G. Li, and Y. Yu. "Cross-modal relationship inference for grounding referring expressions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4145–4154.

[88] S. Yang, G. Li, and Y. Yu. "Dynamic graph attention for referring expression comprehension". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4644–4653.

[89] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. S. Davis, and J. Kautz. "STEP: Spatio-Temporal Progressive Learning for Video Action Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[90] X. Yang, X. Liu, M. Jian, X. Gao, and M. Wang. "Weakly-Supervised Video Object Grounding by Exploring Spatio-Temporal Contexts". In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 1939–1947.

[91] Z. Yang, T. Chen, L. Wang, and J. Luo. "Improving One-stage Visual Grounding by Recursive Sub-query Construction". In: *arXiv preprint arXiv:2008.01059* (2020).

[92] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo. "A fast and accurate one-stage approach to visual grounding". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4683–4693.

[93] Z. Yang, J. Gao, and R. Nevatia. "Spatio-temporal action detection with cascade proposal and location anticipation". In: *arXiv preprint arXiv:1708.00042* (2017).

[94] Y. Ye, X. Yang, and Y. Tian. "Discovering spatio-temporal action tubes". In: *Journal of Visual Communication and Image Representation* 58 (2019), pp. 515–524.

[95] 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*. 2016, pp. 2678–2687.

[96] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. "Mattnet: Modular attention network for referring expression comprehension". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1307–1315.

[97] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. "Temporal action localization with pyramid of score distribution features". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3093–3102.

[98] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng. "Temporal action localization by structured maximal sums". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3684–3692.

[99] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan. "Graph Convolutional Networks for Temporal Action Localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7094–7103.

[100] W. Zhang, W. Ouyang, W. Li, and D. Xu. "Collaborative and Adversarial Network for Unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3801–3809.

[101] Z. Zhang, Z. Zhao, Z. Lin, B. Huai, and J. Yuan. "Object-Aware Multi-Branch Relation Networks for Spatio-Temporal Video Grounding". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 2020, pp. 1069–1075.

[102] Z. Zhang, Z. Zhao, Y. Zhao, Q. Wang, H. Liu, and L. Gao. "Where Does It Exist: Spatio-Temporal Video Grounding for Multi-Form Sentences". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10668–10677.

[103] 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*. 2017, pp. 2914–2923.

[104] X. Zhou, D. Wang, and P. Krähenbühl. "Objects as Points". In: *arXiv preprint arXiv:1904.07850*. 2019.

[105] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox. "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2904–2913.