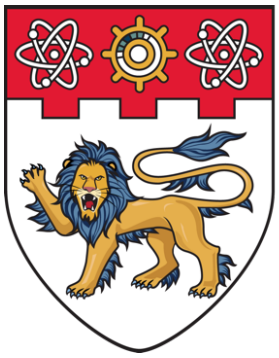


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Towards Temporal Sentence Grounding in Videos

Zhang Hao

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

25/07/2022

.....

Date

ZHANG HAO

.....

Zhang Hao

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

25/07/2022

.....

Date

.....

A/Prof. Aixin SUN

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journals / from papers accepted at conferences and and 2 under-reviewed manuscripts in which I am listed as an author.

Chapter 2 is submitted as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. The Elements of Temporal Sentence Grounding in Videos: A Survey and Future Directions. Under review.

The contributions of the co-authors are as follows:

- Prof. Sun suggested the research direction and the general framework.
- I collected the papers, categorized and analyzed existing work, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised and proofread the manuscript.

Chapter 3 is published as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based Localizing Network for Natural Language Video Localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6543–6554, Online, 2020.

The contributions of the co-authors are as follows:

- Prof. Sun provided the initial research direction. Then we discussed about several model designs at the early stage.
- I came up with the idea, designed and performed all experiments, conducted data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised and proofread the manuscript.

Chapter 4 is accepted as Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural Language Video Localization:

A Revisit in Span-based Question Answering Framework. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2021.3060449, 2021.

The contributions of the co-authors are as follows:

- Prof. Sun suggested the research direction.
- I came up with the idea, designed and performed all experiments, conducted data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised the manuscript.
- Dr. Zhen and Goh proofread the manuscript.

Chapter 5 is published as Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Siow Mong Rick Goh. Parallel Attention Network with Sequence Matching for Video Grounding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (Long Papers)*, pages: 776–790, Online, 2021.

The contributions of the co-authors are as follows:

- Prof. Sun suggested the research direction.
- I came up with the idea, designed and performed all the experiments, conducted the data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised the manuscript.
- Dr. Zhen and Goh proofread the manuscript.

Chapter 6 is submitted as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Towards Debiasing Temporal Sentence Grounding in Video. Under review.

The contributions of the co-authors are as follows:

- Prof. Sun suggested the research direction.

- I came up with the idea, designed and performed all experiments, conducted data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised and proofread the manuscript.

Chapter 7 is published as Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Video Corpus Moment Retrieval with Contrastive Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (Long Papers)*, pages: 685–695, Online, 2021.

The contributions of the co-authors are as follows:

- Prof. Sun suggested the research direction.
- I came up with the idea, designed and performed all experiments, conducted data analysis, wrote the first manuscript, and revised the drafts.
- Prof. Sun, Dr. Jing and Zhou revised the manuscript.
- Dr. Nan, Zhen and Goh proofread the manuscript.

25/07/2022
.....
Date

.....
Zhang Hao

Acknowledgements

First and foremost, I would like to give my special gratitude to my supervisor Prof. Sun Aixin for his guidance, support and encouragement throughout my Ph.D. study, which is one of the most meaningful periods in my life. Besides being a supervisor, Prof. Sun is also a friendly and great leader. Working under his supervision was a fruitful and enjoyable experience, which allowed me to not just gain substantial knowledge about my research topic, but also broaden my perspective to related fields.

I would also like to thank my co-supervisor Dr. Zhou Joey Tianyi for his support and insightful advice. Without his sharing of knowledge and experience, it would be difficult for me to complete the research work favourably. I am thankful to my colleagues, Dr. Goh Rick Siow Mong, Dr. Liu Ping, Dr. Jing Wei, Dr. Yan Ming, Dr. Zhen Liangli, and Dr. Zhu Hongyuan for their support and help.

I am thankful to my seniors, Dr. Chen Zhe, Dr. Jin Di, Dr. Nan Guoshun, Dr. Niu Yulei for their sharing of knowledge and valuable comments; and my collaborators and lab-mates (in chronological order), Dong Kuicai, Lin Ting, Ji Yitong, Ma Yubo, Puang En Yen, Toh Wei Qi, Wang Jing, Wang Tianying, Xue Fuzhao, Wu Yin, Yu Sicheng, Zhou Shaowen, for their companionship in the journey of pursuing knowledge and expanding the boundaries of our understanding. The staffs in the graduate student office, especially Ms. Ker Shen Hui, helped me a lot in various of affairs, and I am very grateful to them.

Last but not least, I would like to thank my parents for their unconditional love and encouragement. This thesis is dedicated to them who are the motivation for my hard work.

Abstract

Temporal sentence grounding in videos (TSGV), *a.k.a.,* natural language video localization (NLVL) or video moment retrieval (VMR), aims to retrieve a temporal moment (*i.e.,* a fraction of a video) that semantically corresponds to a language query from an untrimmed video. Connecting computer vision and natural language, TSGV has drawn significant attention from researchers in both communities. Successful retrieval of temporal moments enables machines to understand and organize multimodal information in a systematic manner. Different from humans who can quickly identify temporal moments, which is semantically related to a given language query, using their inference-making ability and commonsense knowledge, machines do not have such intelligence. The main challenge is that machines require to understand the semantics of both video and language query, as well as the precise cross-modal reasoning between them. As video and language query are different modalities, the recognition and localization of temporal moments greatly depend on machine understanding of the input contents and interactions between them.

In this thesis, we introduce several novel approaches to tackle the TSGV problem from a new perspective. First, we propose to formulate TSGV as a span-based question answering (QA) task by treating the input video as a text passage. Then we devise a video span localizing network (VSLNet), on top of a typical span-based QA framework, to address TSGV by considering the differences between TSGV and span-based QA. The proposed method demonstrates that adopting a span-based QA framework is a promising direction to solve TSGV, and superior performance is obtained on several benchmark datasets.

Second, despite the promising performance achieved by VSLNet, we observe existing solutions, including VSLNet, only perform well on short videos, but fail to generalize on long videos. To address the issue of performance degradation on long videos, we extend VSLNet to VSLNet-L by applying a multi-scale split-and-concatenation strategy. VSLNet-L splits the untrimmed video into short clip segments and predicts which clip segment contains the target moment and suppresses the importance of other segments. Experimental results show that VSLNet-L well addresses the issue of performance degradation on long videos.

Third, when evaluation metric becomes strict, the results of TSGV methods drop significantly. That is, the predicted moment boundaries cannot well fit the ground truth. Based on VSLNet, we investigate a sequence matching approach, which incorporates the concepts of named entity recognition (NER) to remedy moment boundary prediction errors. We first analyze the relationships between TSGV and NER and reveal that the moment boundary prediction of TSGV is a generalized entity boundary detection problem. This insight leads us to equip a NER-style boundary detection module and develop a more effective and efficient TSGV algorithm.

Fourth, we analyze the annotation distributional bias in widely used datasets for TSGV. Existence of such bias "hints" a model to capture the statistical regularities of moment annotations. To address this issue, we propose two debiasing strategies, *i.e.,* data debiasing and model debiasing, on top of VSLNet to "force" a TSGV model to focus on cross-modal reasoning for precise moment retrieval. Experimental results show that both strategies are effective in improving model generalization capability and suppressing the effects of bias.

Finally, we study the video corpus moment retrieval (VCMR) task, which aims to retrieve a temporal moment from a collection of untrimmed and unsegmented videos. VCMR is an extension of the TSGV task, but it is more practical since VCMR does not hold the strict hypothesis that a video-query pair must be given. In this task, we first study the characteristics of two general frameworks for VCMR, where one framework is of high efficiency but inferior retrieval performance, while the other is of better performance but low efficiency. We then propose a retrieval and localization network with contrastive learning to remedy the contradiction between the efficiency and accuracy of existing approaches.

All in all, despite TSGV having been established and investigated for years, this thesis contributes several key ideas to solve TSGV from different perspectives, *i.e.,* from the view of span-based QA and NER in NLP. Besides, we propose to address the annotation distributional bias of TSGV and extend it to a more practical scenario. Meanwhile, we also shed light on a few potential directions for future work.

Contents

List of Figures

List of Tables

Acronyms

AVEL	Audio-visual event localization
CV	Computer vision
CNN	Convolutional neural network
FFN	Feed-forward neural network
GRU	Gated recurrent unit
iid	Independent and identical distribution
IoU	Intersection over union
IVR	Image-to-video Retrieval
LSTM	Long-short term memory
mIoU	Mean intersection over union
MPQA	Multi-paragraph question answering
NER	Named entity recognition
NLP	Natural language processing
NLVL	Natural language video localization
ood	Out of distribution
QA	Question answering
RNN	Recurrent neural network
STSGV	Spatio-temporal sentence grounding in videos
TSGV	Temporal sentence grounding in video
VCMR	Video corpus moment retrieval
VMR	Video moment retrieval
VRL	Video re-localization

Chapter 1

Introduction

1.1 Motivation

Designing a system that can mimic complex human behaviors such as understanding human languages and perceiving the surroundings (*e.g.,* human vision simulation) was conceived long before the development of computers. Prior research efforts mainly focus on solving computer vision and natural language processing problems separately. Examples are image classification [1, 2], object detection [3–5], action recognition [6, 7] and semantic segmentation [8, 9] tasks in computer vision (CV) community, and language modeling [10–12], machine translation [13, 14], question answering [15, 16] and natural language understanding [17, 18] tasks in natural language processing (NLP) community. With the fast development and innovation in communication and media creation technologies, video has gradually become a major type of information transmission media, and connections between vision (*e.g.,* images and videos) and language (*e.g.,* text) are more stronger. A video is formed from a sequence of continuous image frames possibly accompanied by audio and subtitle. Compared to the image and text, video conveys richer semantic knowledge, as well as more diverse and complex activities. Despite the strengths of video, searching for content from the video is challenging. Thus, there is a high demand for techniques that could quickly retrieve video segments of user interest, specified in natural language. Recently, many research work [19–23] explore jointly modeling both vision and language information with a unified framework and aim to solve vision-and-language tasks, which involves both computer vision and natural language inputs. The temporal sentence grounding in videos (TSGV)[1] is one of the fundamental and challenging problems in the vision-and-language understanding research area.

[1]The temporal sentence grounding in videos (TSGV) is also known as natural language video localization (NLVL) or video moment retrieval (VMR). In this thesis, we use the term "TSGV" to represent this task.

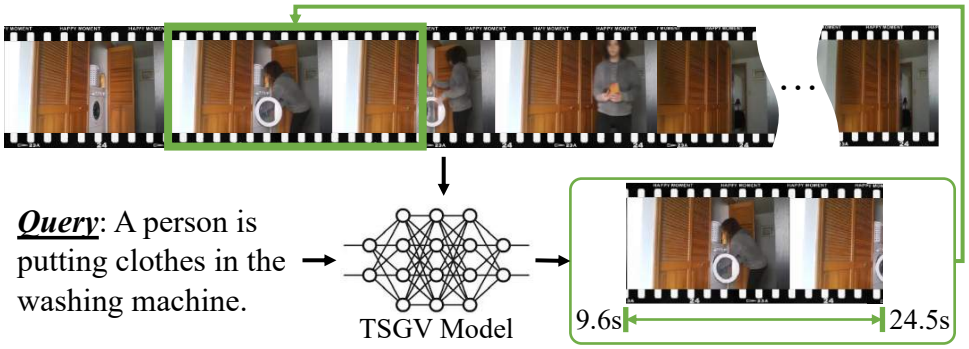


FIGURE 1.1: An illustration of temporal sentence grounding in videos (TSGV).

Given an untrimmed video, TSGV is to retrieve a video segment, also known as a temporal moment, that semantically corresponds to a query in natural language, *i.e.,* sentence. As illustrated in Figure 1.1, for query "*A person is putting clothes in the washing machine.*", TSGV needs to return the start and end timestamps (*i.e.,* $9.6s$ and $24.5s$) of a video moment from the input video as the answer, where the answer moment should contain the actions or events described by the language query. As a canonical vision-and-language task, TSGV shares similarities with some classic tasks in both CV and NLP. For instance, video action recognition [6, 7] in CV is to detect video segments, which perform specific actions, from video. Although video action recognition localizes temporal segments with activity information, it is constrained by the predefined action categories. In contrast, TSGV is more flexible and aims to retrieve complicated and diverse activities from video via arbitrary language queries. Meanwhile, TSGV is similar to reading comprehension/question answering task [15, 16] in NLP, which is to retrieve a span of words from a text passage to answer a given question. The core of reading comprehension is the interaction between text passage and query, while TSGV models the interaction between two different modalities, making it more arduous and challenging.

Why the task is essential? TSGV serves as an intermediate step for a few downstream vision-and-language tasks, such as video question answering and video-grounded dialogue. These tasks require localizing relevant moments about questions, then discovering or generating the answer to the input question by analyzing the retrieved moments. An accurate moment localization process could significantly improve answer prediction or generation for the downstream vision-and-language tasks. Naturally, TSGV connects CV and NLP communities and benefits from advancements made in both areas.

Moreover, as a fundamental vision-and-language task, TSGV has multiple real-world application scenarios. For instance, it can be applied to intelligent video surveillance services. With the help of TSGV techniques, admin or public security can quickly retrieve the moment

of interest, *e.g.,* dangerous behavior, from a long surveillance video by simply typing a language search query. Also, it can be used to support intelligent video creation. A general video creation procedure contains: writing a storyline; selecting appropriate materials; resizing/cutting the materials as segments; and integrating segments into a video. Although, there are already several video editing tools, selecting materials and resizing them as segments remain time-consuming and labor-intensive. TSGV helps to facilitate these processes.

What are the challenges? TSGV contributes to an ultimate goal which is to help machines to mimic complex human behaviors from both vision and language perspectives. Different from humans who can use their prior knowledge to quickly understand and align the contents and semantics between video and natural language, machines do not have such knowledge.

As illustrated in Figure 1.1, the video is taken under the same scene, where a person performs different actions over time. Given a query "*A person is putting clothes in the washing machine.*", it contains four key components "*person*", "*putting*", "*clothes*" and "*washing machine*". To achieve TSGV, the system requires capturing the related objects/actions in the video and maintaining their relations in the video, as well as linking those concepts to the corresponding context in the language query. At the same time, it also needs to suppress the effects of the background. Moreover, the system has to filter out similar but irrelevant content in the video to avoid false predictions. Thus, TSGV is not simply a retrieval task; it further requires the semantic understanding of both video and language contents as well as the fine-grained multimodal interactions between video and language.

The video and language query are from different modalities, and their corresponding feature learning/extraction methods are varying a lot. The video is usually encoded by 3D-based ConvNets [24, 25], while the language query is encoded by pre-trained word embeddings [26] or language models [10]. The different data nature and feature extraction procedure lead to significant discrepancies between the video and language query in the feature space. Thus, how jointly modeling the uncorrelated modalities and conducting cross-modal reasoning between them are challenging. Meanwhile, the target moment usually occupies a small portion of the video, making the positive and negative frames extremely imbalanced. The prevalent solutions [27–32] primarily treat TSGV as a ranking task, and solve it with a propose-and-rank pipeline. They mainly rely on dense proposal generation and multimodal matching with language query for moment candidate selection. These methods do not well consider the global video information as each proposal interacts with language query separately, which leads to severe information loss. Densely sampling proposal candidates also hinders the flexibility of TSGV models. Although an additional regression head is applied to adjust the location offset

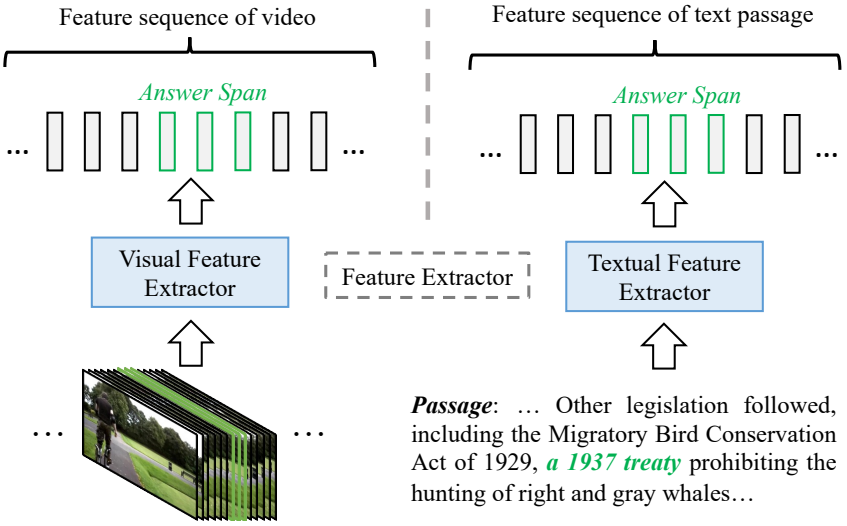


FIGURE 1.2: The comparison of feature processing between input video of TSGV and text passage of span-based QA.

of proposal candidates, the proposal-level representations are not suitable for additional regression tasks. As a result, such methods neglect the moment boundary discrimination as well as the sparsity issue in moment boundary prediction. Note the sparsity issue is caused by densely sampling proposal candidates on video with a large overlap ratio, which is equivalent to repeating the video multiple times depending on the overlap ratio, leading to the numbers of positive and negative samples being even imbalanced. Due to the repeated sampling, TSGV models need to handle more content, causing an inferior efficiency issue. Moreover, the procedure of these methods also deviates from the human perception mechanism, where human generally retrieves the moment of interest by watching the video from start to end.

1.2 Approaches

In this thesis, we solve TSGV from a different perspective by incorporating several concepts from the NLP tasks. The key theme of our approach is to formulate the TSGV into a span-based question answering (QA) problem and utilize the QA-style framework to address this task.

The essence of TSGV is to search for a video moment as the answer to a given language query from an untrimmed video. Although both TSGV and span-based QA have the language query input, they are different tasks intuitively due to the different subjects to be queried. However, both the video and text passage need to be converted into feature representations when tackling the corresponding tasks. As illustrated in Figure 1.2, the video in TSGV is encoded into a video feature sequence using visual feature extractor (*e.g.,* C3D [24] or I3D [25]), while the text passage in span-based QA is encoded into a word feature sequence via textual feature

extractor, such as GloVe [26] or BERT [10]. Accordingly, the temporal moment becomes a subsequence of visual features in TSGV, and the same for the answer of span-based QA. In this sense, TSGV and span-based QA have the same input and output formats at the feature space level, regardless of the differences in their modalities. Thus, by treating the video as a text passage, and the target moment as the answer span, TSGV shares significant similarities with the span-based QA task. Thus, it is available to solve TSGV from the perspective of span-based QA. Meanwhile, the solutions to span-based QA generally learn to extract the answer span in an end-to-end manner, which do not require pre-segmenting the paragraph into multiple overlapped sub-paragraphs. In other words, the paragraph is interacted with the question as a whole, so there is no issue of information loss and this process does not affect efficiency. Besides, solutions to span-based QA basically encode paragraphs and interact with questions sequentially; this process also well fits the human perception mechanism. Based on these reasons, we believe it is a good choice to formulate TSGV as a span-based QA task.

By formulating TSGV as a span-based QA problem, we attempt to solve this task with a multimodal span-based QA approach. Specifically, we apply a standard span-based QA framework, named VSLBase, to solve the TSGV task, where the visual features are analogous to that of text passage; the target moment is regarded as the answer span. VSLBase is trained to predict the start and end boundaries of the answer span directly. Despite the similarities, there are two main differences between span-based QA and TSGV tasks. First, their data nature is different. Video is continuous with continuous causal relation inference between two consecutive video events; while natural language is discrete and words in a sentence demonstrate syntactic structure. Second, small shifts in video frames are less imperceptible to humans than words in a text. By considering these differences, we propose a video span localizing network (VSLNet) on top of VSLBase by introducing a query-guided highlighting (QGH) strategy. In QGH, we consider a region that covers the target moment by extending its starting and ending frames a bit further as foreground, while the rest is treated as background. Through QGH, VSLNet well addresses the two differences. First, the longer region provides additional contexts for locating answer span due to the continuous nature of video content. Second, the highlighted region helps the network to focus on subtle differences between video frames, because the search space is reduced compared to the full video.

We also observe that the performance of many TSGV methods degrades significantly along with the increase in video length. Due to the fact that TSGV models generally perform well on short videos, we propose to solve this issue by splitting a long video into multiple short clip segments, where each clip segment is regarded as a short video. Similarly, by treating a long

video as a document, and a clip segment as a paragraph, TSGV can be viewed as the multi-paragraph question answering (MPQA) task [33]. The target moment in a long video can be considered as the answer span in a document for a given query. However, how properly splitting the long video into clip segments is challenging. Paragraphs in a document are semantically coherent units with boundaries defined by humans. Videos are continuous, and splitting the video into semantically coherent clip segments is difficult. In addition, the answer span in MPQA can be found in one of the paragraphs, but we cannot expect the target moment is within a single clip segment. To this end, we propose a multiscale split-and-concatenation strategy to partition the long video into clips of different lengths. Compared with fixed-length splitting, the multiscale splitting strategy increases the chance of locating a target moment in one segment. Thus, even if a target moment is truncated at one or several scales, segments in other scales may still be able to fully contain it. In this way, we extend VSLNet to VSLNet-L to locate the moment in the clips that are more likely to contain it.

Next, we study the moment boundary prediction issue in TSGV. In practice, the target moment is usually a very small portion of the video, making positive (frames in target moment) and negative (frames not in target moment) samples imbalanced. Further, we aim to predict the exact start/end boundaries of the target moment. If we view from the space of frames, sparsity is a major concern, *e.g.,* catching two frames among thousands. To mitigate this issue, we emphasize the "sequence" nature of frames and adopt the concept of named entity recognition (NER) in NLP to TSGV. Recall that TSGV is to retrieve a sequence of frames with start/end boundaries of the target moment from the video. This is analogous to extract a multi-word named entity from a sentence. The main difference is that words are discrete, so word annotations in sentence are discrete, while the video is continuous and the changes between consecutive frames are smooth. Thus, we relax the annotations on video sequence by specifying video regions, instead of frames. Specifically, we label Begin, Inside, End, and Outside regions on the video, and propose a parallel attention network with sequence matching (SeqPAN) to solve TSGV.

Besides, the commonly used benchmark datasets contain substantial distributional bias, *i.e.,* many annotated moments locate in several specific regions in videos. Consequently, many TSGV models rely on exploiting such statistical regularities of annotation distribution for moment retrieval, rather than the correct cross-modal reasoning, to achieve good performance. To alleviate the bias issue, we propose two model-agnostic debiasing strategies, data debiasing and model debiasing. Specifically, the data debiasing strategy aims to re-balance the moment annotations via data augmentation, *i.e.,* creating more samples through video truncation. Model debiasing strategy aims to explicitly capture the bias via two unimodal modules, and disentangle bias from the TSGV model by adjusting losses to compensate for biases dynamically.

Finally, we explore an extension of the TSGV task, video corpus moment retrieval (VCMR). Instead of localizing temporal moment from the single video, VCMR aims to retrieve a temporal moment from a collection of untrimmed videos. There are two common frameworks for VCMR. One is the unimodal encoding framework, which encodes video and text separately, and learns the matching through late feature fusion. Another is the cross-modal interaction framework, which takes in a video as a sequence of visual features, and the query as a sequence of word features to learn their interactions. The unimodal encoding framework generally leads to higher efficiency but inferior retrieval accuracy, while another framework has better retrieval accuracy but lower efficiency. Based on the observations, we propose to remedy the contradiction between retrieval accuracy and efficiency of existing VCMR methods via contrastive learning.

1.3 Research Contributions

We summarize our key contributions as follows:

- First, we investigate the idea to formulate TSGV as a span-based QA problem. We have shown that TSGV is conceptually similar to span-based QA, and adopt a standard span-based QA framework to solve TSGV. We further analyze the differences between them and propose a video span localizing network (VSLNet) to mitigate these differences. The experimental results demonstrate that formulating TSGV as span-based QA is a promising direction as well as the superior performance of VSLNet over other methods. This work is reported in the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) [34].

- Second, we study the issue of performance degradation among existing TSGV solutions on long videos. We propose to incorporate the concepts of multi-paragraph QA to address it and devise a multi-scale split-and-concatenation strategy to simulate the multi-paragraph setting in the video. The experiment shows that the proposed method effectively mitigates the performance degradation issue on long videos while maintaining comparable performance on short videos. This work is published in the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) [35].

- Third, we investigate a sequence-matching approach for TSGV. We study the sparsity issue of moment boundary prediction in TSGV. Inspired by the NER task and its labels for named entities. We formulate TSGV as NER, which relaxes the boundary annotation to regions, instead of two frames. We then propose a parallel attention network

with sequence matching (SeqPAN). In our evaluation, SeqPAN yields more precise moment localization results and achieves state-of-the-art performance on multiple benchmark datasets. This work is reported in the Findings of the Association for Computational Linguistics: ACL-IJCNLP [36].

- Forth, we analyze the bias in TSGV benchmark datasets. There are substantial distributional biases that exist in TSGV datasets, where many annotated moments locate in several specific regions in the video. Existing models usually rely on exploiting statistical regularities to achieve good performance. We propose two model-agnostic debiasing strategies, data debiasing, and model debiasing, to mitigate the effect of bias by forcing the model to focus on correctly cross-modal interactions. This work is under review [37].

- Finally, we explore an extension of TSGV named video corpus moment retrieval (VCMR). We study two commonly used frameworks for VCMR and reveal the contradiction between retrieval accuracy and efficiency among the two frameworks. Based on the observations, we propose to remedy the contradiction issue through contrastive learning. The experiment shows that the proposed method well balances the efficiency and retrieval accuracy. This work is reported in the Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) [38].

This thesis also provides a summary of fundamental concepts in TSGV and current research status, as well as future research directions. As the background, we present a common structure of functional components in TSGV, in a tutorial style. Then we construct a taxonomy of TSGV techniques and elaborate methods in different categories with their strengths and weaknesses. We also discuss issues with the current TSGV research and share our insights about promising research directions. This work is under review [39].

1.4 Thesis Outline

This thesis contains the introduction (this chapter), a literature review, five main contributions, and a conclusion. In Chapter 2, we provide the readers with a common structure of functional components in TSGV in a tutorial style and a review of related work in TSGV. Chapter 3 starts to investigate the TSGV problem. In this chapter, we propose to formulate the TSGV problem as a span-based QA task and devise VSLNet based on a standard span-based QA framework to address TSGV. Chapter 4 investigates the performance degradation of existing TSGV solutions on long videos and proposes to incorporate the concepts of multi-paragraph QA to mitigate the such issue by applying a multi-scale split-and-concatenation strategy. Chapter 5 studies a

sequence-matching approach in which we apply the concepts of named entity recognition to remedy the moment boundary prediction errors in TSGV. Chapter 6 analyzes the annotation distributional bias in the commonly used benchmark datasets for TSGV. We then propose two simple yet effective debiasing strategies, *i.e.,* data debiasing and model debiasing, to alleviate the bias issue. Next, Chapter 7 explores an extension of TSGV task termed video corpus moment retrieval. In this task, we propose to remedy the contradiction between retrieval efficiency and retrieval accuracy of existing approaches via contrastive learning. Finally, Chapter 8 concludes the thesis and discusses several potential directions for future work.

Chapter 2

Literature Review

In this chapter[1], we present an overview of existing approaches for temporal sentence grounding in videos as well as commonly used benchmark datasets and measures. To elaborate on what a TSGV model looks like generally, in Section 2.1, we first introduce the background of TSGV by presenting a common structure of functional components in TSGV, in a tutorial style: from feature extraction from raw video and language query, to answer prediction of target moment. We also review techniques for multimodal understanding and interaction, which is the key focus of TSGV for effective alignment between the two modalities. In Section 2.2, we categorize existing TSGV approaches into different groups according to the models' architectures and learning algorithms, and construct a taxonomy of TSGV approaches to reveal the hierarchical relationships among different categories. We then provide a literature survey about the TSGV methods in different categories with a consideration of their strengths and weaknesses. Finally, we brief the commonly used benchmark datasets and measures of TSGV in Section 2.3.

2.1 TSGV Background

As is illustrated in Figure 2.1, general pipeline architecture for TSGV consists of six components, where the dotted line in the figure indicates that the proposal generator is an optional component, and it may be placed at different stages. To be specific, a TSGV method takes a video-query pair as input, where the video is a collection of consecutive image frames, and the query is a sequence of words. The preprocessor prepares inputs for feature extraction, *e.g.,* downsampling and resizing image frames in the video, and tokenizing words in the query. Feature extractor converts video frames and query words into their corresponding vector feature representations. Then the encoder module maps video and query features to the same dimension

[1]This chapter is submitted as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "The Elements of Temporal Sentence Grounding in Videos: A Survey and Future Directions". Under review [39].

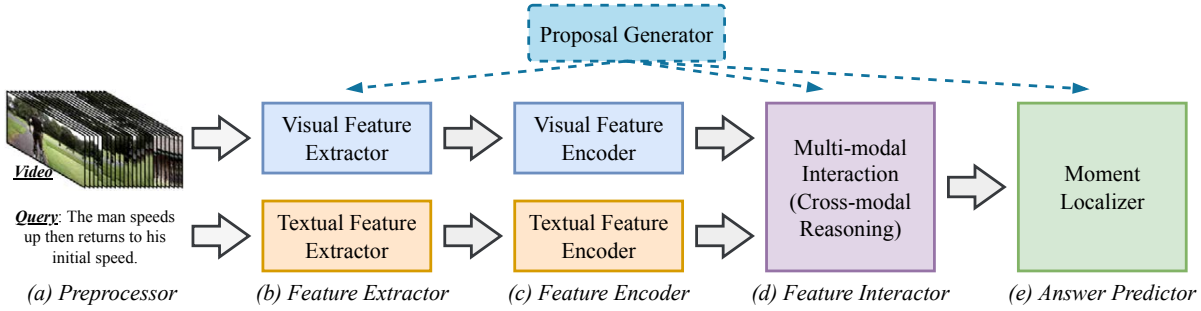


FIGURE 2.1: A general pipeline for temporal sentence grounding in videos.

and aggregates contextual information to enhance the representations. The interactor module, an essential component in TSGV, learns the multimodal representations by modeling the cross-modal interaction between video and query. Finally, the answer predictor generates moment predictions based on the learned multimodal representations. According to whether the proposal generator is applied, TSGV models can be roughly categorized into proposal-based and proposal-free methods. For proposal-based methods, the answer predictor makes predictions based on proposals generated by the proposal generator. A proposal (a video segment sampled from input video) can be considered as a candidate answer moment, which can be generated at different stages. Proposal-free methods predict answers directly without the need of generating proposals.

To better elaborate details of each TSGV component, we define the following notations. Given a TSGV dataset, we denote its video corpus as $\mathcal{V} = \{V^1, V^2, \ldots, V^N\}$ and its query set as $\mathcal{Q} = \{Q^1, Q^2, \ldots, Q^M\}$, where N and M are the number of videos and queries, respectively. Note multiple queries can be posed to the same video with its different moments as answers; typically $M \geq N$ in TSGV datasets. Given a video-query pair, a video V contains T frames $V = [f_1, f_2, \ldots, f_T]$ and a query Q has m words $Q = [q_1, q_2, \ldots, q_m]$, the start and end timestamps of ground-truth moment are denoted by τ_s and τ_e, $1 \leq \tau_s < \tau_e \leq T$. Here, we use the frame index to represent time points, based on a fixed frame rate or fps. Mathematically, TSGV is to retrieve the target moment starting from τ_s and ending at τ_e by giving a video V and query Q:

$$\mathcal{F}_{TSGV} : (V, Q) \mapsto (\tau_s, \tau_e). \tag{2.1}$$

2.1.1 Preprocessor

Video is a series of still images and the number of frames can be very large. For instance, a 2-minute video with 20 fps has $2,400$ frames in total. Thus, it is infeasible (and often unnecessary) to process every frame in a video due to computational cost. On the other hand, video

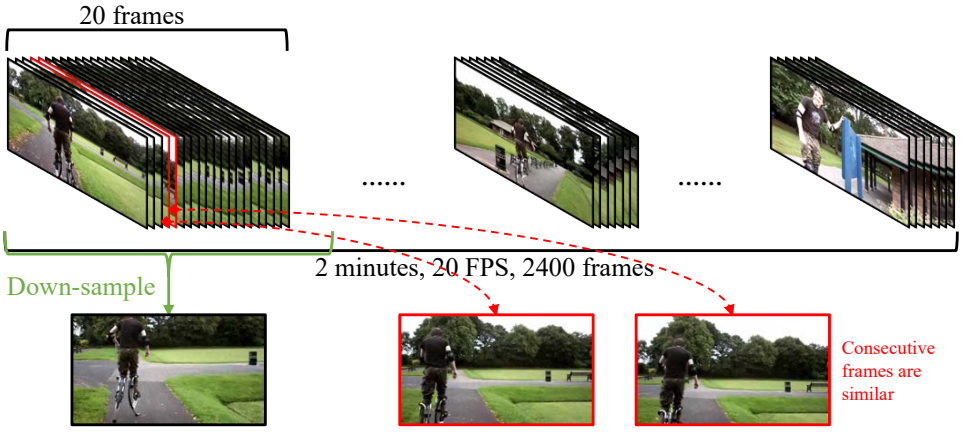


FIGURE 2.2: An example of video frames down-sampling.

is continuous, *i.e.,* changes between consecutive frames are usually small and smooth. Hence it is reasonable to downsample video for efficient computation. As illustrated in Figure 2.2, if we sample 1 frame from every 20 consecutive frames, we only need to process 120 frames instead of 2400 frames for this 2-minute video. With a downsample rate r_{ds}, the number of video frames becomes $T' = T/r_{ds}$. Downsample rate has a direct impact on video quality and should be carefully selected depending on the dataset.

Language query is discrete and words in a sentence demonstrate syntactic structure. Different word combinations lead to very different semantic meanings. For instance, given a query sentence "*The man speeds up then returns to his initial speed.*", the words "*initial*" and "*speed*" carry different meanings, and their combination describes a specific scene. For preprocessing, the query is typically tokenized into word tokens. If a query contains too many words, a common strategy is truncation, *i.e.,* taking a fixed number of words from the beginning and discarding the rest.

2.1.2 Feature Extractor

The feature extractor bridges the raw inputs and the model by converting inputs into feature representations. **Textual Feature Extractor** maps a query sentence to an embedding space, which can be categorized into token-level and sentence-level extractors. Token-level extractor converts each word into its corresponding word embedding by using pre-trained word embeddings (PWE), *e.g.,* Word2Vec [40] and GloVe [26], or pre-trained language models (PLM), *e.g.,* BERT [10] and RoBERTa [41]. We represent token-level extraction as:

$$Q = [q_1, \ldots, q_m] \xmapsto{\text{PWE/PLM}} \boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m] \in \mathbb{R}^{m \times d_q}, \tag{2.2}$$

where d_q denotes word embedding dimension.

Sentence-level extractor encodes the entire query into a sentence feature in d_s dimension, by using the pre-trained sentence encoder (PSE), *e.g.,* the Skip-Thought [42], InferSent [43], Sentence-BERT [44], or the PWE/PLM with a trainable sentence encoder (TSE). We represent the process as:

$$\begin{aligned} Q = [q_1, \ldots, q_m] &\overset{\text{PSE}}{\longmapsto} \boldsymbol{q}_s \in \mathbb{R}^{d_s}, \text{ or} \\ Q = [q_1, \ldots, q_m] &\overset{\text{PWE/PLM}}{\longmapsto} \boldsymbol{Q} \in \mathbb{R}^{m \times d_q} \overset{\text{TSE}}{\longmapsto} \boldsymbol{q}_s \in \mathbb{R}^{d_s}. \end{aligned} \tag{2.3}$$

Visual Feature Extractor converts video frames into a sequence of visual features. Depending on whether proposals are generated directly on input video, there are two types of feature extraction. Recall that a proposal is a candidate answer. A straightforward approach is to sample video segments from input video as proposals. Proposals may contain the different number of frames. Suppose there are n_{seg} video segments as proposals, the feature extraction process is described as:

$$V \in \mathbb{R}^{T' \times frame} \overset{\text{proposals}}{\longmapsto} \{\text{segment}_i \in \mathbb{R}^{\chi \times frame}\}_{i=1}^{n_{\text{seg}}} \overset{\text{visual feature extractor}}{\longmapsto} \boldsymbol{V} = \{\boldsymbol{v}_{p,i} \in \mathbb{R}^{d_v}\}_{i=1}^{n_{\text{seg}}}, \tag{2.4}$$

where χ is the number of frames in a proposal, and d_v denotes the dimension of extracted features. The task becomes to determine whether a proposal represented by $\boldsymbol{v}_{p,i}$ is the correct answer.

If proposals are not generated directly from the input video, then the video is uniformly decomposed into a sequence of non-overlapping snippets. Suppose there are n_{snp} video snippets and each snippet contains ξ frames, the extraction process is:

$$V \in \mathbb{R}^{T' \times frame} \overset{\text{decompose}}{\longmapsto} [\text{snippet}_i]_{i=1}^{n_{\text{snp}}} \in \mathbb{R}^{n_{\text{snp}} \times \xi \times frame} \overset{\text{visual feature extractor}}{\longmapsto} \boldsymbol{V} = [\boldsymbol{v}_i]_{i=1}^{n_{\text{snp}}} \in \mathbb{R}^{n_{\text{snp}} \times d_v}. \tag{2.5}$$

Here we distinguish "video snippet" from "video segment". The video segment is sampled as a proposal to match the target moment, and the video snippet is a very short clip that only contains a few frames, *i.e.,* $\xi << \chi$ in general. Further, as each video segment is one candidate answer, video segments are irrelevant to each other and they are further processed separately in TSGV. As very short clips, video snippets are maintained in a sequence, and are jointly processed in later stages. Each frame is a still image. From video frames to features, the commonly used visual feature extractor are (i) 3D-based ConvNet pre-trained for action recognition, *e.g.,* C3D [24] or I3D [25], and (ii) 2D-based ConvNet pre-trained for object detection, *e.g.,* VGG [45] or ResNet [1].

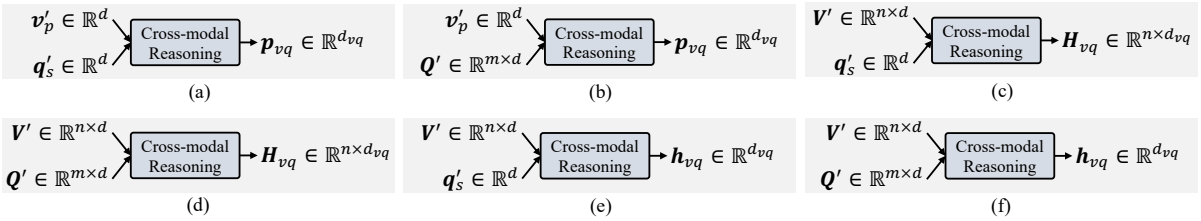


FIGURE 2.3: The common input/output (I/O) feature formats of feature interactor in TSGV, where $\mathbf{p}_{vq} \in \mathbb{R}^{d_{vq}}$ denotes the learned multimodal proposal feature; $\mathbf{H}_{vq} = [\mathbf{h}^1_{vq}, \ldots, \mathbf{h}^n_{vq}] \in \mathbb{R}^{n\times d_{vq}}$ is the multimodal snippet feature sequence; $\mathbf{h}_{vq} \in \mathbb{R}^{d_{vq}}$ is the pooled multimodal snippet feature; and d_{vq} denotes the dimension of output multimodal feature.

2.1.3 Feature Encoder and Interactor

Feature encoder maps visual and textual features to the same dimension, and refines their feature representations by encoding their corresponding contextual information. Existing TSGV methods use various feature encoders, from simple multi-layer perceptrons to complex transformers and graph neural networks. The design of the feature encoder highly depends on the model architecture. Briefed in Section 2.1.1, there are token-level and sentence-level query features. There are also two types of visual features, depending on whether the proposal generator is applied to input video, *i.e.,* proposal feature and video snippet feature sequence. Let d be the target dimension for both visual and textual features. Mapping of sentence-level and token-level query features is defined as:

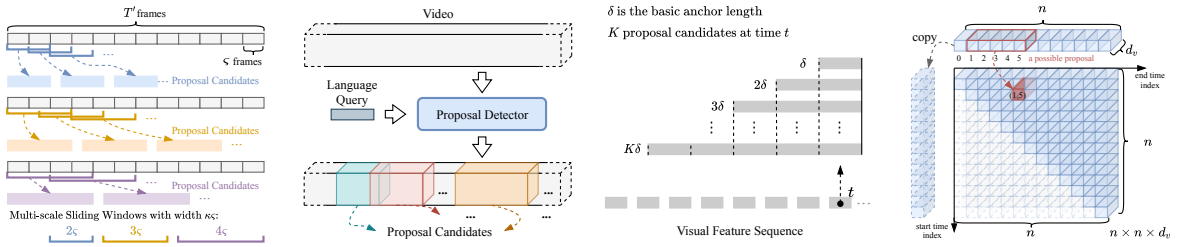
$$\boldsymbol{q}_s \in \mathbb{R}^{d_s} \xmapsto[\text{encoder}]{\text{textual feature}} \boldsymbol{q}'_s \in \mathbb{R}^d, \text{ and } \boldsymbol{Q} \in \mathbb{R}^{m\times d_q} \xmapsto[\text{encoder}]{\text{textual feature}} \boldsymbol{Q'} \in \mathbb{R}^{m\times d}. \tag{2.6}$$

For the proposal feature and video snippet feature sequence, the mapping is written as:

$$\boldsymbol{v}_p \in \mathbb{R}^{d_v} \xmapsto[\text{encoder}]{\text{visual feature}} \boldsymbol{v}'_p \in \mathbb{R}^d, \text{ and } \boldsymbol{V} \in \mathbb{R}^{n\times d_v} \xmapsto[\text{encoder}]{\text{visual feature}} \boldsymbol{V'} \in \mathbb{R}^{n\times d}, \tag{2.7}$$

where we simply use $\boldsymbol{v}_p \in \mathbb{R}^{d_v}$ to represent the visual feature of a proposal, and n to replace the n_{snp} or n_{seg}.

Feature interactor, an essential component in any TSGV method, aims to learn the cross-interaction between video and query. Recall the goal of TSGV is to retrieve a target moment from the video that *semantically corresponds* to the query. Thus, the feature interactor requires understanding the semantic meaning of the query and recognizing various activities in the video simultaneously. It then performs to fuse query and video representations by emphasizing the portion of video content that is most relevant to the query semantics. In general, the quality of the feature interactor determines the performance of a TSGV method to a large extent.



(a) SW strategy (b) PG strategy (c) Anchor-based strategy (d) 2D-Map strategy

FIGURE 2.4: Illustration of the sliding window (SW), proposal generated (PG), anchor-based, and 2D-Map strategies.

Figure 2.3 summarizes the various input and output formats of different feature interactors among existing TSGV methods. The input is determined by the types of query features (token-level or sentence-level), and the types of visual features (proposal or snippet sequence). The common output feature formats include (i) the learned multimodal proposal feature $\boldsymbol{p}_{vq} \in \mathbb{R}^{d_{vq}}$, (ii) the multimodal snippet feature sequence $\boldsymbol{H}_{vq} = [\boldsymbol{h}_{vq}^1, \ldots, \boldsymbol{h}_{vq}^n] \in \mathbb{R}^{n \times d_{vq}}$, and (iii) the pooled multimodal snippet feature $\boldsymbol{h}_{vq} \in \mathbb{R}^{d_{vq}}$. Here, d_{vq} is the dimension of the multimodal feature. The output format of the feature interactor is highly correlated to the answer predictor in a TSGV method. The answer predictor may depend on proposals that can be generated at different stages. We next brief proposal generation, before answer predictor.

2.1.4 Proposal Generation

Depending on whether a proposal generation module is used, existing TSGV methods can be roughly categorized into *proposal-based* and *proposal-free* methods. As shown in Figure 2.1, proposal generator can be integrated into the model at various positions. For instance, proposals can be directly sampled on the input video. Proposals can also be generated before or after the feature interactor based on the visual features. Anchor-based methods generate proposals during answer prediction. A method may also engage multiple proposal generation strategies.

Sliding window-based (SW) strategy [27, 28, 46–48] generates proposal candidates by densely sampling fixed-length video segments on input video, using pre-defined multi-scale sliding windows. SW strategy is usually performed directly on video frames. Illustrated in Figure 2.4(a), given a downsampled video with T' frames and a set of sliding windows, each sliding window samples video segments sequentially, with a preset overlap ratio. In our illustration, we use three different sliding windows $sw \in \{\kappa\varsigma\}_{\kappa=2,3,4}$ (ς denotes a basic window size) and set the overlap ratio as 0.5. The overlap ratio is necessary to increase the chance of covering the target moment. Then we have a set of video segments as proposals.

Proposal-generated (PG) strategy [29, 49–52] produces proposals by utilizing auxiliary modules, *e.g.,* a pre-trained segment proposal network (SPN) [53] or a carefully designed proposal detector. The PG strategy is usually performed on visual features, but it involves the query as input to guide its proposal generation process, illustrated in Figure 2.4(b). Hence, the proposal generated is related to the query. Depending on the position of the proposal detector, the PG strategy may also involve a feature encoder and interactor.

Anchor-based strategy [30, 31, 54, 55] generates proposals with pre-set multi-scale anchors. Different from the SW strategy, it is performed on the encoded visual features and is integrated into the answer predictor. Suppose we have K different scale anchors, and the length of a basic anchor is δ. Figure 2.4(c) plots a commonly used anchor-based strategy. This strategy applies K preset anchors to generate proposals, ending at a time step t, where t is the index of multimodal visual feature in the feature sequence.

Another version of anchor-based strategy is 2D-Map strategy [32, 56–59]. Different from the standard anchor-based strategy above, the 2D-Map strategy is usually applied after the feature extractor, *i.e.,* before the answer predictor. It generates proposals by modeling temporal relations between video moments through a two-dimensional map. One dimension indicates the start time of a moment; the other indicates the end time. Given a visual feature sequence with $n \times d_v$, all possible proposal candidates are computed based on a 2D temporal feature map. Shown in Figure 2.4(d), a candidate proposal representation can be computed by max-pooling the corresponding visual features across the specific time span, resulting in the 2D feature map with $n \times n \times d_v$. Note the start (a) and end (b) timestamps of a proposal candidate should satisfy $a \leq b$. Therefore, only proposal candidates that locate in the upper triangular part of the 2D map are valid.

2.1.5 Answer Predictor and Objective

The answer predictor is responsible for predicting the position of a target moment based on the learned multimodal features. Next, we brief the commonly used answer predictors and their corresponding objectives, for both proposal-based and proposal-free methods. Methods may combine multiple answer predictors or incorporate various auxiliary objectives to boost performance. In this background section, we only focus on the main objectives.

For proposal-based methods, the answer predictor computes a score for each proposal. Ideally, a proposal gets a higher score if it is closer to the ground-truth moment. Specifically, given a multimodal proposal feature $\boldsymbol{p}_{vq}$, its score is computed as $s = \sigma(\mathcal{A}(\boldsymbol{p}_{vq})) \in \mathbb{R}^1$, where $\mathcal{A}$ denotes the answer predictor and σ is an (optional) activation function. Then, the proposal with

the highest score is selected as the answer. If the proposals are generated by anchor-based strategy, the score is computed based on the multimodal snippet feature sequence $\boldsymbol{H}_{vq}$ by applying multi-scale anchors in the answer predictor.

Various learning objectives have been developed for proposal-based methods. The alignment loss is commonly used for SW and PG strategies, defined as:

$$\mathcal{L}_{aln} = \gamma \log(1 + e^{-s_{i,i}}) + \sum_{j=0, j\neq i}^{N_{neg}} \log(1 + e^{s_{i,j}}), \tag{2.8}$$

where $s_{i,i}$ is the score of aligned (or positive) proposal-query pair, and $s_{i,j}$ is the score of misaligned (or negative) pair; γ is a hyper-parameter to control the weight between positive and negative pairs; N_{neg} represents the number of negative pairs. For a given query, a proposal is considered positive if it has a good overlap with the ground truth moment, measured by IoU (intersection area over union area). Otherwise, the proposal is negative. Nevertheless, a negative pair can also be constructed by replacing a random query or pairing random but unmatched proposals and queries. In general, $\mathcal{L}_{aln}$ encourages aligned proposal-query pairs to have positive scores and misaligned pairs to have negative scores. Besides, the triple-based ranking loss has also been used for SW and PG strategies:

$$\mathcal{L}_{triple} = \max(0, \eta + s' - s), \tag{2.9}$$

where s denotes the score of matched proposal-query pair and s' is the score of the mismatch proposal-query pair. Similarly, $\mathcal{L}_{triple}$ encourages similarities between aligned pairs to be greater than misaligned pairs by some margin $\eta > 0$.

For anchor and 2D-Map strategies, binary cross-entropy is usually adopted, defined as:

$$\mathcal{L}_{bce} = \gamma \cdot y \cdot \log s + (1 - y) \cdot \log(1 - s), \tag{2.10}$$

where γ is an optional balance weight, determined based on the number of positive and negative samples. y is the corresponding anchor label for the proposal; $y = 1$ if the proposal candidate has IoU with ground-truth moment larger than a threshold θ, *i.e.,* positive. Otherwise $y = 0$. y may also be defined as the scaled IoU value between the proposal and the ground-truth moment.

Proposal-free methods do not generate proposals. They use a regressor or span predictor as the answer predictor. Specifically, regression-based predictor aims to regress the start and end times of the target moment directly. It takes the pooled multimodal snippet feature $\boldsymbol{h}_{vq}$ as input and predicts the temporal positions (t_s, t_e). Mathematically, we represent this process as $(t_s, t_e) = \sigma(\mathcal{A}(\boldsymbol{h}_{vq})) \in \mathbb{R}^2$, where $\mathcal{A}$ denotes the regressor, and σ is (optional) Sigmoid

activation to normalize the output to $[0, 1]$. Given a predicted (t_s, t_e) and the normalized ground-truth (τ_s, τ_e), the smoothed L_1 loss or MSE loss, *i.e.,* $R \in \{\text{smooth}_{L_1}, \text{MSE}\}$, is commonly used as learning objective:

$$\mathcal{L}_{reg} = R(t_s - \tau_s) + R(t_e - \tau_e). \tag{2.11}$$

Span predictor also predicts the start and end boundaries of the target moment directly. Different from the repression-based predictor, the span predictor computes the probability of each video snippet being the start and end points of the target moment. Specifically, it takes the multimodal snippet feature sequence $\boldsymbol{H}_{vq}$, and computes the start and end boundary scores as $(\boldsymbol{S}_s, \boldsymbol{S}_e) = \mathcal{A}(\boldsymbol{H}_{vq}) \in \mathbb{R}^{n\times 2}$. Then, the probability distributions of boundaries are computed by $\boldsymbol{P}_s = \text{softmax}(\boldsymbol{S}_s) \in \mathbb{R}^n$ and $\boldsymbol{P}_e = \text{softmax}(\boldsymbol{S}_e) \in \mathbb{R}^n$, where $\boldsymbol{P}^t_{s/e}$ denotes the probability of t-th snippet be the start/end boundary. Cross-entropy and Kullback-Leibler (KL) divergence are both commonly used for span prediction. The cross-entropy objective is defined as:

$$\mathcal{L}_{span} = f_{XE}(\boldsymbol{P}_s, \boldsymbol{Y}_s) + f_{XE}(\boldsymbol{P}_e, \boldsymbol{Y}_e), \tag{2.12}$$

where f_{XE} is the cross-entropy loss; $\boldsymbol{Y}_s$ and $\boldsymbol{Y}_e$ denote the ground-truth labels for start and end boundaries, respectively. $\boldsymbol{Y}_{s/e}$ is a n-dim one-hot vector, which is generated by setting the index of the snippet contains $\tau_{s/e}$ as 1, and others as 0. Similarly, the KL-divergence objective is defined as:

$$\mathcal{L}_{span} = D_{KL}(\boldsymbol{P}_s||\hat{\boldsymbol{Y}}_s) + D_{KL}(\boldsymbol{P}_e||\hat{\boldsymbol{Y}}_e), \tag{2.13}$$

where D_{KL} denotes KL-divergence; $\hat{\boldsymbol{Y}}_s$ and $\hat{\boldsymbol{Y}}_e$ are the ground-truth start and end boundary distributions. Not specified in an one-hot $\boldsymbol{Y}_{s/e}$, the ground-truth boundary distribution is formulated as $\hat{\boldsymbol{Y}}_{s/e} \sim \mathcal{N}(\tau_{s/e}, \sigma^2_{std})$, where $\mathcal{N}(\mu, \sigma^2_{std})$ is the normal distribution with expectation μ and standard deviation σ_{std}.

To summarize, we brief the main components of TSGV methods from input processing to answer prediction. Although existing TSGV methods may contain more sophisticated structures and diverse ancillary modules, their model frameworks generally follow this pipeline. Among the components, the effectiveness of the feature interactor highly affects TSGV performance. Proposal generation strategies are highly correlated with the design of answer predictor, and each strategy has its own advantages and drawbacks. Lastly, all methods rely on effective feature extractors, mainly developed in CV and NLP areas.

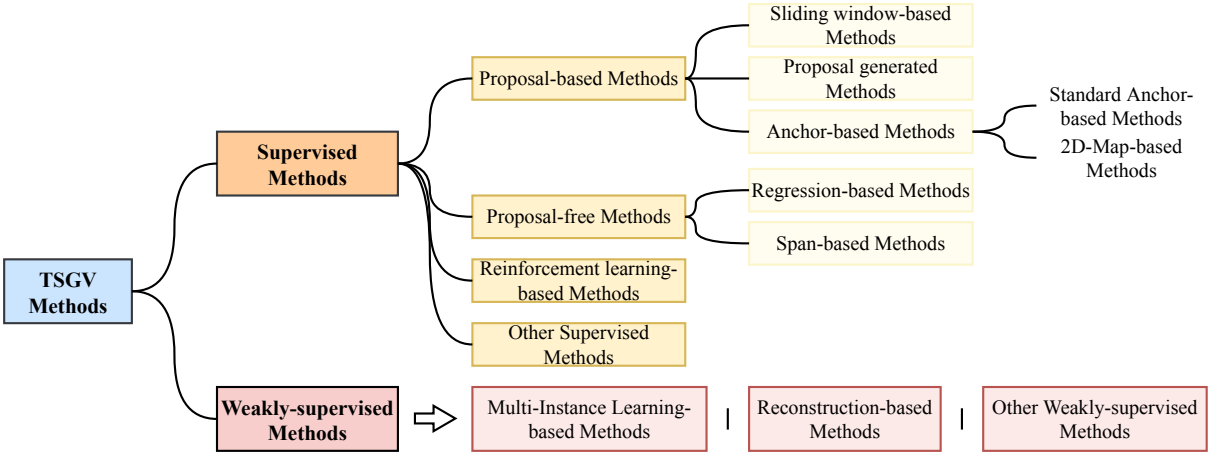


FIGURE 2.5: The taxonomy of TSGV methods.

2.2 TSGV Approaches

The majority of solutions proposed for TSGV belong to the supervised learning paradigm. Early solutions mainly rely on sliding windows or segment proposal network to pre-sample proposal candidates from input video. Then, the proposals are paired with the query to generate the best answers through cross-modal matching. However, this two-stage "*propose-and-rank*" pipeline is inefficient, because densely sampling candidates with overlap are essential to achieve high accuracy, leading to redundant computation and low efficiency. Meanwhile, the pairwise proposal-query matching may also neglect the contextual information. To overcome these drawbacks, alternative solutions like anchor-based and proposal-free methods are developed to address TSGV in an "*end-to-end*" manner. These methods encode the entire video sequence and all video information is maintained in the model, gradually becoming predominant solutions for TSGV.

Supervised learning requires a large number of annotated samples to train a TSGV method. Considering the difficulty and cost of data annotation, recent studies attempt to solve TSGV with weakly-supervised learning. These methods relieve the annotation burden by learning from video-query pairs without the detailed annotation of temporal locations of events in videos.

Accordingly, the simple classification of proposal-based and proposal-free methods in Section 2.1 is incapable of well covering all TSGV methods. Based on methods' architectures and learning algorithms, we propose a new taxonomy in Figure 2.5 to categorize TSGV methods. Next, we review solutions to TSGV following this taxonomy and discuss the characteristics of each method category. Because the majority are supervised learning solutions, this section is organized mainly based on the categories under supervised learning.

2.2.1 Supervised Method

2.2.1.1 Proposal-based Method

Depending on the ways to generate proposal candidates, proposal-based methods can be split into three categories, *i.e.,* sliding window-based, proposal-generated, and anchor-based methods. Sliding window-based and some proposal-generated methods follow a two-stage propose-and-rank pipeline, where the generation of proposal candidates is separated from the model computation. Anchor-based methods incorporate proposal generation in model computation to achieve end-to-end learning.

Sliding Window-based Method. The sliding window-based method adopts multi-scale sliding windows (SW) to generate proposal candidates (ref. Figure 2.4(a)). Then multimodal matching module finds the best matching proposal for a query. CTRL [27] and MCN [46] are two canonical SW methods, which are also pioneering work in TSGV. They define the task and construct corresponding benchmark datasets.

CTRL first produces proposals in various lengths through sliding windows, then encodes these proposals by a visual encoder, shown in Figure 2.6 (reproduced from Gao et al. [27]). The query is converted to sentence representation via a textual encoder. For cross-modal reasoning, it builds a relatively simple multi-modal processing module with three operators, *i.e.,* add, multiply, and fully connected (FC) layer, to fuse visual and textual features. CTRL designs multi-task objectives by using both alignment predictor and regressor. The alignment predictor computes matching score between the proposal and query (ref. Equation (2.8)). However, for an aligned proposal-query pair, the position of the proposal may not match the ground-truth moment exactly. The regressor uses smoothed L_1 loss to compute the corresponding offsets (ref. Equation (2.11)) to better align the proposal.

Different from CTRL, MCN aims to project both proposal and query features to a common embedding space. Then, it encourages the distance between the query and aligned proposal to be smaller than that of negative proposals. Specifically, MCN minimizes the squared distance between query and proposals to supervise model learning. Negative proposals can be misaligned proposals within the same video (intra-video), or proposals from other videos (inter-video). Thus, MCN builds both intra and inter-triple-based ranking loss (ref. Equation (2.9)) as objectives. The intra-loss differentiates subtle differences within a video, and the inter-loss differentiates broad semantic concepts. Based on MCN, Hendricks et al. [60] further propose MLLC, which treats video context as a latent variable and unifies MCN and CTRL.

The prior methods encode the entire query into one feature vector and apply simple cross-modal reasoning for feature fusion. However, treating queries holistically may obfuscate the

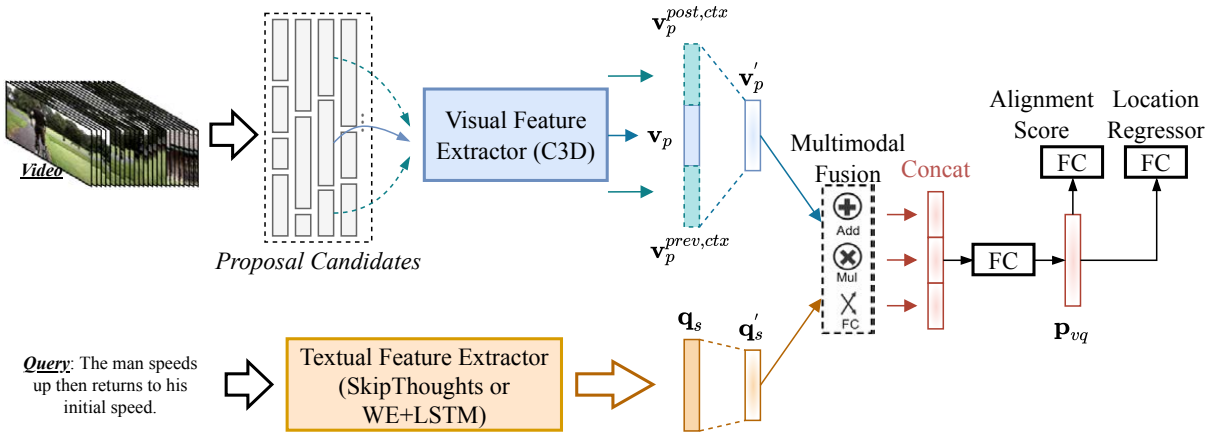
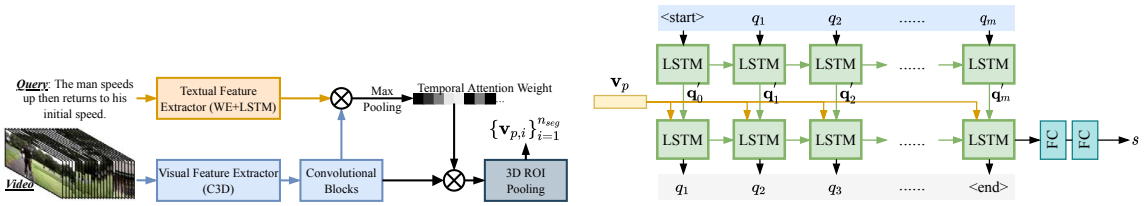


FIGURE 2.6: An illustration of the CTRL architecture, which is a canonical sliding window-based method.

keywords that have rich temporal and semantic cues. The simple fusion strategy also leads to inferior cross-modal understanding. Temporal dependencies and reasoning between video events and texts are not fully considered. Also, spatial-temporal information inside the video or query is overlooked. A number of methods are proposed to address these issues. Among them, ROLE [47], MCF [48], ACRN [61], TCMN [62], and ASST [63] mainly focus on refining the multimodal interaction/fusion between visual and textual features, through more sophisticated structures or semantic decomposition of video/query. ACL [28], built upon CTRL, explicitly mines activity concepts from both video and language modalities as prior knowledge, to calibrate the confidence of the proposal to be the target moment. In addition to multimodal interaction refinement, SLTA [64] and MMRG [65] also exploit to incorporate appearance knowledge, *i.e.,* object-level spatial visual features, to enhance cross-modal reasoning as an additional view of video content. Instead of generating proposals at the initial stage, Ning et al. [66] equip SW strategy inside their model enabling end-to-end training.

In general, early SW-based methods have simple architectures. These methods lack both in-depth analyses of semantic knowledge of modalities and fine-grained multimodal fusion mechanisms, leading to inferior performance. The following up work attempts to address these weaknesses by devising various techniques to better exploit video content and query, enhancing cross-modal reasoning between them. Despite continuous improvements, the two-stage sliding window-based methods suffer from inevitable drawbacks. Specifically, densely sampling proposals with multi-scale sliding windows results in heavily computational cost, as many overlapped areas are re-computed. These methods are also sensitive to negative samples, where fallacious negative samples may lead to inferior results.

Proposal Generated Method. Proposal generated (PG) method alleviates the computation



(a) Query-guided segment proposal network. (b) The early fusion retrieval model of QSPN.

FIGURE 2.7: An illustration of the QSPN architecture, which is a canonical proposal generated method.

burden of SW-based methods by avoiding a dense sampling process. Instead, PG methods generate proposals conditioned on the query. The number of proposals hence reduces remarkably.

Early proposal-generated methods still follow the two-stage propose-and-rank pipeline. Xu et al. [50] employ a pre-trained segment proposal network (SPN) [53] for proposal candidate generation, rather than adopting sliding windows. Based on Xu et al. [50], QSPN [29] further ameliorates SPN to produce query-specific proposal candidates. As illustrated in Figure 2.7(a) (reproduced from Xu et al. [29]), QSPN interacts query embedding with visual features to derive temporal attention weights and re-weights the visual features to refine proposal generation. With the generated proposal feature, QSPN sequentially encodes the proposal with each token in query and predicts the similarity score, at last, shown in Figure 2.7(b) (reproduced from Xu et al. [29]). QSPN is optimized by triple-based ranking loss (ref. Equation (2.9)), while a captioning loss is adopted to improve performance via query re-generation. Similarly, SAP [49] directly trains a visual concept detector to generate proposal candidates by measuring visual-semantic correlations between query and video frames.

Although the two-stage PG methods mitigate computation complexity to some degree, they still encounter some ineluctable drawbacks. To achieve good performance, PG methods still need to sample proposal candidates relatively densely, to increase the chance that at least one proposal can cover or is close to the ground-truth moment. Similar to SW-based methods, the two-stage PG methods also rely on ranking-based objectives, making them sensitive to negative samples. Besides, proposal candidates are processed separately; hence individual pairwise proposal-query matching may neglect the contextual information.

To overcome these defects, recent solutions [51, 52, 67] reformulate the pipeline of PG methods to a single-pass pattern, in an end-to-end manner. Specifically, BPNet [51] and APGN [52] propose to replace the separate proposal generator with a proposal-free module (detailed in Section 2.2.1.2) and jointly train it with the main model. In this case, the proposal generation is supervised by ground-truth moment, and only a few proposals are required to be generated. Besides, since the whole video is encoded as a feature sequence (ref. Section 2.1.3), visual

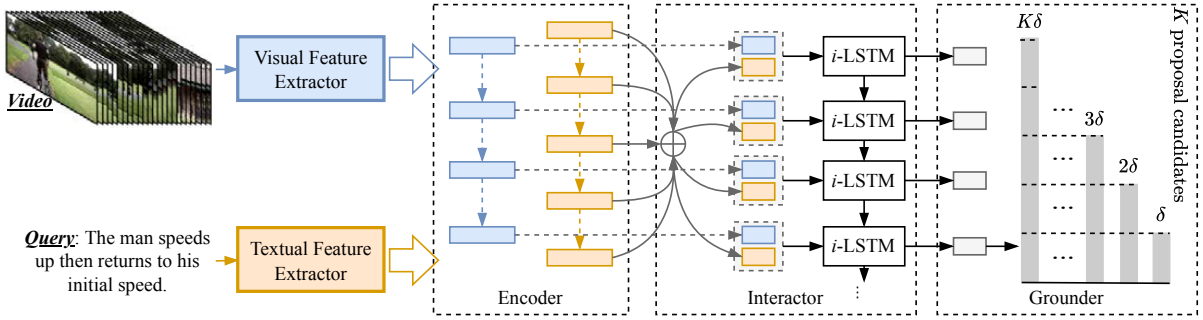


FIGURE 2.8: An illustration of the TGN architecture, which is a canonical standard anchor-based method.

features are jointly learned and interacted with the query. Thus, the model is able to consider contextual information. LPNet [67] maintains a boundary-aware predictor and a learnable proposal module in parallel, where the boundary-aware predictor could refine predictions of the learnable proposal module. Furthermore, CMHN [68] generates proposal candidates with 1D regular convolution, and modeling proposal-query matching in the Hamming space through cross-modal hashing.

Anchor-based Method. Sliding window and the early proposal-generated methods follow the two-stage propose-and-rank pipeline which suffers from various drawbacks. Researchers then source for alternative structures without pre-cutting proposal candidates at the input stage. One kind of solution is anchor-based methods, which incorporate proposal generation into answer prediction and maintain the proposals with various learning modules. According to how the anchors are produced and maintained, we further classify them into standard anchor-based and 2D-Map methods.

Standard Anchor-based Methods. Methods in this category produce proposal candidates with multi-scale anchors and maintain them sequentially or hierarchically. They aggregate contextual multimodal information and generate the final grounding result in one pass. The first anchor-based method for TSGV is Temporal GroundNet (TGN) by Chen et al. [30], shown in Figure 2.8 (reproduced from Chen et al. [30]). TGN temporally captures the evolving fine-grained frame-by-word interactions between video and query. At each time step, multi-scale proposal candidates ending at the current time are generated using pre-set anchors. Then a sequential LSTM grounder simultaneously scores the group of proposals. TGN adopts weighted binary cross-entropy loss (ref. Equation (2.10)) to optimize the model. In contrast, MAN [54] and SCDM [31] adopt temporal convolutional network to produce proposal candidates hierarchically. That is, proposals with different scales are generated at different levels of the stacked temporal convolution module.

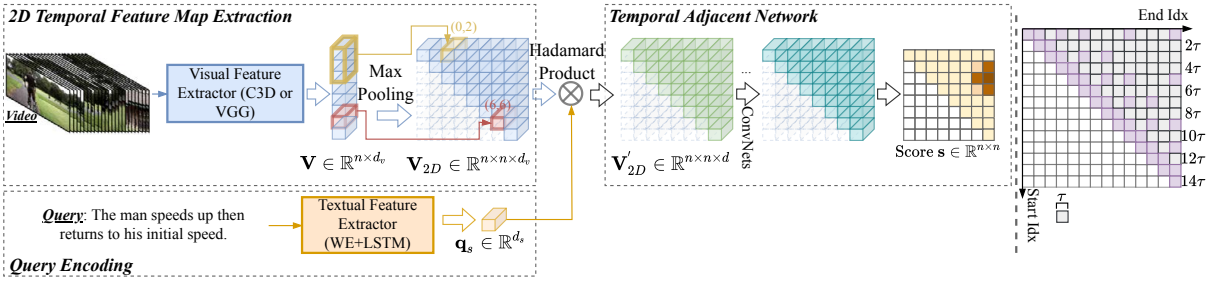


FIGURE 2.9: An illustration of the 2D-TAN architecture, which is a canonical 2D-Map based method.

Subsequent work generally follows the strategies of TGN or SCDM with more sophisticated learning modules and/or auxiliary objectives. To be specific, CMIN [55], CBP [69], FIAN [70], HDRR [71], and MIGCN [72] adopt the strategy of TGN, while CSMGAN [73], RMN [74], IA-Net [75], and DCT-Net [76] apply the strategy of SCDM. These solutions design various cross-modal reasoning strategies to perform more fine-grained and deeper multi-modal interaction between video and query, for precise moment localization. In addition, CBP [69] introduces an auxiliary boundary module to compute the confidence of the feature at each time step to be the boundary of the target moment. Some other works adopt boundary regression module to refine the start and end timestamps of generated moments. MIGCN [72] develops a rank module apart from the boundary regression module to distinguish the optimal proposal from a set of similar proposal candidates.

2D-Map Anchor-based Methods. Standard anchor-based method produces proposal candidates with preset multi-scale anchors and maintains them sequentially or hierarchically. These proposals are individually processed and their temporal dependencies are not well considered. Further, the lengths of proposals are restricted by preset anchors. 2D-map anchor-based methods use a two-dimensional map to model temporal relations between proposal candidates, shown in Figure 2.4(d). Theoretically, 2D-Map could enumerate all possible proposals at any length, while maintaining their adjacent relations.

Before 2D-Map methods, a prior work TMN [77] first proposes to enumerate all possible consecutive segments as proposals and predict the best-matched proposal as result through interacting each proposal with query. However, TMN generates proposals in the answer predictor; its enumeration strategy is more like a variant of the standard anchor-based strategy.

2D-TAN [32] is the first solution modeling proposals with a 2D temporal map, and its overall architecture is shown in Figure 2.9 (reproduced from Zhang et al. [32]) left. 2D-TAN first extracts visual features and converts them into a 2D feature map, while the query is encoded in

sentence-level representation. A temporal adjacent network is proposed to fuse the query feature with each proposal feature and embed the video context information with a convolutional network. As shown in Figure 2.9 right, 2D-TAN divides video into evenly spaced video snippets with duration τ, where (i, j) on the 2D map denotes a proposal candidate from time $i\tau$ to $j\tau$[2]. Instead of enumerating all possible consecutive segments as proposals, 2D-MAN proposes a sparse sampling strategy to remove redundant moments which have large overlaps with the selected proposals. The model adopts binary cross-entropy loss for model learning. 2D-TAN is further extended with multi-scale modeling [78] to achieve a larger receptive field and obtain a richer context. The extended version reduces the complexity of proposal generation from quadratic to linear, making dense video prediction more efficient.

Due to its effectiveness, a series of works follows 2D-TAN's proposal generation[3] or its overall structure. 2D-TAN directly encodes query into sentence-level representation and interacts with proposals via simple Hadamard product. In this sense, multimodal interaction is overlooked. To remedy, PLN [79], SMIN [57], CLEAR [80], and STCM-Net [81] disentangle video proposals into different temporal granularities [79, 81] or different semantic contents [57, 80], and perform cross-modal reasoning at both coarse- and fine-grained granularities. VLG-Net [82] and RaNet [58] maintain query words and video proposals in a graph, and adopt GCN [83, 84] to conduct intra- and inter-modal interactions for cross-modal reasoning. SV-VMR [85] decomposes query into multiple semantic roles [86] and performs multi-level cross-modal reasoning at semantic level. MATN [56] further concatenates proposals and query words into a sequence and encodes them through a single-stream transformer network. It also devises a novel multi-stage boundary regression to refine the predicted moments. Instead of using the simple Hadamard product, DMN [87] proposes to project proposals and query features to common embedding space and leverage metric learning for cross-modal pair discrimination. Moreover, FVMR [59] claims that the standard cross-modal interaction module is inefficient and replaces it with a semantic embedding module to model multimodal interaction.

2.2.1.2 Proposal-free Method

Proposal-based methods perform various proposal generations and essentially depend on the ranking of proposal candidates. In contrast, proposal-free methods directly predict the start and end boundaries of the target moment on fine-grained video snippet sequence, without ranking a vast of proposal candidates. Depending on the format of moment boundaries, proposal-free methods are categorized into regression-based and span-based methods.

[2] $i \leq j$, *i.e.,* only the upper triangular area of 2D map is valid

[3] Some methods follow 2D-TAN's proposal generation to produce proposal candidates, but they may not maintain the proposals with 2D map.

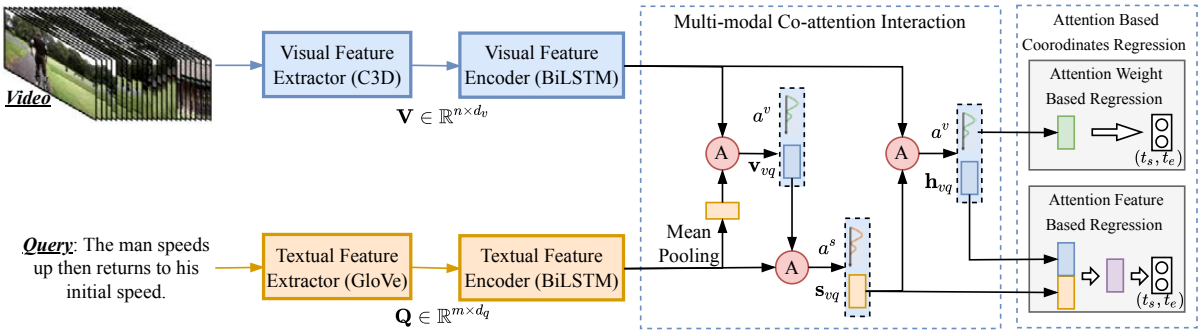


FIGURE 2.10: An illustration of ABLR architecture, which is a canonical regression-based method.

Regression-based Method. Regression-based method computes a time pair (t_s, t_e) and compares the computed pair with ground-truth (τ_s, τ_e) for model optimization. Attention-based location regression (ABLR) [88] is one of the first regression-based solutions for TSGV. Depicted in Figure 2.10 (reproduced from Yuan et al. [88]), ABLR extracts visual and textual features and encodes them through BiLSTM networks to aggregate contextual information, respectively. Then, a three-stage multimodal co-attention is developed to perform cross-modal reasoning. The multimodal feature is fed to the regressor for moment prediction. ABLR explores two types of regressors. One is attention weight-based regression, which takes video attention weights as inputs. Another is attended feature-based regression, which fuses the attended visual and textual features as inputs. The model is optimized by smoothed L_1 loss. ABLR also devises an attention calibration loss to refine video attention, which encourages higher attention weights to video snippets within the ground-truth moment.

Concurrently, ExCL [89] also addresses TSGV by regression and designs three different answer predictors following ideas from reading comprehension in NLP [15, 16, 90]. Similar to proposal-based methods, subsequent regression work [91–98] dives in designing various feature encoding and cross-modal reasoning strategies for superior multimodal interaction and accurate moment localization. From the perspective of regression, DEBUG [91], GDP [92], and DRN [94] analyze data imbalance issue in TSGV: the number of video frames is large, but the positive samples are sparse *i.e.,* only two frames for start and end timestamps. They regard all frames within the ground-truth moment as positive and densely predict the distances to boundaries for each frame within the ground-truth moment to mitigate the sparsity issue. CMA [93] and DeNet [97] study bias issue in TSGV. Specifically, CMA [93] rectifies the inevitable annotation bias by moment boundary ambiguities via a two-branch cross-modality attention network and a task-specific regression loss. DeNet [97] disentangles query into relation and modified features and devises a debias mechanism to alleviate both query uncertainty and annotation bias issues.

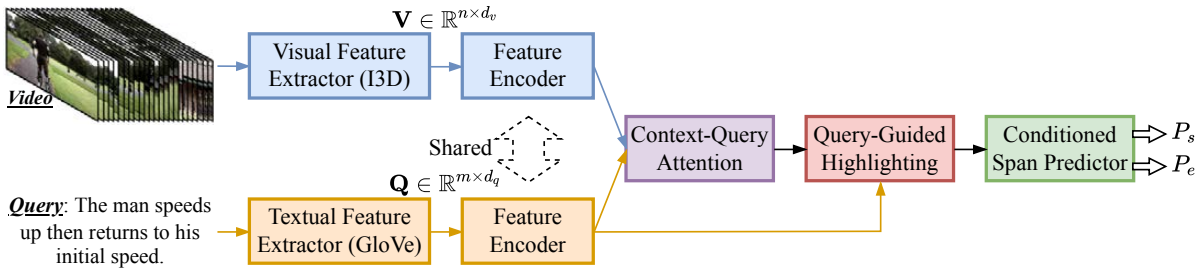


FIGURE 2.11: An illustration of VSLNet architecture, which is a canonical span-based method.

There are also regression-based methods [99–101] incorporating additional modalities from video to improve localization performance. For instance, HVTG [99] extracts both appearance and motion features from video. In addition, PMI [100] further exploits audio features from video extracted by SoundNet [102]. DRFT [101] leverages visual, optical flow, and depth flow features of video, and analyzes retrieval results of different feature view combinations.

Span-based Method. Span-based methods aim to predict the probability of each video snippet/frame being the start and end positions of the target moment. Inspired by QA task in NLP [15, 16, 90], L-Net [103] and ExCL [89] first formulate TSGV as span prediction task.

Based on the two methods, Zhang et al. [34] further compares differences between QA and TSGV tasks and propose VSLNet. Shown in Figure 2.11 (reproduced from Zhang et al. [34]), VSLNet exploits a context-query attention modified from QANet [90] to perform fine-grained multimodal interaction. Then a conditioned span predictor computes the probabilities of the start/end boundaries of the target moment. VSLNet also introduces a query-guided highlighting module to bridge the gaps between QA and TSGV. This module effectively narrows down moment search space to a smaller highlighted region. Existing methods including VSLNet generally perform better on short videos than on long videos. Their follow-up work [35] extends VSLNet to handle long videos by incorporating the concepts from multi-paragraph question answering [33]. Long videos are split into multiple short videos and a hierarchical searching strategy is deployed for moment localization.

In general, overall frameworks of regression- and span-based methods are similar. Thus, the continuous performance improvements of subsequent work [36, 104–115] are also achieved by modifying the feature encoding and multimodal interaction modules, introducing auxiliary objectives, and/or augmenting additional features. In particular, SeqPAN [36] introduces the concepts of NER [116–118] in NLP by splitting snippet sequence into begin, inside, and end regions of target moment, and background region. IVG-DCL [106] introduces a dual contrastive learning mechanism to enhance multimodal interaction and leverages structured causal model [119] to address the selection bias of TSGV. CI-MHA [108] proposes to remedy the

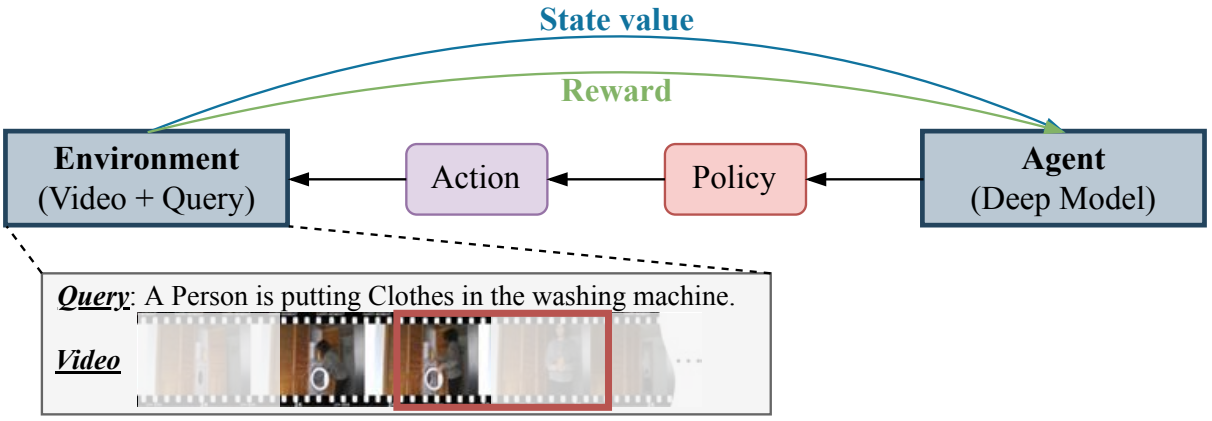


FIGURE 2.12: An illustration of sequence decision making formulation in TSGV.

start/end prediction noise caused by annotator disagreement via auxiliary moment segmentation task. ABIN [111] devises auxiliary adversarial discriminator networks to produce coordinate and frame correlation distributions for moment boundary refinement. DORi [113] incorporates appearance features and captures relations between objects and actions guided by the query. CBLN [114] addresses TSGV from a new perspective. It reformulates TSGV by scoring all pairs of start and end indices simultaneously and predicting moments with a biaffine structure.

2.2.1.3 Reinforcement Leaning-based Method

From the perspective of proposal usage, reinforcement learning (RL) based methods are also proposal-free methods. However, the task formulation of the RL-based method is fundamentally different from the proposal-free methods reviewed earlier. RL-based method formulates TSGV as a sequence decision-making problem and utilizes deep reinforcement learning techniques to solve it. Illustrated in Figure 2.12, the RL-based method usually maintains a sliding window (the dark red rectangle). The sliding window here is different from that discussed in Section 2.2.1.1. The RL-based method only adopts a single window, controlled by an agent. An agent, *i.e.,* a learnable module, controls the movement of the window based on a set of handcraft-designed temporal transformations, *e.g.,* shifting, and scaling. At each learning step, after each movement, a reward is generated to indicate whether the window is closer or farther away from the target moment. The agent will adapt its action for the next step within the pre-defined action space.

RWM-RL [120] is one of the first works to define and solve TSGV with an RL framework. Shown in Figure 2.13 (reproduced from He et al. [120]), RWM-RL consists of three modules. The environment module converts the query, global video, and local video segment within the window into corresponding representations. Then the observation network fuses query and

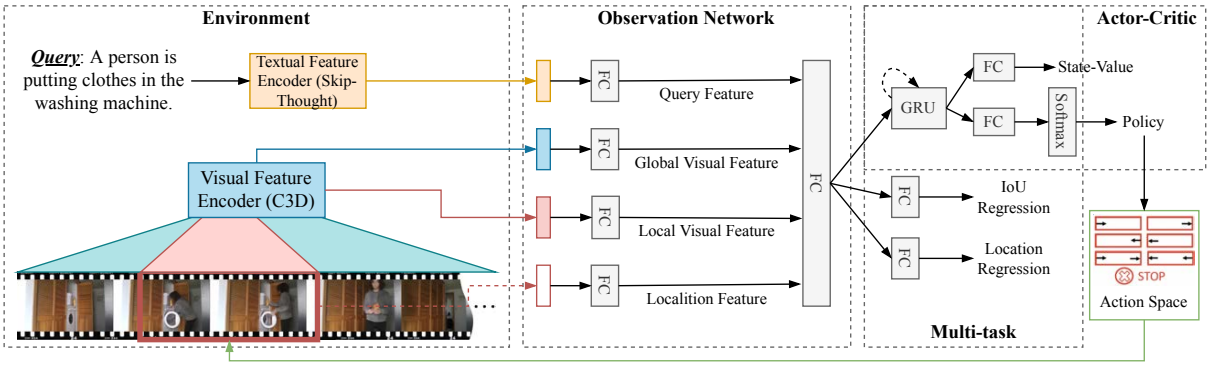


FIGURE 2.13: An illustration of RWM-RL architecture, which is a canonical reinforcement learning-based method.

video features to output the current state of the environment, *i.e.,* multimodal representation, at each learning step. In the decision-making module (*i.e.,* agent), RWM-RL leverages actor-critic algorithm [121] to compute state-value and action policy, *i.e.,* the probabilistic distribution of all pre-designed actions in action space. The state-value is used for reward computation, and the action policy determines the movement of the sliding window to adjust temporal boundaries. RWM-RL defines 7 actions: six moving/shifting actions and a STOP action. In general, the iterative process ends when encountering STOP action or reaching the preset maximum number of iteration steps. RWM-RL adopts GRU to model the sequential decision-making process for the actor-critic module. A reward is computed at each step, where the reward is designed to encourage the agent to find a better matching position step by step. All rewards are accumulated for model optimization by utilizing the advantage function [121] as objective and Monte Carlo sampling [122] for policy gradient approximation. To increase action diversity, RWM-RL further introduces entropy of the policy output as an auxiliary objective following A2-RL [123].

SM-RL [124] presents an RNN-based semantic matching RL model to selectively observe proposal candidates produced by a controllable agent. TSP-PRL [125] designs a hierarchical action space with a tree-structured policy, inspired by human's coarse-to-fine decision-making mechanism. The action selection is controlled by a switch over an interface in a tree-structured policy. AVMR [126] treats the RL-based module as a generator and devises a Bayesian ranking module as a discriminator to rank proposals. STRONG [127] considers both appearance and motion features and employs parallel spatial- and temporal-level RL modules for moment localization. TripNet [128] mainly focuses on ameliorating observation network to boost performance. Instead of using sliding windows, MABAN [129] leverages two individual agents to model start and end points separately.

2.2.1.4 Other Supervised Method

In addition to the aforementioned method categories, researchers also explore other types of formulation to address TSGV, or under different settings. Shao et al. [130] design a unified framework based on TAG [131] to perform both video-level retrieval and moment-level localization simultaneously. The two tasks could reinforce each other. DPIN [132] devises a dual-path interaction network to integrate the benefits of both proposal-based and proposal-free methods. Inspired by Patrick et al. [133], Ding et al. [134] propose a support-set based cross-supervision strategy to enhance multimodal interaction learning, through discriminative contrastive and generative caption objectives. Bao *et al.* [135] claim that multiple moments in a video are semantically correlated and temporally coordinated according to their order. Thus, they explore a novel setting of TSGV, dubbed as dense events grounding. This setting allows jointly localizing multiple moments described in a paragraph, *i.e.,* multiple sentences. SNEAK [136] studies the adversarial robustness of TSGV models by examining three facets of vulnerabilities, *i.e.,* vision, language, and cross-modal interaction, from both attack and defense aspects. Xu et al. [137] further investigate model pre-training for TSGV by constructing a large-scale synthesized dataset with annotations and designing a novel boundary-sensitive pretext task. Cao et al. [138] reformulate TSGV as a set prediction task, and propose a multimodal transformer model inherited from DETR [5].

2.2.1.5 Summary of Supervised Method

Hereto we have reviewed different categories of supervised TSGV methods, as well as their advantages and shortcomings. In general, early sliding window-based and proposal-generated methods suffer from low efficiency and flexibility, because of dense and overlapped proposals. These methods also rely on ranking-based loss, making them sensitive to negative samples. Anchor-based methods, another form of the proposal-based solution, learn TSGV in an end-to-end manner. The proposal generation process is incorporated into the model, abnegating the ineffective SW and PG strategies. Anchor-based methods also enable contextualized representation learning and fine-grained multimodal interaction. However, the anchor-based methods still need to maintain a mass of proposals during prediction, which hinders model efficiency.

Proposal-free methods directly learn to predict the boundaries of the target moment, without maintaining any proposals. These methods are more efficient and flexible to adapt to moments with diverse lengths. Nevertheless, compared to proposal-based methods, proposal-free methods overlook the rich information between start and end boundaries and fail to exploit the proposal-level interaction. They also suffer from severe imbalance issues between the positive and negative training samples, *i.e.,* only two (start and end) frames are positive in the

whole video. Also belonging to the proposal-free category, the design of RL-based methods is intuitive and effective, kind of simulating human's decision-making strategy. However, their performance is unstable due to the difficulty of optimizing RL-based methods.

Despite the vast number of methods in each category, all methods focus on ameliorating cross-modal reasoning module, to achieve fine-grained and precise multimodal interaction. That also means that the high-level pipeline of methods in each category is similar in general. Recall feature interactor is responsible for understanding semantic concepts of both query and video, and fusing them to emphasize video contents that are semantically relevant to the query. In this sense, the quality of the interactor module determines TSGV performance to a great extent.

2.2.2 Weakly-supervised TSGV Method

Supervised learning usually needs a large number of annotations for model training. Annotating temporal boundaries on video with text description is extremely time-consuming and labor-intensive, often not scalable. Further, annotations also suffer from the inaccurate issue, *i.e.,* action boundaries in videos are usually subjective and inconsistent across different annotators. Under a weakly-supervised setting, TSGV methods only need video-query pairs but not the annotations of starting/end time. They explore to find results in a shared multimodal feature space or with a reconstruction-based strategy. In general, existing weakly-supervised TSGV methods can be roughly grouped into multi-instance learning and reconstruction-based models.

Multi-Instance Learning Method. Multi-instance learning method generally regards the input video as a bag of instances with bag-level annotations. The prediction of instance, *i.e.,* proposal candidates, is aggregated as the bag-level prediction. TGA [139] first solves TSGV under the multi-instance learning setting. As shown in Figure 2.14 (reproduced from Mithun et al. [139]), TGA first encodes video and query features and presents text-guided attention to learn text-specific global video representations. Then both visual and textual features are projected into joint space. TGA regards the video and its corresponding query descriptions as positive pairs, while the video with other queries and the query with other videos as negative pairs. TGA learns visual-text alignment at the video level by maximizing matching scores of positive samples while minimizing scores of negative samples.

To achieve good performance, MIL-based methods have to perform precise semantics alignment between video and query. Thus, subsequent solutions [140–152] mainly focus on devising sophisticated cross-modal alignment module, designing robust proposal selection strategy, and/or building effective learning objectives. WSLLN [140] models alignment and detection

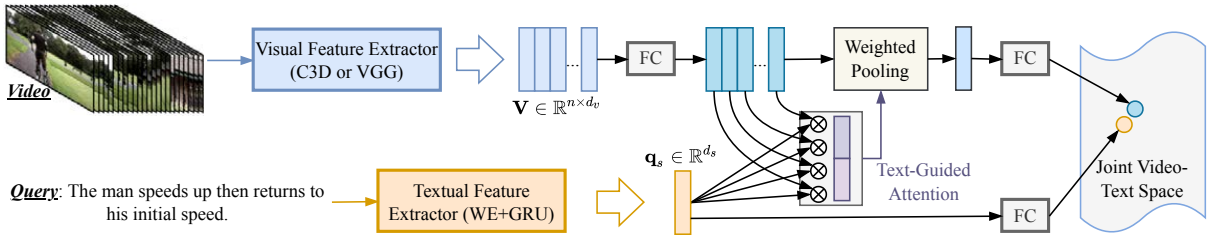


FIGURE 2.14: An illustration of TGA architecture, which is a canonical multi-instance learning method.

modules in parallel to perform proposal selection and video-level alignment simultaneously. VLANet [142] designs a surrogate proposal selection module to prune irrelevant proposal candidates. Chen et al. [141] and Teng et al. [150] perform video-query alignment at multiple granularities. CCL [144] and VCA [146] introduce contrastive learning mechanism to effectively distinguish positive and negative (or counterfactual positive) proposals. BAR [143] involves an additional RL module to progressively refine retrieved proposals. FSAN [147], WSTAN [151], and LoGAN [152] focus on mining video and query contents and their correlations. Then they design a fine-grained cross-modal alignment module for accurate moment localization. Da et al. [145] study the uncertain false-positive frame issue, *i.e.,* an object might appear sparsely across multiple frames and devise an AsyNCE loss to mitigate the issue by disentangling positive pairs from negative ones. CRM [148] uses a cross-sentence relation mining strategy to explicitly model cross-sentence relations in the paragraph and explore cross-moment relations in the video. LCNet [149] further deploys self-supervised cycle consistent loss to guide video-query matching.

Reconstruction-based Method. Reconstruction-based method tackles TSGV in an indirect way. Methods in this category first take video and query as inputs to produce desired proposals matched to the query. Then the proposals are used to reconstruct the query, where the intermediate proposals are treated as localization results. The idea of reconstruction is first explored by Duan et al. [153]. They propose a method to solve weakly supervised dense event captioning (WS-DEC), where moment localization is an auxiliary sub-task to assist model training. The authors indicate that moment localization and event captioning is a pair of dual tasks. Moment localization is to learn a mapping $l_{\theta_1} : (V, Q) \mapsto \boldsymbol{m}$, *i.e.,* retrieving a moment $\boldsymbol{m}$ corresponded to the caption C_i from video V. Event captioning inversely generates caption Q for the given $\boldsymbol{m}$, *i.e.,* $g_{\theta_2} : (V, \boldsymbol{m}) \mapsto Q$. Since Q and $\boldsymbol{m}$ are a one-to-one correspondence, the dual problems exist simultaneously, and Q and $\boldsymbol{m}$ are tired together. By nesting the dual functions,

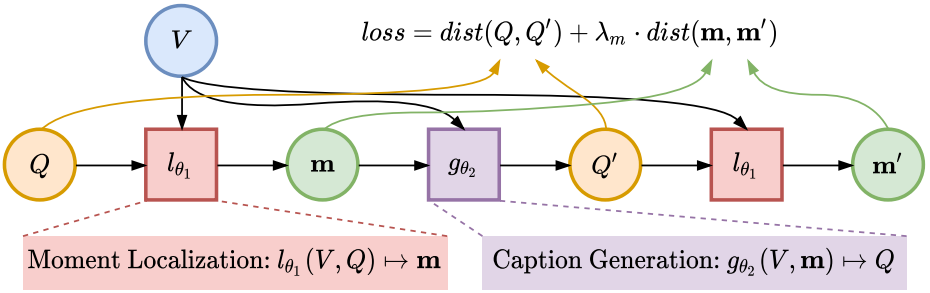


FIGURE 2.15: An illustration of WS-DEC architecture, which is a canonical reconstruction-based method.

caption-moment pair $(Q, \boldsymbol{m})$ becomes a fixed-point solution as:

$$Q = g_{\theta_2}(V, l_{\theta_1}(V, Q)), \quad \boldsymbol{m} = l_{\theta_1}(V, g_{\theta_2}(V, \boldsymbol{m})), \tag{2.14}$$

where l_{θ_1} and g_{θ_2} are localization and captioning modules, respectively. Shown in Figure 2.15 (reproduced from Duan et al. [153]), WS-DEC first retrieves moment $\boldsymbol{m}$ by giving video V and caption Q; Then the retrieved $\boldsymbol{m}$ and V are used to reconstruct the caption, denoted by Q'; Finally, the reconstructed Q' and V are utilized to relocate the moment $\boldsymbol{m}'$ again. The objective of WS-DEC is to minimize the distances of $\boldsymbol{m}$-$\boldsymbol{m}'$ and Q-Q' pairs simultaneously.

SCN [154] adopts a similar idea as WS-DEC. However, SCN is designed for solving weakly supervised TSGV directly; it does not use a specific caption generation module, but switches to reconstruct the masked query. Specifically, SCN first retrieves a set of proposals from video. The model then selects top-K proposals as inputs to reconstruct masked query, and compute rewards based on reconstruction loss. The rewards further act as feedback to refine proposal generation. MARN [155] leverages both proposal-level and clip-level video features to produce more accurate proposal candidates. The proposal-level and clip-level features are generated by 2D-Map strategy [32] and BMN [156], respectively. EC-SL [157] improves WS-DEC by introducing a concept learner and an induced set attention block.

Other Weakly-supervised Method. In addition to MIL and reconstruction-based methods, Zhang et al. [158] consider both inter- and intra-sample confrontments to address the drawback of standard MIL-based methods. The latter generally ignores intra-sample confrontment between moments with semantically similar contents. Luo et al. [159] present a new setting of TSGV task, also in a semi-supervised way. They construct a teacher-student network. The teacher module produces instant pseudo labels for unlabeled samples based on predictions. The student module learns from pseudo labels via self-supervised learning. Nam et al. [160] further propose to learn a TSGV model in a zero-shot manner to eliminate annotation cost. In the

zero-shot setting, video-query pairs are not provided. They utilize an off-the-shelf object detector and pseudo-query generation module fine-tuned on RoBERTa [41] to produce proposals and queries, and simulate the standard TSGV learning. Gao and Xu [161] explore leveraging an off-the-shelf visual concept detector and a pre-trained image-sentence embedding space to perform TSGV without using text annotations on video.

2.3 Datasets and Measures

2.3.1 Benchmark Datasets

Datasets are essential resources for building and evaluating TSGV methods. A TSGV dataset typically contains a collection of videos. Each video may come with one or more annotations, *i.e.,* moment-query pairs. Each annotation has a query corresponding to a moment in the video. A few TSGV datasets have been developed, covering various scenarios with distinct characteristics, *e.g.,* different scenes, and activity complexities, summarized in Table 2.1. For DiDeMo and MAD datasets, we directly obtain their statistical results from original papers. For others, we conduct statistics on raw datasets. We also filter out or modify some invalid annotations in each dataset.

DiDeMo. DiDeMo dataset has its root in YFCC100M [162] dataset, and the latter contains over $100k$ Flickr videos about various human activities. Hendricks et al. [46] randomly select over $14,000$ videos, then split and label video segments. Each segment is a five-second video clip, hence the length of the ground-truth moment is five seconds. DiDeMo dataset consists of $10,464$ videos and $40,543$ annotations in total, on average 3.87 annotations per video. Note that, the videos are released in the form of extracted visual features, hence we cannot provide detailed statistics in Table 2.1. Hendricks et al. [60] further collect a TEMPO dataset, which is built on top of DiDeMo, by augmenting more language queries via template model and human annotators. In particular, the template model utilizes the language templates to augment the original sentences in DiDeMo with template words. The template allows to generate a large number of sentences with known ground truth base and context moments. However, template language lacks the complexity of human language, so additional fully user-constructed samples are collected by human annotators, which contain more specific temporal words. Compared to DiDeMo, TEMPO contains more complex human-language queries.

Charades-STA. Charades-STA dataset is built by Gao et al. [27] from the Charades dataset [163]. The Charades dataset contains $9,848$ annotated videos about human daily indoor activities for

TABLE 2.1: Statistics of the TSGV benchmark datasets.

Dataset	DiDeMo	Charades-STA	ActivityNet Captions	$\text{TACoS}_{\text{org}}$	$\text{TACoS}_{\text{2DTAN}}$	MAD
Source	Flickr	Homes	YouTube	Lab Kitchen		Movie
Domain	Open	Indoor Activity	Open	Cooking		Open
# Videos	10,464	6,672	14,926	127		650
# Moments	26,892	11,767	71,953	3,290	7,069	-
# Queries	40,543	16,124	71,953	18,818	18,227	384,600
$\bar{L}_{A/V}$	3.87	2.42	4.82	148.17	143.52	-
N_{vocab}	7,785	1,303	15,505	2,344	2,287	61,400
$\bar{L}_V$	30.00s	30.60s	117.60s	286.59s		6,646.20s
$L_V^{\text{min / max}}$	-	5.50s / 194.33s	1.58s / 755.11s	48.30s / 1,402.18s		-
$\bar{L}_m$	-	8.09s	37.14s	6.10s	27.88s	4.10s
$L_m^{\text{min / max}}$	-	1.68s / 80.80s	0.05s / 408.80s	0.31s / 166.97s	0.48s / 843.20s	-
$\bar{L}_Q$	-	7.22	14.41	10.05	9.42	12.70
$L_Q^{\text{min / max}}$	-	3 / 13	4 / 91	2 / 229	2 / 69	-

* $\bar{L}_{A/V}$ denotes the average queries or annotations per video; N_{vocab} represents the vocabulary size; $\bar{L}_V$ and $L_V^{\text{min / max}}$ denote the average and min/max video length in seconds, respectively; $\bar{L}_m$ and $L_m^{\text{min / max}}$ are the average and min/max moment length in seconds, respectively; and $\bar{L}_Q$ and $L_Q^{\text{min / max}}$ represent the average and min/max query length. Note the number of moments and queries are not equal, since different queries may correspond to the same moment.

video activity recognition. The original dataset provides $27,847$ video-level sentence descriptions, $66,500$ temporally localized intervals for 157 action categories, and $41,104$ labels for 46 object categories. Based on Charades, Gao et al. [27] design a semi-automatic way to generate sentence temporal annotation and construct the Charades-STA dataset. They first decompose long video descriptions into sub-sentences by a set of conjunctions. Then, they parse the activity labels from video descriptions using Stanford CoreNLP [164] and match labels with sub-sentences. Finally, they align sub-sentences with the original label-indicated temporal intervals. In this way, a collection of (sentence query, target moment) pairs is generated as annotations. Because the original descriptions are quite short, Gao et al. [27] further combine consecutive descriptions into a more complex sentence to enhance description complexity for the test. Charades-STA dataset contains $6,672$ videos and $16,124$ annotations, where $12,404$ annotations for training and $3,720$ annotations for test. Average video length, moment length, and query length are 30.60 seconds, 8.09 seconds, and 7.22 words, respectively.

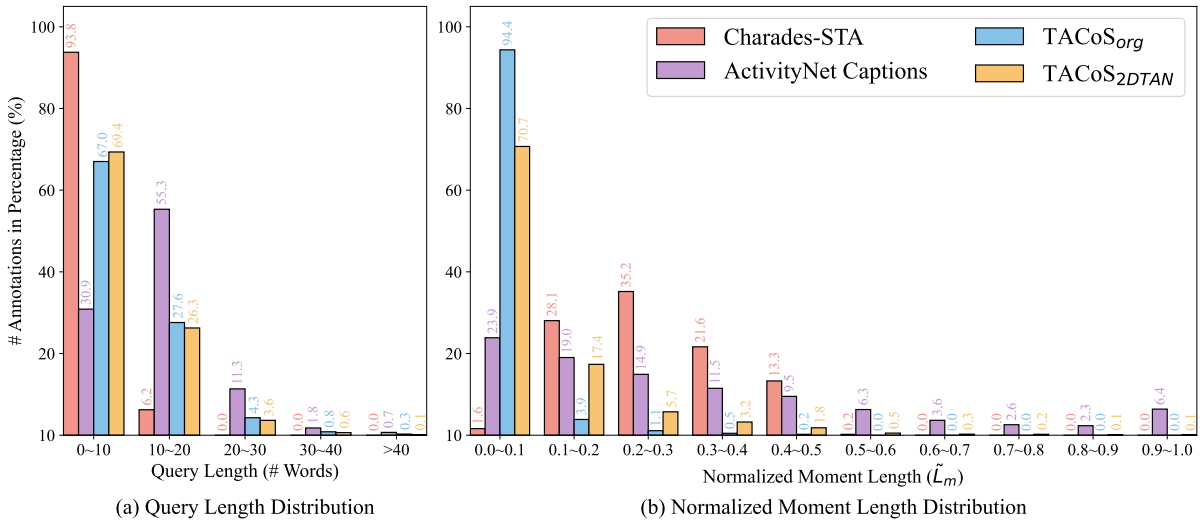
ActivityNet Captions. ActivityNet Captions dataset is developed by Krishna et al. [165] for dense video captioning task. However, the sentence-moment pairs in this dataset can naturally be adopted for the TSGV task. Specifically, the videos of ActivityNet Captions are taken from the ActivityNet [166] dataset, a human activity understanding benchmark. ActivityNet contains around 20k videos and provides samples from 203 activity classes, with an average of 137

untrimmed videos per class and 1.41 activity instances per video [166]. Since the official test set of ActivityNet Captions is withheld for competition, existing work mainly uses the official "val1" and/or "val2" development sets as test sets. Thus, statistics of ActivityNet Captions in Table 2.1 do not consider its official test set. In total, there are $14,926$ videos and $71,953$ annotations in ActivityNet Captions, where each video contains 4.82 temporally localized sentences on average. Average video and moment lengths are 117.60 and 37.14 seconds, respectively. The average query length is about 14.41 words.

TACoS. TACoS [167] is selected from the MPII Cooking Composite Activities dataset [168], which is originally developed for human activity recognition under specific scene, *i.e.,* composite cooking activities in lab kitchen. TACoS contains 127 videos, and each video is associated with two types of annotations: (1) fine-grained activity labels with temporal location, and (2) natural language descriptions with temporal locations. The natural language descriptions are from crowd-sourcing annotators, who describe the video content by sentences [167]. TACoS has $18,818$ moment-query pairs. Average video and moment lengths are 286.59 and 6.10 seconds, and the average query length is 10.05 words. Each video in TACoS contains 148.17 annotations on average. Here we name this dataset $\text{TACoS}_{\text{org}}$ in Table 2.1. A modified version $\text{TACoS}_{\text{2DTAN}}$ is made available by Zhang et al. [32]. $\text{TACoS}_{\text{2DTAN}}$ has $18,227$ annotations with $9,790$, $4,436$, and $4,001$ for training, validation, and test, respectively. On average, there are 143.52 annotations per video. The average moment length and query length after modification are 27.88 seconds and 9.42 words, respectively.

MAD. MAD [169] is a large-scale dataset containing mainstream movies. Compared to previous datasets, MAD aims to avoid hidden biases and provide accurate and unbiased annotations for TSGV. Instead of relying on crowd-sourced annotations, Soldan et al. [169] adopt a scalable data collection strategy. They transcribe the audio description track of a movie and remove sentences associated with the actor's speech, to obtain highly descriptive sentences that are grounded in long-form videos. MAD contains 650 movies with over $1,200$ hours of video length in total, where the training, validation, and test sets of MAD consist of 488, 50, and 112 movies, respectively. The average video duration is around 110 minutes. Each video in MAD is a full movie without pruning. MAD has $348,600$ queries with vocabulary size of $61,400$. The average query length is 12.7 words. The average length of the temporal moment in MAD is mere 4.1 seconds, making the localization process more challenging. Note the MAD dataset is not publicly available, hence we cannot provide detailed statistics in Table 2.1.

Dataset Analysis. Videos in aforementioned datasets may be from open domain or constrained in narrow and specific scenes (see Table 2.1). Open domain videos contain more diverse and



(a) Query Length Distribution

(b) Normalized Moment Length Distribution

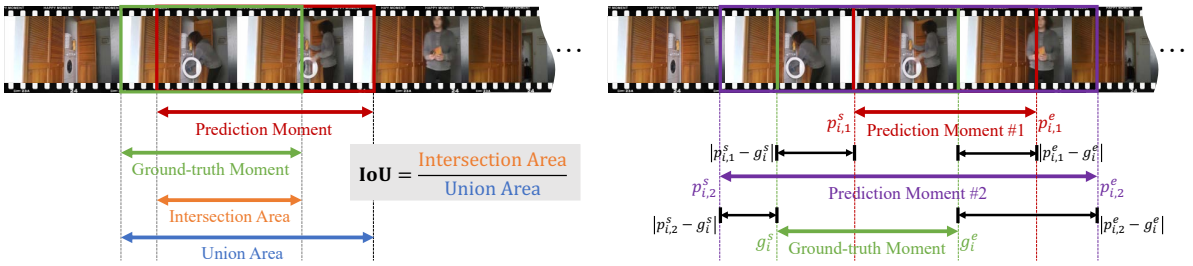
FIGURE 2.16: Statistics of the query length and normalized moment length ($\tilde{L}_m$) over the Charades-STA, ActivityNet Captions and TACoS benchmark datasets, where $\tilde{L}_m$ is computed as moment length divided by the corresponding video length.

complex activities, making them more challenging, but are closer to real-world scenarios. Although DiDeMo videos are from the open domain, answers in this dataset are fixed-length, *i.e.,* five-second. The fixed length considerably reduces the complexity of finding answers in DiDeMo. ActivityNet Captions and DiDeMo have much larger vocabulary sizes than Charades-STA and TACoS, suggesting that the former two datasets provide rich variations in language queries. From the perspective of query length (see Figure 2.16(a)), a large portion of queries in Charades-STA (93.8%) and TACoS ($> 67.0\%$) has fewer than 10 words. Query length distribution indicates that ActivityNet Captions contain more queries with complicated expressions. Figure 2.16(b) depicts the normalized moment length ($\tilde{L}_m$) distribution, computed against the length of its source video. A small $\tilde{L}_m$ means the moment is difficult to retrieve due to moment sparsity [35]. The figure shows more than 70.7% of the moments in TACoS has $\tilde{L}_m \leq 0.1$, while 70.1% moments in Charades-STA are in the range of $0.2 < \tilde{L}_m \leq 0.5$.

2.3.2 Evaluation Metrics

TSGV is generally evaluated by comparing predictions with ground-truth annotations. The widely used measures include: mean IoU (mIoU), $\langle$R@n, IoU@$\mu\rangle$, and $\langle$dR@n, IoU@$\mu\rangle$.

Intersection over Union (IoU) is a metric commonly used in object detection [170–172] for measuring similarity between two bounding boxes. Hence the standard IoU in object detection is defined on a two-dimensional spatial space. TSGV focuses on the temporal dimension only. Thus, temporal IoU is adopted to measure the similarity between the ground truth and predicted moments in TSGV, illustrated in Figure 2.17(a). IoU is computed as the ratio of intersection



(a) An illustration of temporal IoU.

(b) An illustration of dR@n,IoU@m.

FIGURE 2.17: The temporal intersection over union (IoU), and the discounted-R@n,IoU@m (dR@n,IoU@m), where p_i^s and p_i^e are start and timestamps of predicted moments, $g^{s/e}$ is start/end timestamp of ground-truth moment, and $|\cdot|$ denotes the absolute operation.

area over union area between two moments, in the range of 0.0 to 1.0. A larger IoU means the two moments match better, and IoU $= 1.0$ denotes an exact match. The mIoU metric is the average temporal IoUs among all annotations in the test set. Mathematically, mIoU is defined as:

$$\text{mIoU} = \frac{1}{N_q}\sum_{i=1}^{N_q}\text{IoU}_i, \tag{2.15}$$

where N_q denotes the total number of annotations, and IoU_i is the IoU value of i-th sample.

The mIoU is computed based on the single top-ranked prediction for each query. However, given a query, the top-ranked prediction by a TSGV model may not always have the best match with ground truth. It is reasonable to relax the evaluation by considering top-n retrieved moments for each query. The $\langle$R@n, IoU@$\mu\rangle$ [173] is the percentage of queries, having at least one result whose temporal IoU with ground-truth is larger than μ among the top-n retrieved moments. For query q_i, among its top-n retrieved moments, if there exists at least one moment whose IoU with ground-truth is larger than μ, then q_i is considered as positive, denoted by $r(n,\mu,q_i)=1$. Otherwise, $r(n,\mu,q_i)=0$. Thus, $\langle$R@n, IoU@$\mu\rangle$ is calculated as:

$$\text{R@}n\text{,IoU@}\mu = \frac{1}{N_q}\sum_{i=1}^{N_q} r(n,\mu,q_i). \tag{2.16}$$

Yuan et al. [174] reveal that $\langle$R@n, IoU@$\mu\rangle$ is unreliable on small IoU thresholds. A method tends to generate long predictions if a substantial proportion of ground-truth moments are long in a dataset. In this way, the method increases its chance of correct prediction under small

IoU thresholds. Discounted-R@n, IoU@μ $\langle$dR@n, IoU@$\mu\rangle$ is proposed to alleviate this problem [174]. This new measure leverages "temporal distance" between the predicted and ground-truth moments to discount $r(n, \mu, q_i)$ value. $\langle$dR@n, IoU@$\mu\rangle$ is calculated as:

$$\text{dR@}n\text{,IoU@}\mu = \frac{1}{N_q}\sum_{i=1}^{N_q} r(n, \mu, q_i) \cdot \alpha_i^s \cdot \alpha_i^e, \tag{2.17}$$

where the discounted ratio $\alpha_i^* = 1 - |p_i^* - g_i^*|$, $* \in \{s, e\}$. $|p_i^* - g_i^*|$ is the absolute distance between the boundaries of predicted and ground-truth moments (see Figure 2.17(b)). Note both p_i^* and g_i^* are normalized in 0.0 to 1.0 by dividing the corresponding whole video length. If the predicted moment exactly matches ground-truth, then discounted ratio $\alpha_i^* = 1$, and the metric degrades to $\langle$R@n, IoU@$\mu\rangle$. Otherwise, even if IoU threshold is met, $r(n, \mu, q_i)$ is discounted by α_i^*, which helps to restrain over-long predictions.

In Figure 2.17(b), $(p_{i,1}^s, p_{i,1}^e)$ and $(p_{i,2}^s, p_{i,2}^e)$ are two example predicted moments of query q_i, and (g_i^s, g_i^e) is ground-truth moment. Suppose both Predictions 1 and 2 in Figure 2.17(b) have the same IoU value which satisfies IoU $\geq \mu$, ($\mu \leq 0.5$ here), $\langle$dR@n, IoU@$\mu\rangle$ penalizes more on Prediction 2 since its temporal boundaries are farther from ground-truth. With respect to $\langle$R@n,IoU@$\mu\rangle$ and $\langle$dR@n,IoU@$\mu\rangle$ metrics, community is habituated to set $n \in \{1, 5, 10\}$ and $\mu \in \{0.3, 0.5, 0.7\}$.

Chapter 3

Span-based Question Answering for TSGV

3.1 Introduction

TSGV aims to retrieve a matching span, *i.e.,* a temporal moment, from a given video that semantically corresponds to a given language query. Prior solutions primarily treat TSGV as a ranking task, which rely on various proposal generation modules and solve TSGV with multimodal matching architecture to retrieve the best matching video segment for the given language query. Some work explores to model cross-interactions between video and query, and to regress the temporal locations of target moment directly. There are also studies to formulate TSGV as a sequence decision making problem and to solve it by reinforcement learning. Given an untrimmed video and a language query, the key to correctly retrieve the target moment is the semantic understanding of both video and language query, as well as the cross-modal reasoning between them. In this chapter[1], we investigate a novel TSGV framework that utilizes the concepts of span-based question answering in NLP to perform moment localization.

The essence of TSGV is to search for a video moment as the answer to a given language query from an untrimmed video. Before the untrimmed video can be used for TSGV model training or inference, it requires to be converted into a sequence of visual features through pre-trained 3D-CovNet. Similarly, in span-based question answering task, the text passage also need to be transformed into a sequence of word embeddings. As illustrated in Figure 1.2, by treating the video as a text passage, and the target moment as the answer span, TSGV

[1]This chapter is published as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "Span-based Localizing Network for Natural Language Video Localization". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6543–6554, Online, 2020 [34].

shares significant similarities with span-based question answering (QA) task. That is, the span-based QA framework [15, 16, 175] can be adopted for TSGV theoretically. Inspired by the similarities between span-based QA and TSGV tasks, we attempt to solve the TSGV problem with a multimodal span-based QA approach.

Although TSGV and span-based QA tasks share significant similarities, there are still two main differences between them:

- First, video is continuous and causal relations between video events are usually adjacent. Natural language, on the other hand, is inconsecutive and the words in a sentence demonstrate syntactic structure. For instance, changes between adjacent video frames are usually very small, while the adjacent word tokens may carry distinctive meanings. As the result, many events in a video are directly correlated and can even cause one another [165]. The causalities between word spans or sentences are usually indirect and can be far apart in general.

- Second, compared to word spans in text, human is insensitive to small shifting between video frames. In other words, small offsets between video frames do not affect the understanding of video content, but the differences of a few words or even one word could change the meaning of a sentence.

Thus, we will address two research questions: *Whether the TSGV task can be formulated as span-based question answering, and be solved with the standard span-based QA framework?* and *How to effectively address the differences between span-based QA and TSGV tasks?*

Our Approach. To answer the first question, we directly apply a standard span-based QA framework with slightly modifications, termed VSLBase, to solve the TSGV task. Specifically, visual features are analogous to that of text passage; the target moment is regarded as the answer span. VSLBase is trained to predict the start and end boundaries of the answer span.

Since VSLBase does not address the two aforementioned major differences between video and natural language. We then propose an improved version named Video Span Localizing Network, *i.e.,* VSLNet. VSLNet introduces a Query-Guided Highlighting (QGH) strategy in addition to VSLBase. Here, we regard the target moment and its adjacent contexts as foreground, while the rest as background, *i.e.,* foreground covers a slightly longer span than the answer span. With QGH, VSLNet is guided to search for the target moment within a highlighted region. Through region highlighting, VSLNet well addresses the two differences. First, the longer region provides additional contexts for locating answer span due to the continuous nature of video content. Second, the highlighted region helps the network to focus on subtle differences between video frames, because the search space is reduced compared to the full video.

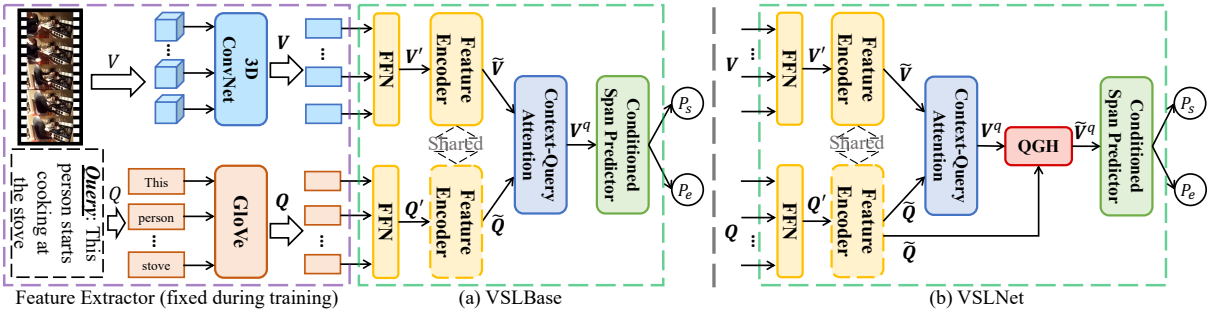


FIGURE 3.1: An overview of the proposed architecture for TSGV. The visual and textual feature extractors are fixed during training. Figure (a) depicts the adoption of standard span-based QA framework, *i.e.,* VSLBase. Figure (b) shows the structure of VSLNet.

We study the performance of our approaches on three benchmark datasets. The results show that adopting span-based QA framework is suitable for TSGV. With a simple network architecture, VSLBase delivers comparable performance to strong baselines. In addition, VSLNet further boosts the performance and achieves the best among all evaluated methods.

3.2 VSLNet Framework

Let $V = \{f_t\}_{t=1}^{T}$ be the untrimmed video and $Q = \{q_j\}_{j=1}^{m}$ be the language query, where T and m are number of frames and words, respectively. τ_s and τ_e represent start and end timestamps of temporal moment, *i.e.,* answer span. To address TSGV from the perspective of span-based question answering (QA), its data should be transformed into a set of SQuAD style triples $(Context, Question, Answer)$ [176]. For each video, we extract its video snippet feature sequence $\boldsymbol{V} = \{\boldsymbol{v}_i\}_{i=1}^{n}$ through a pre-trained 3D ConvNet [25], where n is number of extracted features. Here, $\boldsymbol{V}$ can be regarded as sequence of word embeddings for a text passage with n tokens. Similar to word embeddings, each feature $\boldsymbol{v}_i$ here is a video feature vector.

Since span-based QA aims to predict start and end boundaries of an answer span, the start/end time of a video sequence needs to be mapped to the corresponding boundaries in the visual feature sequence $\boldsymbol{V}$. Suppose the video duration is $\mathcal{T}$, the start (end) span index is calculated by $a_{s(e)} = \langle \tau_{s(e)}/\mathcal{T} \times n \rangle$, where $\langle \cdot \rangle$ denotes the rounding operator. During the inference, the predicted span boundary can be easily converted to the corresponding time via $\tau_{s(e)} = a_{s(e)}/n \times \mathcal{T}$. After transforming moment annotations in TSGV dataset, we obtain a set of $(\boldsymbol{V}, Q, \boldsymbol{A})$ triples. Visual features $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$ act as the passage with n tokens; $Q = [q_1, q_2, \ldots, q_m]$ is the query with m tokens, and the answer $\boldsymbol{A} = [\boldsymbol{v}_{a_s}, \boldsymbol{v}_{a_s+1}, \ldots, \boldsymbol{v}_{a_e}]$ corresponds to a piece in the passage. Then, the TSGV task becomes to find the correct start and end boundaries of the answer span, a_s and a_e.

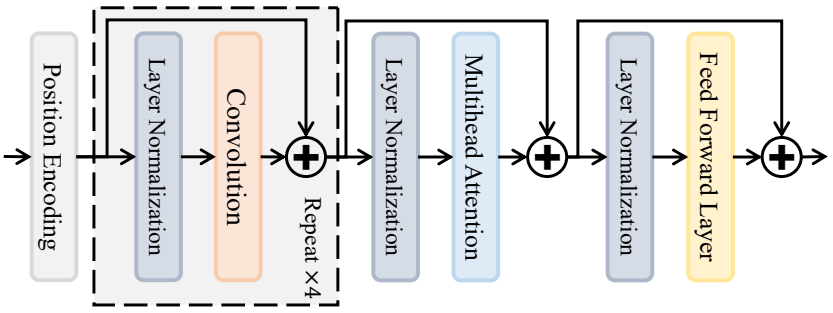


FIGURE 3.2: The structure of Feature Encoder.

3.2.1 Feature Encoder

We already have visual features $\boldsymbol{V} = \{\boldsymbol{v}_i\}_{i=1}^{n} \in \mathbb{R}^{n \times d_v}$. Word embeddings of a text query Q, $\boldsymbol{Q} = \{\boldsymbol{q}_j\}_{j=1}^{m} \in \mathbb{R}^{m \times d_q}$, are easily obtainable, *e.g.,* GloVe [26]. We first project them into the same dimension d, $\boldsymbol{V'} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{Q'} \in \mathbb{R}^{m \times d}$, by two linear layers (see Figure 3.1(a)). Then we build the feature encoder with a simplified version of the embedding encoder layer in QANet [90]. That is, instead of applying a stack of multiple encoder blocks, we use only one encoder block. As shown in Figure 3.2, this encoder block consists of four convolution layers, followed by a multi-head attention layer [177]. A feed-forward layer is used to produce the output. Layer normalization [178] and residual connection [179] are applied to each layer. The encoded visual features and word embeddings are as follows:

$$\begin{aligned} \widetilde{\boldsymbol{V}} &= \texttt{FeatureEncoder}(\boldsymbol{V'}), \\ \widetilde{\boldsymbol{Q}} &= \texttt{FeatureEncoder}(\boldsymbol{Q'}), \end{aligned} \tag{3.1}$$

where the parameters of feature encoder are shared by visual features and word embeddings.

3.2.2 Context-Query Attention

After feature encoding, we use context-query attention (CQA) [16, 90, 180] to capture the cross-modal interactions between visual and textural features. Given two inputs, the goal of CQA is to encode the relationships between each pair of the elements in the two inputs. Specifically, CQA first calculates the similarity scores, $\mathcal{S} \in \mathbb{R}^{n \times m}$, between each visual feature and query feature as:

$$\mathcal{S} = \widetilde{\boldsymbol{V}}^{\top} \cdot \boldsymbol{W} \cdot \widetilde{\boldsymbol{Q}} \in \mathbb{R}^{n \times m}, \tag{3.2}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ and $\mathcal{S} \in \mathbb{R}^{n\times m}$. Then context-to-query ($\mathcal{A}$) attention and query-to-context ($\mathcal{B}$) attention weights are computed as:

$$\begin{aligned}\mathcal{A} &= \mathcal{S}_r \cdot \widetilde{\boldsymbol{Q}} \in \mathbb{R}^{n\times d},\\ \mathcal{B} &= \mathcal{S}_r \cdot \mathcal{S}_c^T \cdot \widetilde{\boldsymbol{V}} \in \mathbb{R}^{n\times d},\end{aligned} \tag{3.3}$$

where $\mathcal{S}_r$ and $\mathcal{S}_c$ are the row- and column-wise normalization of $\mathcal{S}$ by Softmax function, respectively. Finally, the output of context-query attention is calculated as:

$$\boldsymbol{V}^q = \texttt{FFN}\big([\widetilde{\boldsymbol{V}}; \mathcal{A}; \widetilde{\boldsymbol{V}} \odot \mathcal{A}; \widetilde{\boldsymbol{V}} \odot \mathcal{B}]\big), \tag{3.4}$$

where $\boldsymbol{V}^q \in \mathbb{R}^{n\times d}$; FFN is a single feed-forward layer; $\odot$ denotes element-wise multiplication; and ";" represents concatenation operation. In this way, the information of $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{m\times d}$ is properly fused into $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{n\times d}$, resulting in the multimodal representations $\boldsymbol{V}^q \in \mathbb{R}^{n\times d}$.

3.2.3 Conditioned Span Predictor

We construct a conditioned span predictor by using two unidirectional LSTMs and two feed-forward layers, inspired by Ghosh et al. [89]. The main difference between ours and Ghosh et al. [89] is that we use unidirectional LSTM instead of bidirectional LSTM. We observe that unidirectional LSTM (UniLSTM) shows similar performance with fewer parameters and higher efficiency. The two LSTMs are stacked so that the LSTM of end boundary can be conditioned on that of start boundary. Then the hidden states of the two LSTMs are fed into the corresponding feed-forward layers to compute the start and end scores:

$$\begin{aligned}\boldsymbol{h}_t^s &= \texttt{UniLSTM}_{\text{start}}(\boldsymbol{v}_t^q, \boldsymbol{h}_{t-1}^s),\\ \boldsymbol{h}_t^e &= \texttt{UniLSTM}_{\text{end}}(\boldsymbol{h}_t^s, \boldsymbol{h}_{t-1}^e),\\ \boldsymbol{S}_t^s &= \boldsymbol{W}_s \times ([\boldsymbol{h}_t^s; \boldsymbol{v}_t^q]) + \boldsymbol{b}_s,\\ \boldsymbol{S}_t^e &= \boldsymbol{W}_e \times ([\boldsymbol{h}_t^e; \boldsymbol{v}_t^q]) + \boldsymbol{b}_e.\end{aligned} \tag{3.5}$$

Here, $\boldsymbol{S}_t^s$ and $\boldsymbol{S}_t^e$ denote the scores of start and end boundaries at position t; $\boldsymbol{v}_t^q$ represents the t-th feature in $\boldsymbol{V}^q$. $\boldsymbol{W}_{s/e}$ and $\boldsymbol{b}_{s/e}$ denote the weight matrix and bias of the start/end feed-forward layer, respectively. Then, the probability distributions of start and end boundaries are computed by $P_s = \text{SoftMax}(\boldsymbol{S}^s) \in \mathbb{R}^n$ and $P_e = \text{SoftMax}(\boldsymbol{S}^e) \in \mathbb{R}^n$, and the training objective is defined as:

$$\mathcal{L}_{\text{span}} = \frac{1}{2}\big[f_{\text{CE}}(P_s, Y_s) + f_{\text{CE}}(P_e, Y_e)\big], \tag{3.6}$$

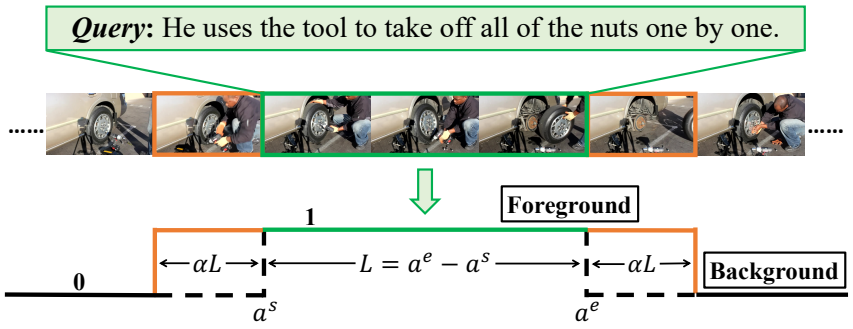


FIGURE 3.3: An illustration of foreground and background of visual features. α is the ratio of foreground extension.

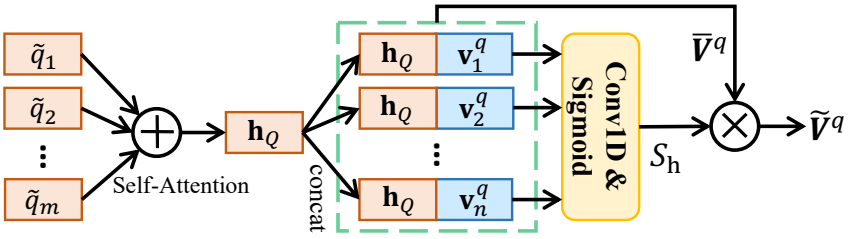


FIGURE 3.4: The structure of Query-Guided Highlighting.

where f_{CE} represents cross-entropy loss function; Y_s and Y_e are the labels for the start (a_s) and end (a_e) boundaries, respectively. During inference, the predicted answer span $(\hat{a}_s, \hat{a}_e)$ of a query is generated by maximizing the joint probability of start and end boundaries by:

$$\begin{aligned} \texttt{span}(\hat{a}_s, \hat{a}_e) &= \arg\max_{\hat{a}_s, \hat{a}_e} P_s(\hat{a}_s) P_e(\hat{a}_e), \\ &\text{s.t. } 0 \leq \hat{a}_s \leq \hat{a}_e < n. \end{aligned} \tag{3.7}$$

We have completed the VSLBase architecture (see Figure 3.1(a)). VSLNet is built on top of VSLBase with query-guided highlighting (QGH), to be detailed next.

3.2.4 Query-Guided Highlighting

A Query-Guided Highlighting (QGH) strategy is introduced in VSLNet, to address the major differences between text span-based QA and TSGV tasks, as shown in Figure 3.1(b). With QGH strategy, we consider the target moment as the foreground, and the rest as background, illustrated in Figure 3.3. The target moment, which is aligned with the language query, starts from a_s and ends at a_e with length $L = a_e - a_s$. QGH extends the boundaries of the foreground to cover its antecedent and consequent video contents, where the extension ratio is controlled by

a hyperparameter α. As aforementioned in Introduction, the extended boundary could potentially cover additional contexts and also help the network to focus on subtle differences between video frames.

By assigning 1 to foreground and 0 to background, we obtain a sequence of 0-1, denoted by Y_{h}. QGH is a binary classification module to predict the confidence a visual feature belongs to foreground or background. The structure of QGH is shown in Figure 3.4. We first encode word features $\widetilde{\boldsymbol{Q}}$ into sentence representation (denoted by $\boldsymbol{h}_Q$), with self-attention mechanism [13]. Then $\boldsymbol{h}_Q$ is concatenated with each feature in $\boldsymbol{V}^q$ as $\bar{\boldsymbol{V}}^q = [\bar{\boldsymbol{v}}_1^q, \ldots, \bar{\boldsymbol{v}}_n^q]$, where $\bar{\boldsymbol{v}}_i^q = [\boldsymbol{v}_i^q; \boldsymbol{h}_Q]$. The highlighting score is computed as:

$$\mathcal{S}_{\text{h}} = \sigma\big(\texttt{Conv1D}(\bar{\boldsymbol{V}}^q)\big), \tag{3.8}$$

where σ denotes Sigmoid activation; $\mathcal{S}_{\text{h}} \in \mathbb{R}^n$. The highlighted features are calculated by:

$$\widetilde{\boldsymbol{V}}^q = \mathcal{S}_{\text{h}} \cdot \bar{\boldsymbol{V}}^q. \tag{3.9}$$

Accordingly, feature $\boldsymbol{V}^q$ in Equation (3.5) is replaced by $\widetilde{\boldsymbol{V}}^q$ in VSLNet to compute $\mathcal{L}_{\text{span}}$. The loss function of query-guided highlighting is formulated as:

$$\mathcal{L}_{\text{QGH}} = f_{\text{CE}}(S_{\text{h}}, Y_{\text{h}}), \tag{3.10}$$

and VSLNet is trained in an end-to-end manner by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}}. \tag{3.11}$$

3.3 Experiments

Our experiments is designed to study the effectiveness of the proposed VSLBase and VSLNet frameworks, and to evaluate their performance in comparison to other state-of-the-art methods. We utilize the Charades-STA, ActivityNet Captions and TACoS$_{\text{org}}$ datasets to report the model's performance and analyze its behaviors.

3.3.1 Experimental Settings

Evaluation Metrics. We adopt "R@n,IoU@μ" and "mIoU" as the evaluation metrics, following the common evaluation settings [27, 61, 88]. Recall the R@n,IoU@μ denotes the percentage of language queries having at least one result whose Intersection over Union (IoU) with

TABLE 3.1: "R@1, IoU@μ" and "mIoU" results (%) compared with the state-of-the-art on Charades-STA.

Model	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
3D ConvNet without fine-tuning as visual feature extractor				
CTRL	-	23.63	8.89	-
ACL-K	-	30.48	12.20	-
QSPN	54.70	35.60	15.80	-
SAP	-	27.42	13.36	-
SM-RL	-	24.36	11.17	-
RWM-RL	-	36.70	-	-
MAN	-	46.53	22.72	-
DEBUG	54.95	37.39	17.69	36.34
VSLBase	61.72	40.97	24.14	42.11
VSLNet	**64.30**	**47.31**	**30.19**	**45.15**
3D ConvNet with fine-tuning on Charades dataset				
ExCL	65.10	44.10	23.30	-
VSLBase	68.06	50.23	30.16	47.15
VSLNet	**70.46**	**54.19**	**35.22**	**50.02**

TABLE 3.2: "R@1, IoU@μ" and "mIoU" results (%) compared with the state-of-the-arts on ActivityNet Captions.

Model	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
TGN	45.51	28.47	-	-
ABLR	55.67	36.79	-	36.99
RWM-RL	-	36.90	-	-
QSPN	45.30	27.70	13.60	-
ExCL*	63.00	**43.60**	24.10	-
DEBUG	55.91	39.72	-	39.51
VSLBase	58.18	39.52	23.21	40.56
VSLNet	**63.16**	43.22	**26.16**	**43.19**

TABLE 3.3: "R@1, IoU@μ" and "mIoU" results (%) compared with the state-of-the-arts on TACoS$_{\text{org}}$.

Model	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
CTRL	18.32	13.30	-	-
TGN	21.77	18.90	-	-
ACRN	19.52	14.62	-	-
ABLR	19.50	9.40	-	13.40
ACL-K	24.17	20.01	-	-
L-Net	-	-	-	13.41
DEBUG	23.45	11.72	-	16.03
VSLBase	23.59	20.40	16.65	20.10
VSLNet	**29.61**	**24.27**	**20.03**	**24.11**

ground truth is larger than μ in top-n retrieved moments. The mIoU is the average IoU over all testing samples. In our experiments, we use $n = 1$ and $\mu \in \{0.3, 0.5, 0.7\}$.

Implementation. For language query Q, we use 300d GloVe [26] vectors to initialize each lowercase word; the word embeddings are fixed during training. For untrimmed video V, we downsample frames and extract RGB visual features using the 3D ConvNet which was pre-trained on Kinetics dataset [25]. We set the dimension of all the hidden layers in the model as 128; the kernel size of convolution layer is 7; the head size of multi-head attention in transformer block is 8. For all datasets, the model is trained for 100 epochs with batch size of 16

and early stopping strategy. Parameter optimization is performed by Adam [181] with learning rate of 0.0001, linear decay of learning rate and gradient clipping of 1.0. Dropout [182] of 0.2 is applied to prevent overfitting.

Baselines. We evaluate our VSLBase and VSLNet with the following TSGV baselines:

- Sliding window-based methods: CTRL [27], ACRN [61], and ACL-K [28]
- Proposal generated methods: QSPN [29] and SAP [49].
- Standard anchor-based methods: TGN [30] and MAN [54].
- Reinforcement learning-based methods: SM-RL [124] and RWM-RL [120].
- Regression-based methods: ExCL [89], ABLR [88], and DEBUG [91].
- Span-based methods: L-Net [103].

3.3.2 Overall Performance

The results of the VSLBase, VSLNet and the state-of-the-art baselines on benchmark datasets are reported in Table 3.1, 3.2 and 3.3, respectively. In all result tables, the scores of compared methods are reported in the corresponding works. Best results are in **bold** and second best underlined.

The results on Charades-STA are summarized in Table 3.1. For fair comparison with ExCL, we follow the same setting in ExCL [89] to use the 3D ConvNet fine-tuned on Charades dataset as visual feature extractor. Observed that VSLNet significantly outperforms all baselines by a large margin over all metrics. It is worth noting that the performance improvements of VSLNet are more significant under more strict metrics. For instance, VSLNet achieves 7.47% improvement in $\mu = 0.7$ versus 0.78% in $\mu = 0.5$, compared to MAN. Without query-guided highlighting, VSLBase outperforms all compared baselines over $\mu = 0.7$, which shows adopting span-based QA framework is promising for TSGV. Moreover, VSLNet benefits from visual feature fine-tuning, and achieves state-of-the-art results on this dataset.

Table 3.2 summarizes the results on ActivityNet Caption dataset. Note that this dataset requires YouTube clips to be downloaded online. We have $1,309$ missing videos, while ExCL reports $3,370$ missing videos. Strictly speaking, the results reported in this table are not directly comparable. Despite that, VSLNet is superior to ExCL with 2.06% and 0.16% absolute improvements over $\mu = 0.7$ and $\mu = 0.3$, respectively. Meanwhile, VSLNet surpasses other baselines.

TABLE 3.4: Comparison between models with alternative modules in VSLBase on the Charades-STA dataset.

Module	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
BiLSTM + CAT	61.18	43.04	26.42	42.83
CMF + CAT	63.49	44.87	27.07	44.01
BiLSTM + CQA	65.08	46.94	28.55	45.18
CMF + CQA	68.06	50.23	30.16	47.15

TABLE 3.5: Performance gains (%) of different modules over "R@1, IoU@0.7" on Charades-STA dataset.

Module	CAT	CQA	Δ
BiLSTM	26.42	28.55	+2.13
CMF	27.07	30.16	+3.09
Δ	+0.65	+1.61	-

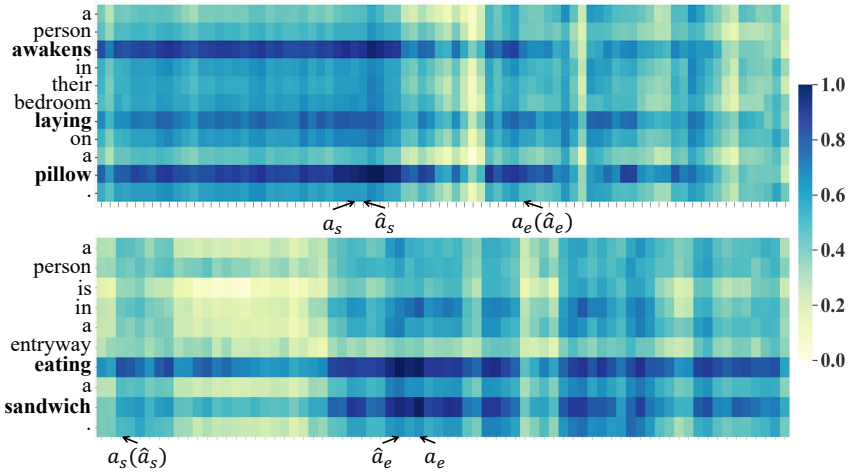


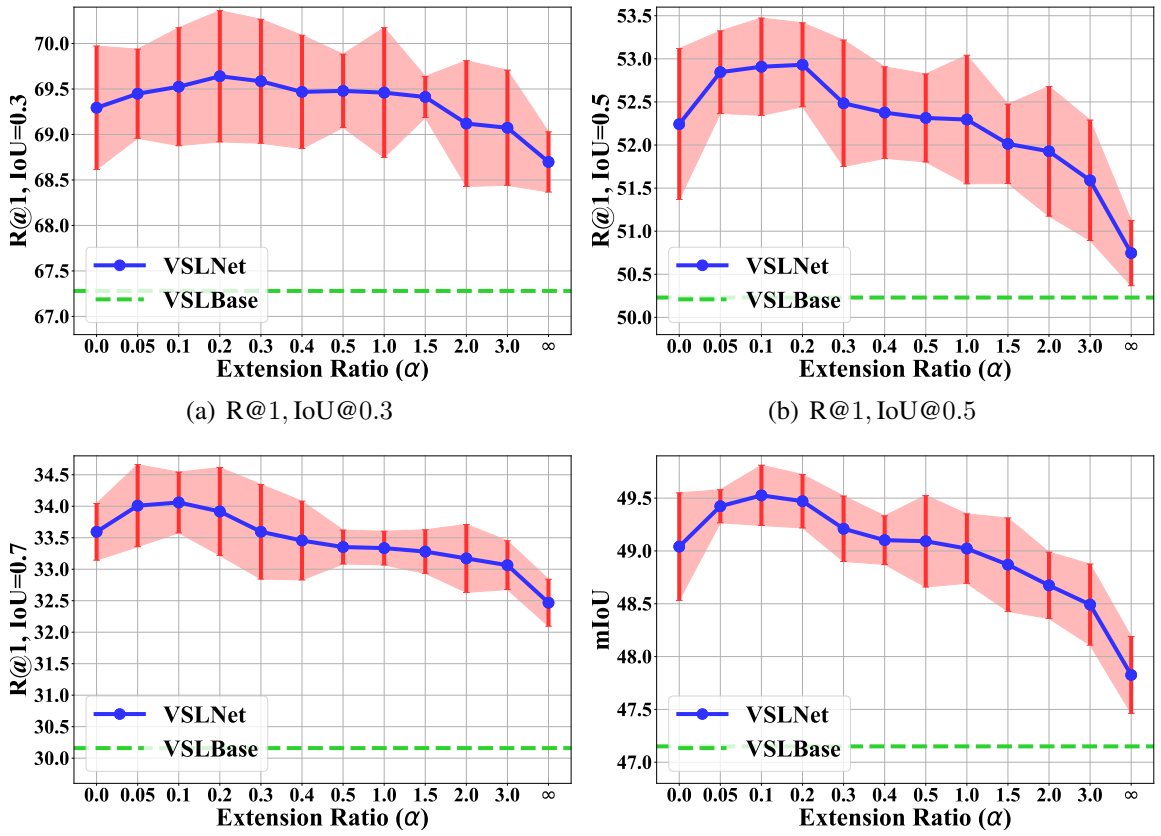
FIGURE 3.5: Similarity scores, $\mathcal{S}$, between visual and language features in the context-query attention. a_s/a_e denote the start/end boundaries of ground truth video moment, $\hat{a}_s/\hat{a}_e$ denote the start/end boundaries of predicted target moment.

Similar observations hold on TACoS$_{\text{org}}$ dataset. Reported in Table 3.3, VSLNet achieves new state-of-the-art performance on all evaluation metrics. Without QGH, VSLBase shows comparable performance with baselines.

3.3.3 Ablation Study

To reveal how the proposed VSLBase and VSLNet work, we conduct ablative experiments to analyze the importance of feature encoder and context-query attention in our approach. We also investigate the impact of extension ratio α (see Figure 3.3) in query-guided highlighting (QGH). Finally we visually show the effectiveness of QGH in VSLNet, and discuss the weaknesses of VSLBase and VSLNet.

Module Analysis. We study the effectiveness of our feature encoder and context-query attention (CQA) by replacing them with other modules. Specifically, we use bidirectional LSTM (BiLSTM) as an alternative feature encoder. For context-query attention, we replace it with a



(a) R@1, IoU@0.3 (b) R@1, IoU@0.5

(c) R@1, IoU@0.7 (d) mIoU

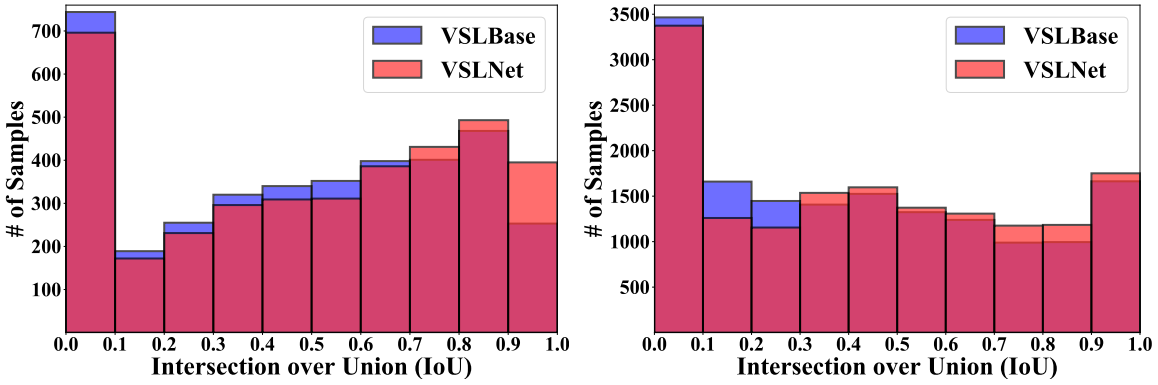
FIGURE 3.6: Analysis of the impact of extension ratio α in Query-Guided Highlighting on the Charades-STA dataset.

simple method (named CAT) which concatenates each visual feature with a max-pooled query feature.

Recall that our feature encoder consists of Convolution + Multi-head attention + Feed-forward layers (see Section 3.2.1), we name it CMF. With the alternatives, we now have 4 combinations, listed in Table 3.4. Observing from the results, CMF shows stable superiority over CAT on all metrics regardless of other modules; CQA surpasses CAT whichever feature encoder is used. This study indicates that CMF and CQA are more effective.

Table 3.5 reports performance gains of different modules over "R@1, IoU@0.7" metric. The results show that replacing CAT with CQA leads to larger improvements, compared to replacing BiLSTM with CMF. This observation suggests CQA plays a more important role in our model. Specifically, keeping CQA, the absolute gain is 1.61% by replacing the encoder module. Keeping CMF, the gain of replacing the attention module is 3.09%.

Figure 3.5 visualizes the matrix of similarity score between visual and language features in the context-query attention (CQA) module ($\mathcal{S} \in \mathbb{R}^{n \times m}$ in Section 3.2.2). This figure shows visual features that are more relevant to the verbs and their objects in the query sentence. For



(a) Charades-STA (b) ActivityNet Captions

FIGURE 3.7: Histograms of the number of predicted results on test set under different IoUs, on the Charades-STA and ActivityNet Captions datasets.

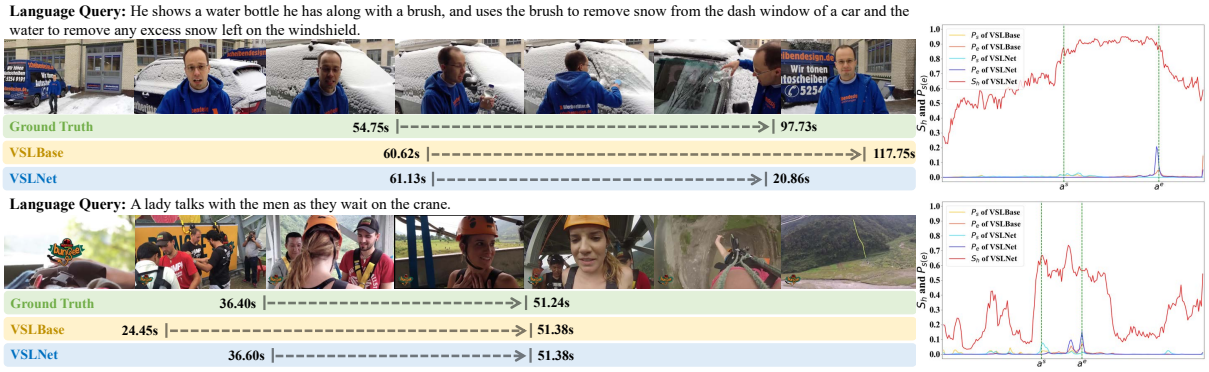
example, the similarity scores between visual features and "*eating*" (or "*sandwich*") are higher than that of other words. We believe that verbs and their objects are more likely to be used to describe video activities. Our observation is consistent with Ge et al. [28], where *verb-object* pairs are extracted as semantic activity concepts. In contrast, these concepts are automatically captured by the CQA module in our method.

The Impact of Extension Ratio in QGH. We now study the impact of extension ratio α in the query-guided highlighting module on the Charades-STA dataset. We evaluated 12 different values of α from 0.0 to ∞ in experiments. 0.0 represents no answer span extension, and ∞ means that the entire video is regarded as foreground.

The results for various α's are plotted in Figure 3.6. It shows that query-guided highlighting consistently contributes to performance improvements, regardless of α values, *i.e.,* from 0 to ∞. Along with α raises, the performance of VSLNet first increases and then gradually decreases. The optimal performance appears between $\alpha = 0.05$ and 0.2 on all metrics. Note that, when $\alpha = \infty$, which is equivalent to no region is highlighted as a coarse region to locate the target moment, VSLNet remains better than VSLBase. As shown in Figure 3.4, when $\alpha = \infty$, QGH effectively becomes a straightforward concatenation of sentence representation with each of the visual features. The resultant feature remains helpful for capturing semantic correlations between vision and language. In this sense, this function can be regarded as an approximation or simulation of the traditional multimodal matching strategy [27, 46, 61].

Qualitative Analysis. Figure 3.7 shows the histograms of predicted results on test sets of Charades-STA and ActivityNet Caption datasets. Results show that VSLNet beats VSLBase by having more samples in the high IoU ranges, *e.g.,* IoU ≥ 0.7 on the Charades-STA dataset. More predicted results of VSLNet are distributed in the high IoU ranges for the ActivityNet

(a) Two example cases on the Charades-STA dataset



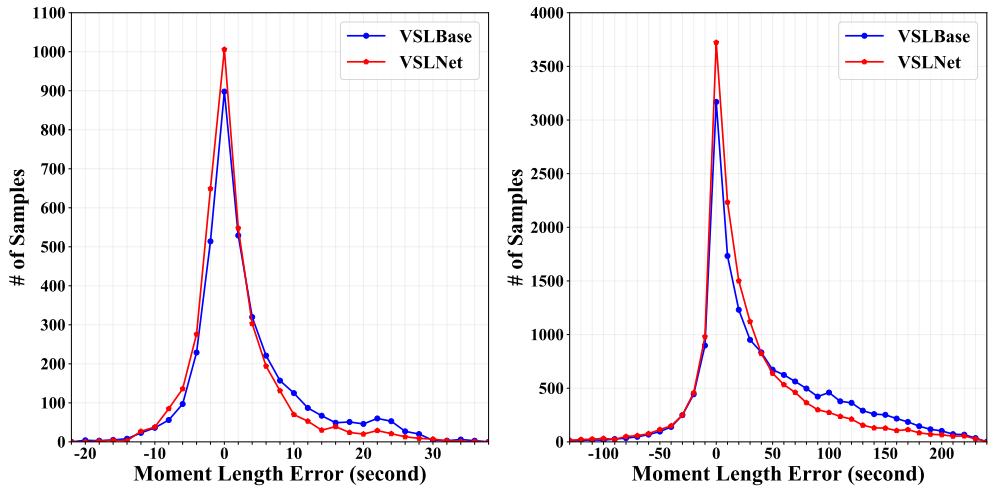
(b) Two example cases on the ActivityNet Caption dataset

FIGURE 3.8: Visualization of predictions by VSLBase and VSLNet. Figures on the left depict the localized results by the two models. Figures on the right show probability distributions of start/end boundaries and highlighting scores.

Captions dataset. This result demonstrates the effectiveness of the query-guided highlighting (QGH) strategy.

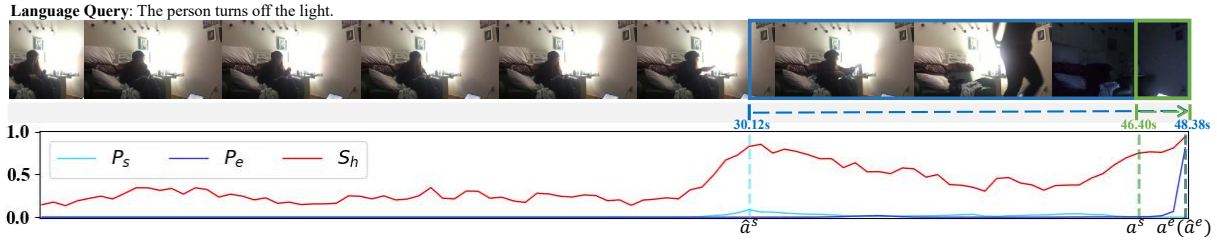
We also show two examples in Figures 3.8(a) and 3.8(b) from the Charades-STA and the ActivityNet Caption datasets, respectively. From the two figures, the localized moments by VSLNet are closer to the ground truth than that by VSLBase. Meanwhile, the start and end boundaries predicted by the VSLNet are roughly constrained in the highlighted regions S_{h}, computed by QGH.

We further study the error patterns of predicted moment lengths, as shown in Figure 3.9. The differences between moment lengths of ground truths and predicted results are measured. A positive length difference means the predicted moment is longer than the corresponding ground truth, while a negative means shorter. Figure 3.9 shows that VSLBase tends to predict longer moments, *e.g.,* more samples with length error larger than 4 seconds in Charades-STA or 30 seconds in ActivityNet. On the contrary, constrained by QGH, VSLNet tends to predict shorter moments, *e.g.,* more samples with length error smaller than -4 seconds in Charades-STA or -20 seconds in ActivityNet Caption. This observation is helpful for future research on

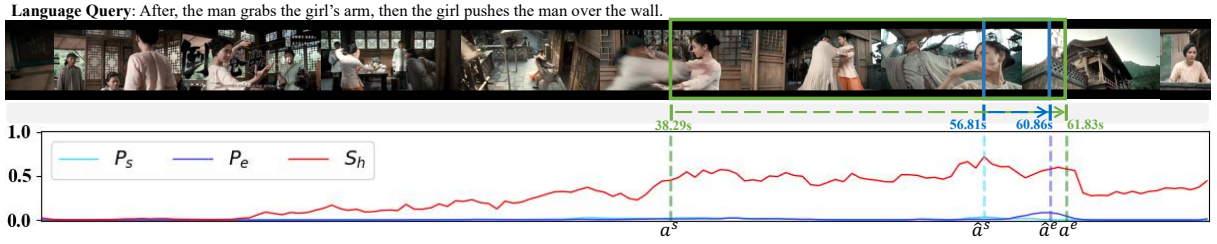


(a) Charades-STA (b) ActivityNet Captions

FIGURE 3.9: Plots of moment length errors in seconds between ground truths and results predicted by VSLBase and VSLNet, respectively.



(a) A failure case on the Charades-STA dataset with IoU@0.11.



(b) A failure case on the ActivityNet Captions dataset with IoU@0.17.

FIGURE 3.10: Two failure examples predicted by VSLNet, a^s/a^e denote the start/end boundaries of ground truth video moment, $\hat{a}^s/\hat{a}^e$ denote the start/end boundaries of predicted target moment.

adopting a span-based QA framework for TSGV.

In addition, we also examine failure cases (with IoU predicted by VSLNet lower than 0.2) shown in Figure 3.10. In the first case, as illustrated by Figure 3.10(a), we observe an action that a person turns towards the lamp and places an item there. The QGH falsely predicts the action as the beginning of the moment "turns off the light". The second failure case involves multiple actions in a query, as shown in Figure 3.10(b). QGH successfully highlights the correct region

by capturing the temporal information of two different action descriptions in the given query. However, it assigns "pushes" with a higher confidence score than "grabs". Thus, VSLNet only captures the region corresponding to the "pushes" action, due to its confidence score.

3.4 Summary

To this end, we have studied the idea of utilizing span-based question answering to perform TSGV in an end-to-end manner. We show that this approach is effective and efficient to deal with the TSGV problem. Our proposed VSLBase is based on the standard QA framework modified from the QANet [90]. VSLBase first converts the TSGV dataset into a set of SQuAD style triples $(Context, Question, Answer)$; Then it learns to retrieve the target moment by jointly predicting the start and end boundaries in a typical QA style. By converting the TSGV into the format of span-based QA, we also study the differences between standard span-based QA and TSGV tasks. To address the differences, we further propose VSLNet by introducing a query-guided highlighting module on the VSLBase. The query-guided highlighting module bridges the span-based QA and TSGV and guides VSLNet to search for the target moment within a highlighted region. Different from the majority of TSGV solutions, our VSLNet is the first work to solve TSGV from the perspective of span-based question answering. Meanwhile, VSLNet belongs to proposal-free approaches, which are free from the low-efficient and computationally expensive proposal generation process, compared to the majority of proposal-based solutions. The experiments on three benchmark datasets have demonstrated the robustness and effectiveness of the proposed model.

Chapter 4

Multi-Paragraph Question Answering for TSGV

4.1 Introduction

The previous chapter has studied the video localizing network for TSGV, which is built on top of a standard span-based question answering framework. Because TSGV shares significant similarities with the span-based QA task, modeling TSGV from the perspective of span-based QA is a promising direction. Considering the different data nature of video in TSGV and text passage in span-based QA, we further develop a query-guided highlighting module to bridge the differences and boost the performance. However, one challenge in TSGV is that the performance of many existing methods, including VSLNet, degrades significantly along with the increase of video length (detailed in Section 4.3.3). In this chapter[1], we develop an extension of VSLNet which alleviates the issue of performance degradation on long videos by incorporating the concepts of multi-paragraph question answering to retrieve the target moment in an coarse-to-fine structure.

Although there is significant performance degradation of existing TSGV solutions on long videos, these methods generally perform well on short videos. Based on this observation, one straightforward solution to address this issue is to split a long video into multiple short clip segments. Then each clip segment is regarded as a short video. By treating a long video as a document, a clip segment as a paragraph, TSGV can be viewed as the multi-paragraph question answering (MPQA) task [33]. The target moment in a long video can be considered as the answer span in a document for a given language query.

[1]This chapter is accepted as Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. "Natural Language Video Localization: A Revisit in Span-based Question Answering Framework". In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2021.3060449, 2021 [35].

However, how to properly split long video into clip segments is still challenging. Paragraphs in a document are semantically coherent units with boundaries defined by humans. Videos are continuous, and splitting the video into semantically coherent clip segments is difficult, even if it is feasible. In addition, the answer span in MPQA can be found in one of the paragraphs, but we cannot expect the target moment can be found within a single clip segment, regardless of how to split the videos.

Our Approach. In order to solve this issue, we propose a *multi-scale split-and-concat strategy* to partition long video into clips of different lengths. Compared with fixed length splitting, the multi-scale splitting strategy increases the chance of locating a target moment in one segment. In this way, even if a target moment is truncated at one or several scales, segments in other scales may still be able to fully contain it. Thus, we can locate the moment in the clips that are more likely to contain it. Based on the multi-scale split-and-concat strategy, we further develop a Nil Prediction Module (NPM) to tackle the multiple short videos simultaneously, and introduce NPM into the VSLNet to construct an enhanced TSGV framework, termed VSLNet-L[2]. With VSLNet-L, we first coarsely search for the video segment which is more likely containing the target moment, then a fine-grained moment localization process is conducted to precisely retrieve the target moment. We study the performance of VSLNet-L on two benchmark datasets. The results show that VSLNet-L well mitigate the issue of the performance degradation on long videos, and it also significantly boost the TSGV performance on the benchmark datasets.

4.2 VSLNet-L Framework

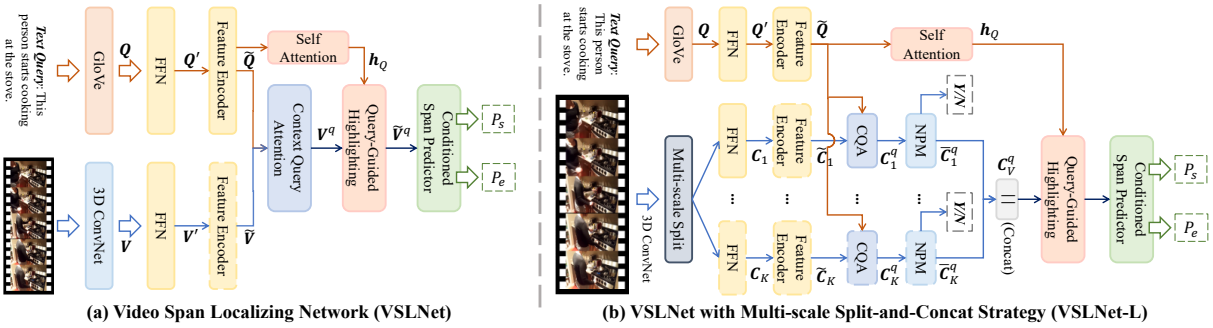
The overall architectures of VSLNet and VSLNet-L are illustrated in Figure 4.1. The detailed introduction of VSLNet are presented in Section 3.2 of Chapter 3. Here, we focus on the elaboration of multi-scale split-and-concat strategy and nil prediction module in VSLNet-L.

As illustrated in Figure 4.2(a), given a long video, VSLNet-L splits it into K clip segments:

$$\boldsymbol{V} = [\boldsymbol{C}_k]_{k=1}^{K}, \text{ and } \boldsymbol{C}_k = [\boldsymbol{v}_i]_{i=(k-1)\times l}^{k\times l}, \tag{4.1}$$

where l is the length of each clip segment $\boldsymbol{C}_k$, *i.e.,* $K \times l = n$. Note that, in our implementation, we perform video split at visual feature level, instead of the untrimmed video itself for computational efficiency. Specifically, we split the features $\boldsymbol{V} = [\boldsymbol{v}_i]_{i=1}^{n}$ obtained from the pre-trained 3D ConvNet (see Section 3.2 of Chapter 3), and use the feature vector $\boldsymbol{V}$ in Equation (4.1) accordingly.

[2]"**L**" represents the multi-scale split-and-concat strategy and the nil prediction module for **L**ong videos.



(a) Video Span Localizing Network (VSLNet)

(b) VSLNet with Multi-scale Split-and-Concat Strategy (VSLNet-L)

FIGURE 4.1: An overview of the proposed architectures for TSGV. The feature extractors, *i.e.,* GloVe and 3D ConvNet, are fixed during training. (a) depicts the structure of VSLNet. (b) shows the architecture of VSLNet-L. The VSLNet-L is built on top of VSLNet by incorporating multi-scale split-and-concat strategy and the nil prediction module (NPM).

Each clip segment $\boldsymbol{C}_k$ is then processed by feature encoder and CQA, to learn query-attended representations $\boldsymbol{C}_k^q = [\boldsymbol{c}_i^q]_{i=(k-1)\times l}^{k\times l} \in \mathbb{R}^{l\times d}$, as shown in Figure 4.1(b). Thus, $\boldsymbol{C}_k^q$ encodes the cross-modal features between clip segment k and query. Then, a Nil Prediction Module (NPM) is introduced in VSLNet-L, to predict whether a clip segment contains or partially contains the temporal moment that corresponds to the text query, as shown in Figure 4.1(b). Next, we detail the structure of the NPM following the illustration in Figure 4.3.

For each clip segment, its query-attended features $\boldsymbol{C}_k^q$ are first encoded by feature encoder as:

$$\widehat{\boldsymbol{C}}_k^q = \texttt{FeatureEncoder}_{\text{NPM}}(\boldsymbol{C}_k^q). \tag{4.2}$$

The self-attention mechanism [13] is adopted to encode $\widehat{\boldsymbol{C}}_k^q$ into clip representation $\boldsymbol{h}_{C_k}^q$, and the nil-score is computed as:

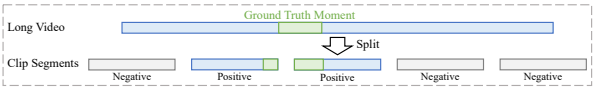
$$\begin{aligned}
\boldsymbol{\alpha}_k &= \texttt{SoftMax}(\texttt{Conv1D}(\widehat{\boldsymbol{C}}_k^q)),\\
\boldsymbol{h}_{C_k}^q &= \sum\nolimits_{i=1}^{l} \alpha_{k,i} \cdot \hat{\boldsymbol{c}}_{(k-1)\times l+i}^q,\\
\mathcal{S}_{\text{nil}}^k &= \sigma(\texttt{FFN}(\boldsymbol{h}_{C_k}^q)),
\end{aligned} \tag{4.3}$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^{l_c}$ and $\boldsymbol{h}_{C_k}^q \in \mathbb{R}^d$. $\mathcal{S}_{\text{nil}}^k \in \mathbb{R}$ is a scalar, which indicates the confidence of clip segment k containing the ground truth moment. The loss of NPM is formulated as:

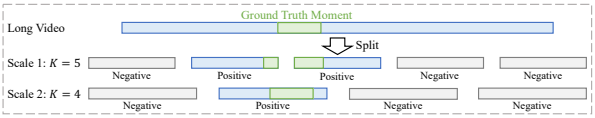
$$\mathcal{L}_{\text{NPM}} = f_{\text{CE}}(\mathcal{S}_{\text{nil}}, Y_{\text{nil}}), \tag{4.4}$$

Y_{nil} is a 0-1 sequence provided during training. A clip segment is positive (label 1) if and only if it overlaps with ground truth moment, illustrated in Figure 4.2. Clip segments that do not contain ground truth moment are negative (label 0). After computing the nil-scores of all

(a) An ideal case of splitting video into clip segments.



(b) A non-ideal case of splitting video into clip segments.



(c) An example of splitting video with two different scales.

FIGURE 4.2: An illustration of splitting video into clip segments.

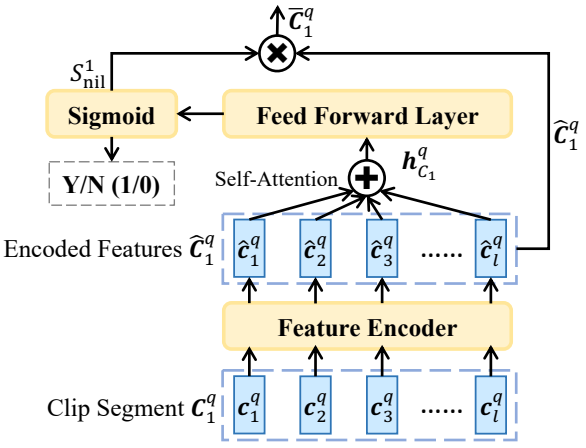


FIGURE 4.3: The structure of Nil Prediction Module (NPM).

clip segments, *i.e.*, $\mathcal{S}_{\text{nil}} = [\mathcal{S}_{\text{nil}}^1, \ldots, \mathcal{S}_{\text{nil}}^K] \in \mathbb{R}^K$, we normalize the scores. The output for clip segment k is:

$$\bar{\mathcal{S}}_{\text{nil}} = \frac{\mathcal{S}_{\text{nil}}}{\max(\mathcal{S}_{\text{nil}})}, \quad \bar{C}_k^q = \bar{\mathcal{S}}_{\text{nil}}^k \times \widehat{C}_k^q, \tag{4.5}$$

where $\bar{\mathcal{S}}_{\text{nil}}$ is the normalized nil-score and $\bar{C}_k^q$ is the re-weighted representations of clip segment k. The NPM highlights the clip segments that contain the target moment, and suppresses other segments. Then the subsequent modules could localize the result by focusing more on the highlighted segments, which is equivalent to narrowing down the searching scope from a long video to a short segment of it.

With the clip segments processed separately in the previous step, we now concatenate the representations of all clip segments, for two reasons. First, a single clip segment may not fully cover the target moment. Second, even if a segment is predicted to be negative (or low confidence), it might provide useful contextual information for localizing the target moment.

$$\bar{C}_V^q = [\bar{C}_1^q \parallel \bar{C}_2^q \parallel \cdots \parallel \bar{C}_K^q], \tag{4.6}$$

where $\parallel$ denotes the concatenation operator and $\bar{C}_V^q \in \mathbb{R}^{n \times d}$. Accordingly, the input feature $\boldsymbol{V}^q$ of QGH is replaced by $\bar{C}_V^q$ in VSLNet-L to compute $\mathcal{L}_{\text{QGH}}$ and $\mathcal{L}_{\text{span}}$. The combined training loss for VSLNet-L is:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} + \mathcal{L}_{\text{NPM}}. \tag{4.7}$$

Unlike a text document, there are no paragraphs as semantic units in a video. Any split in the video may break important contextual information to different clip segments. Although all clip segments will be concatenated again for video level localization, each clip segment is

processed separately, and the contextual information between two segments may not be well captured. To address this issue, we propose a multi-scale split mechanism, to split the video at different segment lengths, *i.e.,* different K (see Figure 4.2(c)). Suppose we use N_s different scales, and for each scale we have:

$$K_i \times l_i = n, \quad \forall\, i = 1, 2, \ldots, N_s, \tag{4.8}$$

where K_i and l_i denote the number of clip segments and clip segment length for the i-th scale, respectively. Through a multi-scale split, contextual information is well captured. Meanwhile, a multi-scale split also provides variation in training samples for the same video and query pair input. The target moment may be located in multiple different clip segments, and its contexts are also different. Note that the same training process as the single-scale split-and-concat applies, except that the clip segments of each scale are fed separately into VSLNet-L, to optimize the objective.

Consequently, we derive N_s predicted moments for a given video and language query pair, because the clip segments at each scale would lead to a pair of start/end boundaries for a predicted moment. During inference, we adopt two simple candidate selection strategies to derive the final target moment. VSLNet-L-P_m strategy selects the candidate with the highest joint boundary probability, $P_m = \max\{P^i_{\text{span}}\}_{i=1}^{N_s}$, where $P^i_{\text{span}} = P^i_s(\hat{a}^s)P^i_e(\hat{a}^e)$ is the maximal joint boundary probability of the moment generated by i-th scale using Equation (3.7). VSLNet-L-U strategy selects two moments with the largest overlap from N_s candidates, and computes their union as the final result.

4.3 Experiments

4.3.1 Experimental Settings

We use the same evaluation metric settings as in Section 3.3.1 of Chapter 3. For benchmark datasets, we utilize ActivityNet Captions, TACoS$_{\text{org}}$, and TACoS$_{\text{2D-TAN}}$. Note the Charades-STA dataset is not used to evaluate VSLNet-L due to its short video length (ref. Table 2.1 in Section 2.3.1 of Chapter 2).

Implementation. For a text query Q, we lowercase all its words and initialize them with 300d GloVe [26] embeddings. The word embeddings are fixed during training. For the untrimmed video V, we extract its visual features using 3D ConvNet pre-trained on Kinetics dataset [25]. We set the maximal feature length n as 300 for ActivityNet Captions, *i.e.,* the extracted visual

feature sequence of a video will be uniformly downsampled if its length $> n$, or zero-padding otherwise. While two maximal feature lengths, $n \in \{300, 600\}$, are used for evaluation on TACoS. When evaluating VSLNet-L, the visual features are split into multiple clip segments using different scales, we use $l = \{60, 75, 100, 150\}$ (*i.e.,* $K = \{5, 4, 3, 2\}$) for $n = 300$, and $l = \{100, 120, 150, 200\}$ (*i.e.,* $K = \{6, 5, 4, 3\}$) for $n = 600$. We set the dimension of all the hidden layers in the model as 128; the kernel size of the convolution layer is 7; the head size of multi-head attention is 8. All the models are trained for 100 epochs with a batch size of 16 and an early stopping strategy for all datasets. Adam [181] is used as the optimizer, with a learning rate of 0.0001, linear decay of learning rate and gradient clipping of 1.0. Dropout [182] of 0.2 is applied to prevent overfitting.[3]

Baselines. For VSLNet-L, we compare it with the following state-of-the-arts:

- Sliding window-based methods: CTRL [27], ACRN [61], ACL-K [28].
- Proposal generated methods: QSPN [29] and SAP [49].
- Standard anchor-based methods: TGN [30], MAN [54], SCDM [31] and CBP [69].
- 2D-map anchor-based methods: 2D-TAN [32].
- Regression-based methods: ABLR [88], ExCL [89], DEBUG [91], GDP [92], LGI [95] and DRN [94].
- Reinforcement learning-based methods: SM-RL [124], RWM-RL [120], TSP-PRL [125].

4.3.2 Overall Performance

Recall that VSLBase is a direct implementation of span-based QA framework on the TSGV task; VSLNet is the extension of VSLBase with QGH; VSLNet-L is a further extension of VSLNet with multi-scale split-concat strategy, designed to handle long videos more effectively. In the following, we show that VSLBase is comparable to existing baselines on TSGV tasks, while VSLNet surpasses VSLBase and all existing baselines, and achieves state-of-the-art results. We then show that VSLNet-L well addresses the issue of performance degradation on TSGV for long videos, compared to VSLNet.

Table 4.1 reports the results on both versions (if available) of TACoS dataset. In general, VSLNet outperforms previous methods over all evaluation metrics. In addition, with the Split-and-Concat mechanism, VSLNet-L further improves the performance, on top of VSLNet. On

[3] All experiments are conducted on dual NVIDIA GeForce RTX 2080Ti GPUs workstation.

TABLE 4.1: "Rank@1, IoU@μ" and "mIoU" results (%) compared with the state-of-the-art on TACoS.

Dataset	Model	Rank@1, IoU = μ $\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
TACoS$_{\text{org}}$	CTRL [27]	18.32	13.30	-	-
	TGN [30]	21.77	18.90	-	-
	ACL [28]	24.17	20.01	-	-
	ACRN [61]	19.52	14.62	-	-
	ABLR [88]	19.50	9.40	-	13.40
	SM-RL [124]	20.25	15.95	-	-
	DEBUG [91]	23.45	11.72	-	16.03
	SCDM [31]	26.11	21.17	-	-
	GDP [92]	24.14	13.90	-	16.18
	CBP [69]	27.31	24.79	19.10	21.59
	DRN [94]	-	23.17	-	-
	VSLNet$^{(S)}$	29.21	24.37	19.37	23.48
	VSLNet$^{(L)}$	29.78	24.71	19.64	23.96
	VSLNet-L-$P_m{}^{(S)}$	**31.94**	26.72	**22.36**	25.71
	VSLNet-L-$U^{(S)}$	31.69	**26.79**	22.02	**25.78**
	VSLNet-L-$P_m{}^{(L)}$	**32.04**	**27.92**	**23.28**	**26.40**
	VSLNet-L-$U^{(L)}$	31.86	27.64	22.72	26.25
TACoS$_{\text{2D-TAN}}$	2D-TAN Pool [32]	37.29	25.32	-	-
	2D-TAN Conv [32]	35.22	25.19	-	-
	VSLNet$^{(S)}$	42.66	32.72	23.12	33.07
	VSLNet$^{(L)}$	41.42	30.67	22.32	31.92
	VSLNet-L-$P_m{}^{(S)}$	**47.66**	**36.15**	**26.19**	**36.24**
	VSLNet-L-$U^{(S)}$	**47.66**	36.12	25.87	35.98
	VSLNet-L-$P_m{}^{(L)}$	**47.11**	**36.34**	**26.42**	**36.61**
	VSLNet-L-$U^{(L)}$	46.44	35.74	26.19	36.05

(S) denotes $n = 300$ and (L) represents $n = 600$.

TABLE 4.2: "Rank@1, IoU@μ" and "mIoU" results (%) compared with state-of-the-arts on ActivityNet Captions.

Model	Rank@1, IoU = μ $\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	mIoU
TGN [30]	45.51	28.47	-	-
ACRN [61]	49.70	31.67	11.25	-
ABLR [88]	55.67	36.79	-	36.99
RWM-RL [120]	-	36.90	-	-
QSPN [29]	45.30	27.70	13.60	-
DEBUG [91]	55.91	39.72	-	39.51
SCDM [31]	54.80	36.75	19.86	-
TSP-PRL [125]	56.08	38.76	-	39.21
GDP [92]	56.17	39.27	-	39.80
CBP [69]	54.30	35.76	17.80	-
DRN [94]	-	**45.45**	24.36	-
LGI [95]	58.52	41.51	23.07	-
2D-TAN Pool [32]	59.45	44.51	26.54	-
2D-TAN Conv [32]	58.75	44.05	27.38	-
VSLBase*	58.18	39.52	23.21	40.56
VSLNet*	**63.16**	43.22	26.16	43.19
VSLNet	61.61	43.78	26.54	43.22
VSLNet-L-P_m	62.18	43.69	27.22	43.67
VSLNet-L-U	62.35	43.86	**27.51**	**44.06**

* denotes that the maximal visual sequence length n of the VSLBase and VSLNet is set as 128, which is consistent with [34].

TACoS$_{\text{org}}$, the results of VSLNet$^{(S)}$ is comparable to that of VSLNet$^{(L)}$, while VSLNet-L$^{(L)}$ surpasses VSLNet-L$^{(S)}$ for both candidate selection strategies. Here, L and S denote the maximal video feature length 600 and 300, respectively. Similar observations hold on TACoS$_{\text{2D-TAN}}$. These results demonstrate that VSLNet-L is more adept than others at localizing temporal moments in longer videos. Moreover, VSLNet-L-P_m is generally superior to VSLNet-L-U under different n for both versions of the TACoS dataset.

The results on the ActivityNet Captions dataset are summarized in Table 4.1. We observe that VSLBase shows similar performance to or slightly better than most of the baselines, while VSLNet further boosts the performance of VSLBase significantly. Comparing VSLNet with $n = 128$ and that with $n = 300$, we find that small n leads to better performance on loose metric (*e.g.,* 63.16 versus 61.61 on IoU $= 0.3$) and large n is beneficial for strict metric (*e.g.,* 26.16 versus 26.54 on IoU $= 0.7$). Meanwhile, the performance of VSLNet-L is comparable to state-of-the-art methods. It is worth noting that 99% annotations in ActivityNet Captions belong to videos that are shorter than 4 minutes. As VSLNet-L is designed to address performance

TABLE 4.3: Statistics of videos and annotations with regard to different video lengths over TSGV datasets.

Dataset	Split	# Videos	# Annots	# of videos / annotations *w.r.t.* different video lengths						
				$0 \sim 2$ min	$2 \sim 4$ min	$4 \sim 6$ min	$6 \sim 8$ min	$8 \sim 10$ min	$10 \sim 12$ min	> 12 min
$\text{TACoS}_{\text{org}}$	Train	75	10,146	27 / 2,847	29 / 4,015	8 / 1,284	4 / 607	3 / 616	2 / 328	2 / 449
	Val	27	4,589	3 / 312	5 / 771	5 / 887	7 / 1,275	1 / 173	4 / 830	2 / 341
	Test	25	4,083	5 / 578	6 / 937	3 / 564	3 / 447	2 / 373	3 / 617	3 / 567
$\text{TACoS}_{\text{2D-TAN}}$	Train	75	9,790	27 / 2,769	29 / 3,840	8 / 1,227	4 / 576	3 / 597	2 / 336	2 / 445
	Val	27	4,436	3 / 311	6 / 929	4 / 639	7 / 1,225	1 / 171	4 / 812	2 / 349
	Test	25	4,001	5 / 594	6 / 907	3 / 535	3 / 428	2 / 370	3 / 598	3 / 569
ANetCap	Train	10,009	37,421	5,278 / 17,806	4,715 / 19,551	8 / 28	3 / 10	4 / 22	0 / 0	1 / 4
	Test	4,917	17,505	2,516 / 8,193	2,392 / 9,274	5 / 19	2 / 9	0 / 0	1 / 5	1 / 5

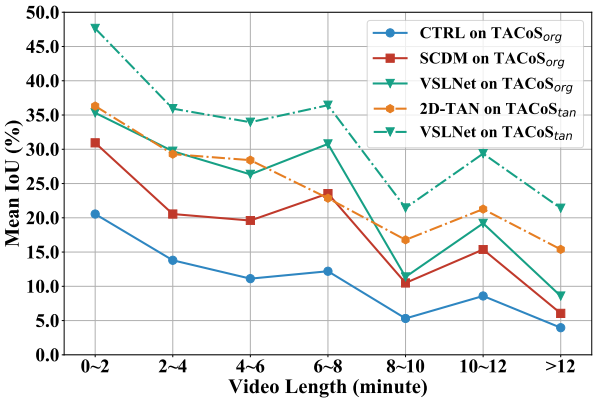


FIGURE 4.4: The Mean IoU (%) performance of CTRL, SCDM, 2D-TAN Pool and VSLNet on the TACoS dataset.

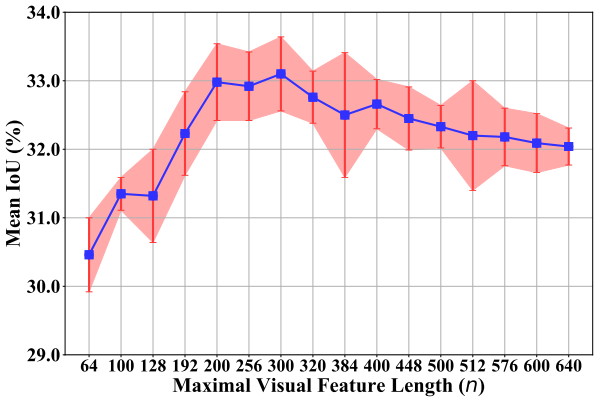


FIGURE 4.5: Mean IoU (%) of VSLNet on $\text{TACoS}_{\text{tan}}$ dataset under different maximal visual representation lengths n.

degradation on long videos, it is reasonable to observe that VSLNet-L achieves less significant performance improvement on ActivityNet Captions compared to TACoS, *w.r.t.* the state-of-the-arts. Moreover, VSLNet-L-U performs better than VSLNet-L-P_m on ActivityNet Captions, different from the observation on TACoS. This could be due to the different ratios of $\bar{L}_{\text{moment}}$ and $\bar{L}_{\text{video}}$ in the two datasets (see Table 2.1 in Section 2.3.1 of Chapter 2). The strategy to select longer spans works better on ActivityNet Captions dataset.

4.3.3 Performance on Videos with Different Length

As discussed in Section 3.1 and illustrated in Figure 4.4, existing methods including VSLNet still underperform on NLVL with long videos. That is, the localization performance decreases dramatically along with the increase of video length. Summarized in Table 4.3, there are fewer videos/annotations along the increase of video length in the datasets. The relatively small number of training samples may lead to instability of the evaluated models, and performance

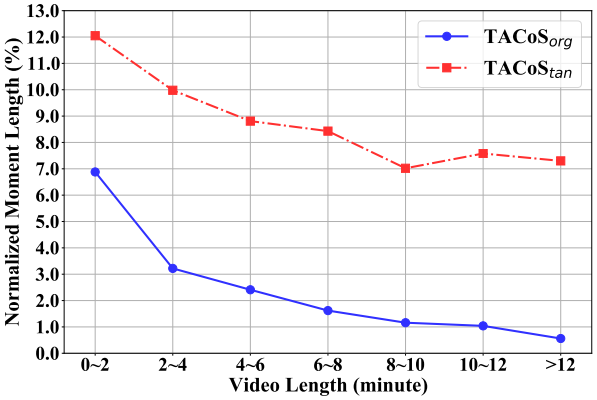


FIGURE 4.6: Statistic of normalized moment lengths in videos for both TACoS$_{\text{org}}$ and TACoS$_{\text{tan}}$.

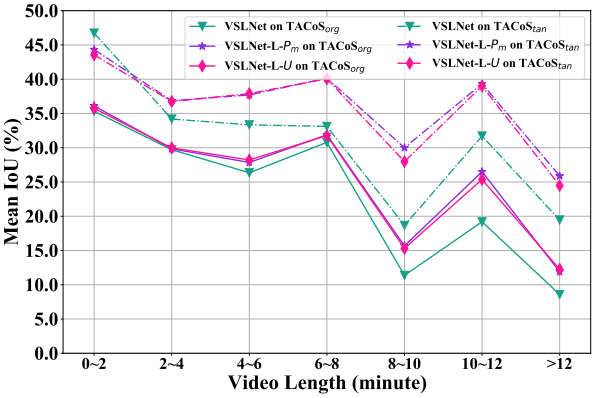


FIGURE 4.7: Performance improvement of VSLNet-L on different video lengths compared to VSLNet on TACoS.

degradation on long videos, to some extent. However, we believe that the following are the two main reasons for the performance degradation:

- Downsampling of visual features of long videos in most existing methods adversely affects localization accuracy due to information loss. As shown in Figure 4.5, sparsely downsampling video feature presentations below certain number (*e.g.,* $n < 200$) would lead to dramatic performance degradation.
- As plotted in Figure 4.6, the average normalized length of ground truth moments gradually decrease along with the increase of video length. The sparsity of moments also contributes to poor performance on long videos.

Table 4.4 reports the "mIoU" gains of VSLNet-L on the TACoS dataset for videos with different lengths. Compared to the results of VSLNet (the best performing method without considering video length), larger improvements are observed on longer videos, which demonstrates the superiority of VSLNet-L for localizing temporal moments in long videos. For instance, VSLNet-L$^{(L)}$ achieves more than 3% absolute improvements in mIoU for videos longer than 8 minutes versus less than 2% gains for videos shorter than 8 minutes on TACoS$_{\text{org}}$. Despite the slight performance reduction on videos shorter than 2 minutes, along with video length raises, consistent improvements are observed on TACoS$_{\text{2D-TAN}}$ for both $n = 300$ (S) and 600 (L), compared to VSLNet. Figure 4.7 plots the performance improvements along video lengths for better visualization. Results on ActivityNet Captions are reported in Table 4.5. Despite that the videos are relatively short, VSLNet-L manages to improve localization performance, with larger improvements observed on longer videos. These results show consistent superiority of VSLNet-L over VSLNet for both candidate selection strategies, on videos of different lengths.

TABLE 4.4: Comparison of mIoU (%) between VSLNet and VSLNet-L on TACoS dataset *w.r.t.* different video lengths.

Dataset	Video Length	$0 \sim 2$ min	$2 \sim 4$ min	$4 \sim 6$ min	$6 \sim 8$ min	$8 \sim 10$ min	$10 \sim 12$ min	> 12 min
$\text{TACoS}_{\text{org}}$	# Test Annotations	578	937	564	447	373	617	567
	VSLNet$^{(S)}$	38.93	27.68	21.56	28.95	12.12	20.59	9.01
	VSLNet-L-$P_m{}^{(S)}$	**39.91** +0.98	28.11 +0.43	24.63 +3.07	**30.45** +1.50	15.26 +3.14	24.56 +3.97	**12.75** +3.74
	VSLNet-L-$U^{(S)}$	39.54 +0.61	**28.69** +1.01	**24.85** +3.29	30.37 +1.42	**15.27** +3.15	**24.87** +4.28	12.15 +3.14
	VSLNet$^{(L)}$	35.32	29.72	26.34	30.78	11.37	19.19	8.58
	VSLNet-L-$P_m{}^{(L)}$	**36.11** +0.79	29.84 +0.12	27.85 +1.51	**31.88** +1.10	**15.68** +4.31	**26.50** +7.31	11.90 +3.32
	VSLNet-L-$U^{(L)}$	35.75 +0.43	**29.98** +0.26	**28.21** +1.87	31.78 +1.00	15.31 +3.94	25.37 +6.18	**12.26** +3.68
$\text{TACoS}_{\text{2D-TAN}}$	# Test Annotations	594	907	535	428	370	598	569
	VSLNet$^{(S)}$	**47.64**	35.93	33.95	36.43	21.44	29.37	21.40
	VSLNet-L-$P_m{}^{(S)}$	46.12 -1.52	**37.76** +1.83	36.51 +2.56	41.37 +4.94	**29.71** +8.27	**34.84** +5.47	25.08 +3.68
	VSLNet-L-$U^{(S)}$	45.91 -1.73	37.52 +1.59	**36.85** +2.90	**41.85** +5.42	28.30 +6.86	33.84 +4.47	**25.20** +3.80
	VSLNet$^{(L)}$	**46.73**	34.19	33.34	33.11	18.66	31.71	19.45
	VSLNet-L-$P_m{}^{(L)}$	44.32 -2.41	**36.86** +2.67	37.70 +4.36	**40.15** +7.04	**29.99** +11.33	**39.33** +7.62	**25.88** +6.43
	VSLNet-L-$U^{(L)}$	43.56 -3.17	36.74 +2.55	**37.90** +4.56	40.06 +6.95	27.97 +9.31	39.01 +7.30	24.48 +5.03

(S) denotes $n = 300$ and (L) represents $n = 600$. Performance gain and loss are indicated in different colors.

TABLE 4.5: Comparison of mIoU (%) between VSLNet and VSLNet-L on ActivityNet Captions *w.r.t.* different video lengths.

Video Length	$0 \sim 2$ min	$2 \sim 4$ min	> 4 min
# Test Samples	8,193	9,274	38
VSLNet	46.21	40.60	35.24
VSLNet-L-P_m	46.59 +0.38	41.11 +0.51	38.40 +3.16
VSLNet-L-U	**47.03** +0.82	**41.46** +0.86	**39.63** +4.39

Performance gain and loss are indicated in different colors.

4.3.4 Ablation Study

In this section, we conduct ablative experiments to analyze the effects of proposed multi-scale split-and-concat strategy in VSLNet-L, as well as visually show the effectiveness of the proposed methods and discuss their limitations.

Compared to VSLNet, VSLNet-L further introduces a multi-scale split-and-concat strategy to address performance degradation on long videos. Here, we study the impact of the multi-scale split on the TACoS dataset with $n = 600$, against the single-scale split. We evaluate 4 different values of single-scale, *i.e.,* $l \in \{100, 120, 150, 200\}$. The multi-scale mechanism is jointly trained with the four scales. The results are summarized in Table 4.6. Compared to VSLNet, split-and-concat strategy in VSLNet-L consistently contributes to performance improvements, regardless of the l value. The best single-scale l is 120 for $\text{TACoS}_{\text{org}}$, and 150 for $\text{TACoS}_{\text{tan}}$. Compared to VSLNet-L with single-scale, VSLNet-L-P_m (-U) further improves all

TABLE 4.6: Results (%) of VSLNet-L on TACoS using different split scales with $n = 600$.

Dataset	Model	Scales (l)	Rank@1, IoU = μ			mIoU
			$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
$\text{TACoS}_{\text{org}}$	VSLNet	-	29.78	24.71	19.64	23.96
	VSLNet-L	100	30.18	25.81	20.77	24.46
		120	31.13	26.87	21.19	25.12
		150	30.42	26.38	20.89	24.73
		200	30.59	26.07	21.01	24.61
		-P_m	32.04	27.92	23.28	26.40
		-U	31.86	27.64	22.72	26.25
$\text{TACoS}_{\text{tan}}$	VSLNet	-	41.42	30.67	22.32	31.92
	VSLNet-L	100	42.24	32.69	23.67	32.41
		120	43.39	33.37	24.19	33.45
		150	44.61	33.99	24.27	33.71
		200	43.74	33.67	23.74	33.50
		-P_m	47.11	36.34	26.42	36.61
		-U	46.44	35.74	26.19	36.05

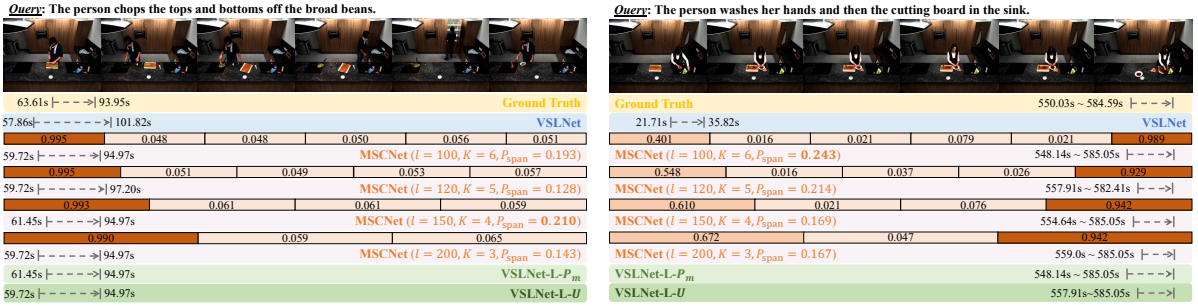


FIGURE 4.8: Visualizations of two predicted examples by VSLNet and VSLNet-L on TACoS dataset.

metrics significantly. The multi-scale split-and-concat mechanism not only alleviates the issue of target moment truncation but also captures contextual information in the video; both improve the generalization ability of VSLNet-L.

Figure 4.8 depicts two predicted examples from the TACoS dataset as case studies. The localized moments by VSLNet-L are more accurate and closer to the ground truth moment than that of VSLNet. Both figures show the results of VSLNet-L are constrained in the positive clip segments, *i.e.,* the clip segment that contains ground truth. In the second example, VSLNet does not capture the concept "the cutting board in the sink" and focuses on retrieving "washes" action only, which leads to an error prediction. In contrast, VSLNet-L correctly understands both "washes" and the position of "cutting board", leading to the correct prediction.

4.4 Summary

In this chapter, we have studied a common weakness of the existing TSGV solutions, that is, the performance of existing methods degrades significantly along with the increase of video length. Then, we presented an enhanced VSLNet architecture termed VSLNet-L to solve the issue of performance degradation on long videos for TSGV. Specifically, the VSLNet-L borrow the concepts of multi-paragraph question answering to decompose the long videos into multiple short videos with multi-scale split-and-concat strategy, where each short video can be regarded as a text passage in the document. In this sense, retrieving target moment from long videos is equivalent to 1) predicting the video segment that more likely contains the target moment and 2) precisely localizing the target moment within the selected video segment. To achieve the video segment selection, we developed a nil prediction module to measure the probabilities of each video segment contains the target moment in parallel. The experimental results demonstrate that our proposed method, VSLNet-L, can well mitigate the performance degradation issue on long videos. Meanwhile, with the distinct improvements on long videos, VSLNet-L further outperforms the evaluated state-of-the-art baselines by a large margin.

Besides, both VSLNet (Chapter 3) and VSLNet-L (Chapter 4) leverage the concepts of question answering task in NLP, where VSLNet is inherited from the standard span-based QA and VSLNet-L simulates the multi-paragraph QA. Although question answering is a typical NLP task, TSGV shares significant similarities with QA at the feature level as presented in Section 3.1 of Chapter 3. Thus, the extension from VSLNet to VSLNet-L further demonstrate the effectiveness of solving TSGV from the perspective of question answering.

Chapter 5

Parallel Attention Network with Sequence Matching

5.1 Introduction

The previous chapters have studied the idea of solving TSGV from the perspective of question answering. Specifically, we first develop a standard span-based QA approach named VSLBase on TSGV task; VSLNet-L is the extension of VSLBase with query-guided highlighting module to bridge the differences between TSGV and span-based QA; VSLNet-L is a further extension of VSLNet with multi-scale split-and-concat strategy, designed to handle long videos more effectively. All of those approaches belongs to the one-stage proposal-free span-based solutions, which directly predict start/end boundaries of target moments, through modeling video-text interactions.

As discussed in Section 2.1 of Chapter 2, the feature interactor plays a key role in TSGV, and the quality of feature interactor determines the performance of TSGV model to a large extent. However, the previous proposed approaches, VSLBase, VSLNet and VSLNet-L, do not explore more on the multimodal interaction between video and text. Besides, despite the high computation-efficiency and localization accuracy, many existing proposal-free methods, which directly predicts the start and end boundaries of target moments, suffer from sparsity issue, *i.e.,* the positive and negative training samples are extremely imbalanced. Such issue hinders the accuracy of moment boundary prediction significantly. To address the aforementioned issues, in this chapter[1], we focus on the video-text interaction modeling and moment boundary prediction to improve the TSGV performance.

[1]This chapter is published as Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Siow Mong Rick Goh. "Parallel Attention Network with Sequence Matching for Video Grounding". In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (Long Papers)*, pages: 776–790, Online, 2021 [36].

Label:	**B-ORG**	**I-ORG**	**I-ORG**	**E-ORG**	**O**	**O**	**O**	**O** ...
Sent:	KinderCare	Learning	Centers	Inc.	said	that	a	debt ...

FIGURE 5.1: An example of the annotations in NER, where "ORG" is for "Organization", "B", "I" and "E" denote the begin, inside and end of the organization entity, respectively.

Video-Text Interaction Modeling. In order to model video-text interaction, various attention-based methods have been proposed [27, 31, 95]. In particular, transformer block [177] is widely used in vision-language tasks and proved to be effective for multimodal learning [19, 21, 22]. In TSGV task, fine-grain scale unimodal representations are also important to achieve good localization performance. However, existing solutions do not refine unimodal representations of video and text when doing cross-modal reasoning, and thus limit the performance. To better capture informative features for multi-modalities, we encode both self-attentive contexts and cross-modal interactions from video and query. That is, instead of solely relying on sophisticated cross-modal learning as in most existing studies, we learn both intra- and inter-modal representations simultaneously, with improved attention modules.

Moment boundary prediction. In terms of the length, target moment is usually a very small portion of the video, making positive (frames in target moment) and negative (frames not in target moment) samples imbalanced. Further, we aim to predict the exact start/end boundaries (*i.e.,* two video frames[2]) of the target moment. If we view from the space of video frames, sparsity is a major concern, *e.g.,* catching two frames among thousands. Recent studies attempt to address this issue by auxiliary objectives, *e.g.,* to discriminate whether each frame is foreground (positive) or background (negative) [88, 95], or to regress distances of each frame within target moment to ground truth boundaries [91, 94]. However, the "sequence" nature of frames or videos is not considered. Thus, we emphasize the "sequence" nature of video frames and adopt the concept of sequence labeling in NLP to video grounding. We use named entity recognition (NER) [116, 183] as an example sequence labeling task for illustration in Figure 5.1. Video grounding is to retrieve a sequence of frames with start/end boundaries of target moment from video. This is analogous to extract a multi-word named entity from a sentence. The main difference is that, words are discrete, so word annotations (*i.e.,* B, I, E, and O tags) in sentence are discrete. In contrast, video is continuous and the changes between consecutive frames are smooth. Hence, it is difficult (and also not necessary) to precisely annotate each frame. We relax the annotations on video sequence by specifying video regions, instead of frames. With

[2]The "frame" is a general description, which can refer to a frame in a video sequence or a unit in the corresponding video feature representation.

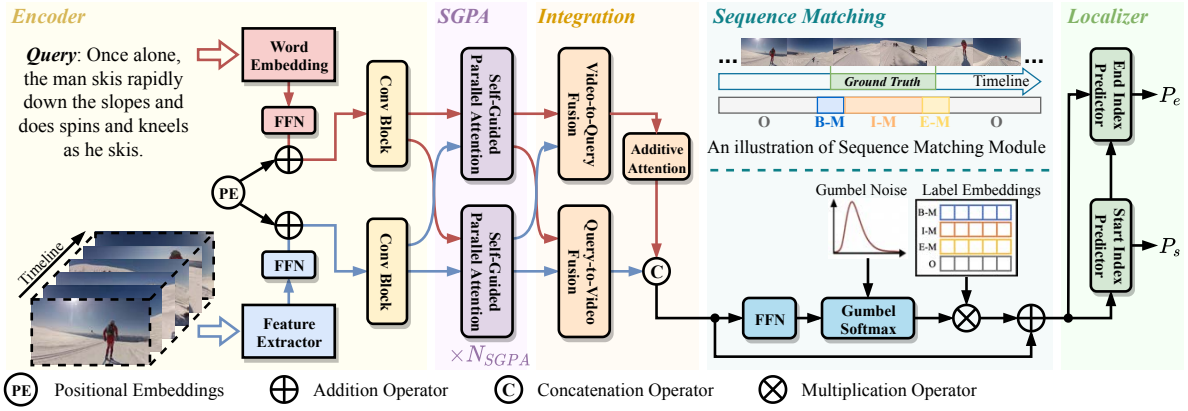


FIGURE 5.2: The architecture of the Parallel Attention Network with Sequence Matching (SeqPAN) for TSGV.

respect to the target moment, we label B, I, E and O (BIEO) regions on video (see Figure 5.2) and introduce label embeddings to model these regions.

Our Approach. Based on the above analysis, we propose a Parallel Attention Network with Sequence matching (SeqPAN) for TSGV task. Note that SeqPAN also follows a similar structure to the span-based question answering framework, *i.e.,* VSLBase and VSLNet. In SeqPAN, we first design a *self-guided parallel attention (SGPA)* module to capture both self- and cross-modal attentive information for each modality simultaneously. In SGPA module, a cross-gating strategy with self-guided head is further used to fuse self- and cross-modal representations. We then propose a *sequence matching (sq-match)* strategy, to identify BIEO regions in video. The label embeddings are incorporated to represent label of frames in each region for region recognition. The sq-match guides SeqPAN to search for boundaries of target moment within constrained regions, leading to more precise localization results. Via extensive experiments on three benchmark datasets, we show that both SGPA and sq-match consistently improve the performance; and SeqPAN surpasses the state-of-the-art methods.

5.2 SeqPAN Framework

The overall architecture of the proposed SeqPAN model is shown in Figure 5.2. Since SeqPAN also follows a similar task formulation as VSLNet, the details of task formulation are same in Section 3.2 of Chapter 3.

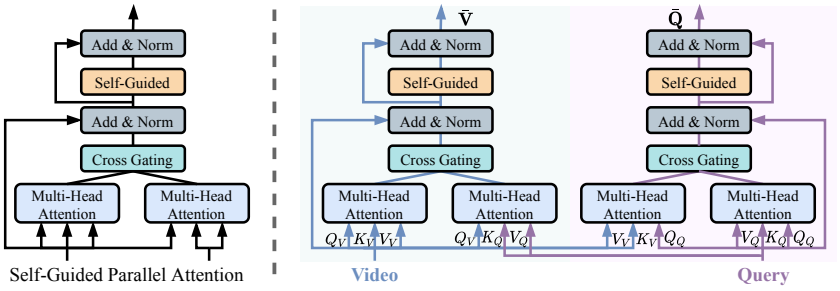


FIGURE 5.3: Self-Guided Parallel Attention (SGPA). Left: the structure of SGPA; Right: the parallel streams of encoding visual and textual inputs.

5.2.1 Encoder Module

Given visual features $\boldsymbol{V} \in \mathbb{R}^{n \times d_v}$ of the video and word embeddings $\boldsymbol{Q} \in \mathbb{R}^{m \times d_q}$ of the language query, we map them into the same dimension d with two FFNs, respectively. The encoder module mainly encodes the individual modality separately. As position encoding offers a flexible way to embed a sequence, when the sequence order matters, we first incorporate a position embedding to every input of both video and query sequences. Then, we adopt stacked 1D convolutional block to learn representations by carrying knowledge from neighbor tokens. The encoded representations are written as:

$$\begin{aligned} \boldsymbol{V'} &= \texttt{ConvBlock}(\texttt{FFN}_\texttt{v}(\boldsymbol{V}) + \boldsymbol{E}_p), \\ \boldsymbol{Q'} &= \texttt{ConvBlock}(\texttt{FFN}_\texttt{q}(\boldsymbol{Q}) + \boldsymbol{E}_p), \end{aligned} \tag{5.1}$$

where $\boldsymbol{V'} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{Q'} \in \mathbb{R}^{m \times d}$; $\boldsymbol{E}_p$ denotes the positional embeddings. FFN denotes the single-layer feed-forward network, where $\texttt{FFN}(\boldsymbol{X}) = \boldsymbol{W} \cdot \boldsymbol{X} + \boldsymbol{b}$. Both position embeddings and convolutional block are shared by the video and text features.

5.2.2 Self-Guided Parallel Attention Module

A self-guided parallel attention (SGPA) module (see Figure 5.3) is proposed to improve multimodal representation learning. Compared with standard transformer (TRM) encoder, SGPA uses two parallel multi-head attention blocks to learn both *uni-modal* and *cross-modal* representations simultaneously, and merge them with a cross-gating strategy.[3] Taking video modality as an example, the attention process is computed as:

$$\hat{\boldsymbol{V}}_\text{S} = \sigma_s\Big(\frac{Q_V K_V^\top}{\sqrt{d}}\Big) \cdot V_V, \text{ and } \hat{\boldsymbol{V}}_\text{C} = \sigma_s\Big(\frac{Q_V K_Q^\top}{\sqrt{d}}\Big) \cdot V_Q, \tag{5.2}$$

[3] A detailed comparison of SGPA and standard TRMs is summarized in the experiments.

where σ_s denotes Softmax; Q_V, K_V and V_V are the linear projections of $\boldsymbol{V'}$; Q_Q, K_Q and V_Q are linear projections of $\boldsymbol{Q'}$; $\hat{\boldsymbol{V}}_{\mathrm{S}}$ encodes the self-attentive contexts within video modality; and $\hat{\boldsymbol{V}}_{\mathrm{C}}$ integrates information from query modality according to cross-modal attentive relations. The self- and cross-modal representations are then merged together by a cross-gating strategy:

$$\hat{\boldsymbol{V}} = \sigma\big(\texttt{FFN}(\hat{\boldsymbol{V}}_{\mathrm{C}})\big) \odot \hat{\boldsymbol{V}}_{\mathrm{S}} + \sigma\big(\texttt{FFN}(\hat{\boldsymbol{V}}_{\mathrm{S}})\big) \odot \hat{\boldsymbol{V}}_{\mathrm{C}}, \tag{5.3}$$

where σ denotes Sigmoid function and $\odot$ represents Hadamard product. The cross-gating explicitly interacts features obtained from the self- and cross-attention encoders to ensure both are fully utilized, instead of relying on only one of them. Finally, we employ a self-guided head to implicitly emphasize the informative representations by measuring the confidence of each element in $\hat{\boldsymbol{V}}$ as:

$$\bar{\boldsymbol{V}} = \sigma\big(\texttt{FFN}_{\sigma}(\hat{\boldsymbol{V}})\big) \odot \texttt{FFN}(\hat{\boldsymbol{V}}). \tag{5.4}$$

The refined representations $\bar{\boldsymbol{Q}}$ for the query modality are obtained in a similar manner (*e.g.,* swapping visual and query features).

5.2.3 Video-Query Integration Module

This module further enhances the cross-modal interactions between visual and textual features. It utilizes context-query attention (CQA) strategy [90] and aggregates text information for each visual element. To better elaborate the computation process of this module, suppose we have two inputs $\boldsymbol{X} \in \mathbb{R}^{N_x \times d}$ and $\boldsymbol{Y} \in \mathbb{R}^{N_y \times d}$. Given the two inputs, the context-query attention first computes similarities between each pair of $\boldsymbol{X}$ and $\boldsymbol{Y}$ elements as:

$$\mathcal{S} = \boldsymbol{X} \cdot \boldsymbol{W} \cdot \boldsymbol{Y}^{\top}, \tag{5.5}$$

where $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ and $\mathcal{S} \in \mathbb{R}^{N_x \times N_y}$. Then X-to-Y and Y-to-X attention weights are computed by:

$$\mathcal{A}_{\mathrm{XY}} = \mathcal{S}_r \cdot \boldsymbol{Y} \in \mathbb{R}^{N_x \times d}, \text{ and } \mathcal{A}_{\mathrm{YX}} = \mathcal{S}_r \cdot \mathcal{S}_c^{\top} \cdot \boldsymbol{X} \in \mathbb{R}^{N_x \times d}, \tag{5.6}$$

where $\mathcal{S}_r$ and $\mathcal{S}_c$ are the row- and column-wise normalization of $\mathcal{S}$ by Softmax function. The final output of context-query attention is calculated as:

$$\boldsymbol{X}^{Y} = \texttt{FFN}\big([\boldsymbol{X}; \mathcal{A}_{\mathrm{XY}}; \boldsymbol{X} \odot \mathcal{A}_{\mathrm{XY}}; \boldsymbol{X} \odot \mathcal{A}_{\mathrm{YX}}]\big), \tag{5.7}$$

where $\odot$ denotes element-wise multiplication, ";" denotes concatenation operation, and $\boldsymbol{X}^{Y} \in \mathbb{R}^{N_x \times d}$. In this way, the information of $\boldsymbol{Y}$ is properly fused into $\boldsymbol{X}$.

By setting $\boldsymbol{X} = \bar{\boldsymbol{V}} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{Y} = \bar{\boldsymbol{Q}} \in \mathbb{R}^{m\times d}$, we can derive the query-aware video representations $\boldsymbol{V}^Q \in \mathbb{R}^{n\times d}$. Similarly, the video-aware query representations $\boldsymbol{Q}^V \in \mathbb{R}^{m\times d}$ is obtained by setting $\boldsymbol{X} = \bar{\boldsymbol{Q}}$ and $\boldsymbol{Y} = \bar{\boldsymbol{V}}$.

Next, we encode $\boldsymbol{Q}^V = [\boldsymbol{q}_0^V, \ldots, \boldsymbol{q}_{M-1}^V]$ into sentence representation $\boldsymbol{q}$ with additive attention [13]:

$$\begin{aligned} \boldsymbol{\alpha} &= \texttt{Softmax}\big(\boldsymbol{Q}^V \cdot \boldsymbol{W}_\alpha\big) \in \mathbb{R}^M, \\ \boldsymbol{q} &= \sum_{i=0}^{M-1} \alpha_i \times \boldsymbol{q}_i^V \in \mathbb{R}^d, \end{aligned} \tag{5.8}$$

where $\boldsymbol{W}_\alpha \in \mathbb{R}^{d\times 1}$. $\boldsymbol{q}$ is then concatenated with each element of $\boldsymbol{V}^Q$ as $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] \in \mathbb{R}^{n\times 2d}$, where $\boldsymbol{h}_i = [\boldsymbol{v}_i^Q; \boldsymbol{q}]$. Finally, the query-attended visual representation is computed as:

$$\bar{\boldsymbol{H}} = \boldsymbol{H} \cdot \boldsymbol{W}_h + \boldsymbol{b}_h, \tag{5.9}$$

where $W_h \in \mathbb{R}^{2d\times d}$ and $b_h \in \mathbb{R}^d$ denote the learnable weight and bias, and $\bar{\boldsymbol{H}} \in \mathbb{R}^{n\times d}$.

5.2.4 Sequence Matching Module

As illustrated in Figure 5.2, we considers the frames within ground truth moment and several neighboring frames as foreground, while the rest as background. Then, we split the foreground into **B**egin, **I**nside, and **E**nd regions. For simplicity, we assign each region a label, *i.e.,* "B-M" for begin, "I-M" for inside, "E-M" for end region, and "O" for background. B-M/E-M explicitly indicate potential positions of the start/end boundaries. We also specify orthogonal label embeddings $\boldsymbol{E}_{\text{lab}} \in \mathbb{R}^{4\times d}$ to represent those labels, and to infuse label information into visual features after region label predictions.

Note our approach is different from previous work [184] on temporal action proposal generation task, where the target proposal is split into start, centre, and end regions. The probability of a frame belonging to each of three regions is predicted separately in a regression manner, leading to three separate probability sequences, one for each region. The maximum probabilities in the sequences are used to guide proposal generations. In contrast, we formulate matching process as a multi-class classification problem and predict a concrete region label for each frame, *i.e.,* same as a sequence labeling task in NLP. Label embeddings are then assigned to the frames based on the labels of the predicted region.

A straightforward solution to predict the confidence of an element belonging to each region is multi-class classifier:

$$\boldsymbol{H}_{\text{seq}} = \texttt{FFN}_{\text{seq}}(\bar{\boldsymbol{H}}),\ \boldsymbol{S}_{\text{seq}} = \sigma_s(\boldsymbol{H}_{\text{seq}}) \in \mathbb{R}^{n\times 4}, \tag{5.10}$$

where $\boldsymbol{S}_{\text{seq}}$ encodes the probabilities of each visual element in different regions. Then label index with highest probability from $\boldsymbol{S}_{\text{seq}}$ is selected to represent the predicted label for each visual element:

$$\boldsymbol{L}_{\text{lab}} = [\arg\max(\boldsymbol{S}^{j}_{\text{seq}})]^{n-1}_{j=0} \in \mathbb{R}^{n}. \tag{5.11}$$

However, a major issue here is that Equation (5.11) needs to sample from a discrete probability distribution, which makes the back-propagation of gradients through $\boldsymbol{S}_{\text{seq}}$ in Equation (5.10) infeasible for optimizer.

To make back-propagation possible, we introduce the categorical reparameterization strategy. Categorical reparameterization, *e.g.,* the reinforce-based approaches [185, 186], straight-through estimators [187] and the Gumbel-Softmax [188, 189], is a strategy that enables discrete categorical variables to back-propagate in neural networks. It aims to estimate smooth gradient with a continuous relaxation for categorical variable. In SeqPAN, we use Gumbel-Softmax to approximate the sequence labels from a probability distribution. Then those labels are applied to lookup the corresponding embeddings for region representation in sequence matching module. Let $\boldsymbol{x} = (x_1, \ldots, x_l)$ be a categorical distribution, where l is the number of categories, x_c is probability score of category c and $\sum_{c=1}^{l} x_c = 1$. Given the independent and identically distributed Gumbel noise $\boldsymbol{g} = (g_1, \ldots, g_l)$ from $\texttt{Gumbel}(0, 1)$ distribution[4], the soft categorical sample can be computed as:

$$\boldsymbol{y} = \texttt{Softmax}\big((\log(\boldsymbol{x}) + \boldsymbol{g})/\tau\big), \tag{5.12}$$

where $\tau > 0$ is annealing temperature, and Equation (5.12) is referred as Gumbel-Softmax operation on $\boldsymbol{x}$. As $\tau \to 0^{+}$, $\boldsymbol{y}$ is equivalent to the Gumbel-Max form [190, 191] as:

$$\hat{\boldsymbol{y}} = \texttt{Onehot}\big(\arg\max(\log(\boldsymbol{x}) + \boldsymbol{g})\big), \tag{5.13}$$

where $\hat{\boldsymbol{y}}$ is an unbiased sample from $\boldsymbol{x}$ and thus we can draw differentiable samples from the distribution during training. Note, when input $\boldsymbol{x}$ is unnormalized, the $\log(\cdot)$ operator in Equation (5.12) and Equation (5.13) shall be omitted [188]. During inference, discrete samples can be drawn with the Gumbel-Max trick directly.

Thus, with the Gumbel-Max [190, 191] trick, we can re-formulate Equation (5.11) as:

$$\hat{\boldsymbol{L}}_{\text{lab}} = \big[\texttt{Onehot}(\arg\max(\boldsymbol{H}^{j}_{\text{seq}} + \boldsymbol{g}))\big]^{n-1}_{j=0}, \tag{5.14}$$

[4]The $\texttt{Gumbel}(0, 1)$ distribution can be sampled using inverse transform sampling by drawing $u \sim \texttt{Uniform}(0, 1)$ and computing $g = -\log(-\log(u))$ [188].

where $\hat{\boldsymbol{L}}_{\text{lab}} \in \mathbb{R}^{n\times 4}$. Then, we utilize the Gumbel-Softmax [188, 189] to relax the $\arg\max$ so as to make Equation (5.14) being differentiable. Formally, we use Equation (5.15) to approximate Equation (5.14) as:

$$\bar{\boldsymbol{L}}_{\text{lab}} = \sigma_s\big((\boldsymbol{H}_{\text{seq}} + \boldsymbol{g})/\tau\big) \in \mathbb{R}^{n\times 4}. \tag{5.15}$$

As $\tau \to 0^+$, $\bar{\boldsymbol{L}}_{\text{lab}} \approx \hat{\boldsymbol{L}}_{\text{lab}}$, while $\tau \to \infty$, each element in $\bar{\boldsymbol{L}}_{\text{lab}}$ will be the same and the approximated distribution will be smooth. Note we use Equation (5.14) during forward pass while Equation (5.15) for backward pass to allow gradient back-propagation. As the result, the embedding lookup process is differentiable and the label-attended visual representations is derived as:

$$\widetilde{\boldsymbol{H}} = \boldsymbol{E}_{\text{lab}} \cdot \hat{\boldsymbol{L}}_{\text{lab}} + \bar{\boldsymbol{H}}. \tag{5.16}$$

In general, the training objective is defined as:

$$\mathcal{L}_{\text{seq}} = f_{\text{XE}}(\bar{\boldsymbol{L}}_{\text{lab}}, \boldsymbol{Y}_{\text{lab}}) + \|\boldsymbol{E}_{\text{lab}}^{\top}\boldsymbol{E}_{\text{lab}} \odot (\boldsymbol{1} - \boldsymbol{I})\|_{\text{F}}^2, \tag{5.17}$$

where $\boldsymbol{Y}_{\text{lab}}$ denotes the ground truth sequence labels, $\boldsymbol{1}$ is the matrix with all elements being 1 and $\boldsymbol{I}$ is the identity matrix. The second term in Equation (5.17) is the orthogonal regularization [192], which ensures $\boldsymbol{E}_{\text{lab}}$ to keep the orthogonality.

5.2.5 Localization Module

Finally, we present a conditioned localizer to predict the start and end boundaries of the target moment. The localizer consists of two stacked transformer blocks and two FFNs. The scores of start and end boundaries are calculated as:

$$\begin{aligned} \boldsymbol{H}_{\text{s}} &= \texttt{TRM}_s(\widetilde{\boldsymbol{H}}),\ \boldsymbol{S}_s = \boldsymbol{W}_s[\boldsymbol{H}_{\text{s}}; \widetilde{\boldsymbol{H}}] + \boldsymbol{b}_s, \\ \boldsymbol{H}_{\text{e}} &= \texttt{TRM}_e(\boldsymbol{H}_{\text{s}}),\ \boldsymbol{S}_e = \boldsymbol{W}_e[\boldsymbol{H}_{\text{e}}; \widetilde{\boldsymbol{H}}] + \boldsymbol{b}_e, \end{aligned} \tag{5.18}$$

where $\boldsymbol{S}_{s/e} \in \mathbb{R}^N$. $\boldsymbol{W}_{s/e}$ and $\boldsymbol{b}_{s/e}$ are the weight and bias of start/end FFNs, respectively. Note the representations of end boundary ($\boldsymbol{H}_{\text{e}}$) are conditioned on that of start boundary ($\boldsymbol{H}_{\text{s}}$) to ensure the predicted end boundary is always after start boundary. Then, the probability distributions of start/end boundaries are computed by $\boldsymbol{P}_{s/e} = \texttt{Softmax}(\boldsymbol{S}_{s/e}) \in \mathbb{R}^N$. The training objective is:

$$\mathcal{L}_{\text{loc}} = \frac{1}{2} \times \big[f_{\text{XE}}(\boldsymbol{P}_s, \boldsymbol{Y}_s) + f_{\text{XE}}(\boldsymbol{P}_e, \boldsymbol{Y}_e)\big], \tag{5.19}$$

where f_{XE} is cross-entropy function, $\boldsymbol{Y}_{s/e}$ is one-hot labels for start/end boundaries.

5.3 Experiments

5.3.1 Experimental Settings

We use the same evaluation metric settings as in Section 3.3.1 of Chapter 3. For benchmark datasets, we utilize Charades-STA, ActivityNet Captions, $\text{TACoS}_{\text{org}}$, and $\text{TACoS}_{\text{2D-TAN}}$.

Implementation. We follow [34, 89, 95, 104] and use 3D ConvNet pre-trained on Kinetics dataset (*i.e.,* I3D) [25] to extract visual features from videos. The maximal visual feature sequence lengths are set to 64, 100, and 256 for Charades-STA, ActivityNet Captions, and TACoS, respectively. This setting is based on the average video lengths in the three datasets. The feature sequence length of a video will be uniformly downsampled if it is larger than the pre-set threshold, or zero-padding otherwise. For the language queries, we lowercase all the words and initialize them with GloVe [26] embeddings. The word embeddings and extracted visual features are fixed during training.

For other hyper-parameters, we use the same settings for all datasets. The dimension of the hidden layers is 128; the head number in multi-head attention is 8; the number of SGPA blocks (N_{SGPA}) is 2; the annealing temperature τ of Gumbel-Softmax is 0.3; The Dropout [182] is 0.2. The maximal training epochs $E = 100$ is used, with batch size of 16 and early stopping tolerance of 10 epochs. We adopt Adam [181] optimizer, with initial learning rate of $\beta_0 = 0.0001$, weight decay 0.01, and gradient clipping 1.0, to train the model. The learning rate decay strategy is defined as $\beta_e = \beta_0 \times (1 - \frac{e}{E})$, where e denotes the e-th training epoch.[5]

Baselines. We compare SeqPAN with the following state-of-the-arts methods:

- Sliding window-based methods: ACL [28].
- Standard anchor-based methods: TGN [30], MAN [54], SCDM [31], and CBP [69].
- 2D-map anchor-based methods: 2D-TAN [32].
- Regression-based methods: DEBUG [91], ExCL [89], GDP [92], LGI [95], DRN [94].
- Span-based methods: VSLNet [34] and TMLGA [104].
- Reinforcement learning-based methods: TSP-PRL [125].

[5]The SeqPAN is implemented using TensorFlow `1.15.0` with CUDA `10.0` and cudnn `7.6.5`. All the experiments are conducted on a workstation with dual NVIDIA GeForce RTX 2080Ti GPUs.

TABLE 5.1: Comparison with the SOTA methods on Charades-STA dataset.

Methods	R@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
DEBUG	54.95	37.39	17.69	36.34
ExCL	61.50	44.10	22.40	-
MAN	-	46.53	22.72	-
SCDM	-	54.44	33.43	-
CBP	-	36.80	18.87	-
GDP	54.54	39.47	18.49	-
2D-TAN	-	39.81	23.31	-
TSP-PRL	-	45.30	24.73	40.93
TMLGA	67.53	52.02	33.74	-
VSLNet	70.46	54.19	35.22	*50.02*
DRN	-	53.09	31.75	-
LGI	*72.96*	*59.46*	*35.48*	-
SeqPAN	**73.84**	**60.86**	**41.34**	**53.92**

TABLE 5.2: Comparison with the SOTA methods on ActivityNet Captions dataset.

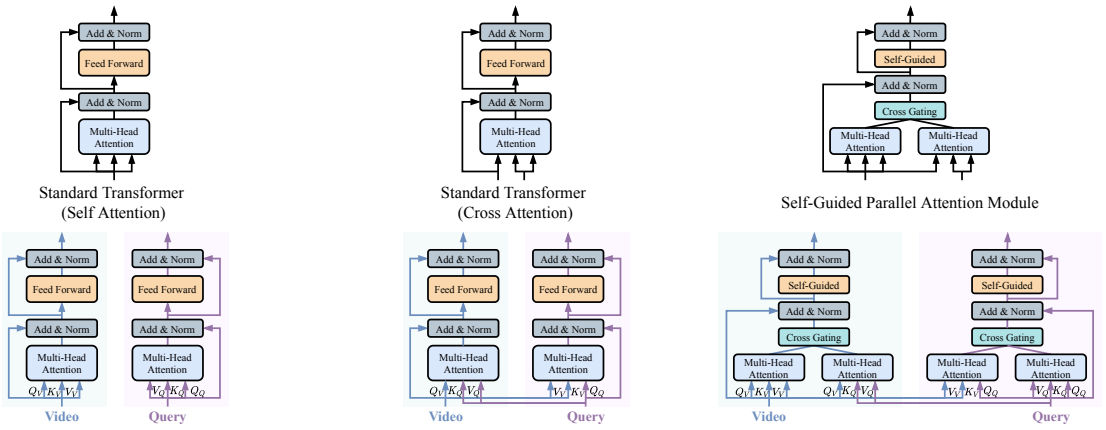
Methods	R@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
DEBUG	55.91	39.72	-	39.51
ExCL	*63.00*	43.60	24.10	-
SCDM	54.80	36.75	19.86	-
CBP	54.30	35.76	17.80	-
GDP	56.17	39.27	-	39.80
2D-TAN	59.45	44.51	*27.38*	-
TSP-PRL	56.08	38.76	-	39.21
TMLGA	51.28	33.04	19.26	-
VSLNet	**63.16**	43.22	26.16	*43.19*
DRN	-	*45.45*	24.36	-
LGI	58.52	41.51	23.07	-
SeqPAN	61.65	**45.50**	**28.37**	**45.11**

TABLE 5.3: Comparison with the SOTA methods on TACoS dataset.

Methods	R@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
TGN	21.77	18.90	-	-
ACL	24.17	20.01	-	-
DEBUG	23.45	11.72	-	16.03
SCDM	26.11	21.17	-	-
CBP	27.31	*24.79*	19.10	21.59
GDP	24.14	13.90	-	16.18
TMLGA	24.54	21.65	16.46	-
VSLNet	*29.61*	24.27	*20.03*	*24.11*
DRN	-	23.17	-	-
SeqPAN	**31.72**	**27.19**	**21.65**	**25.86**
2D-TAN	37.29	25.32	-	-
SeqPAN	**48.64**	**39.64**	**28.07**	**37.17**

5.3.2 Comparison with State-of-the-Arts

The results on the Charades-STA are summarized in Table 5.1. SeqPAN surpasses all baselines and achieves the highest scores over all metrics. Observe that the performance improvements of SeqPAN are more significant under more strict metrics. The results show that SeqPAN can produce more precise localization results. For instance, compared to LGI, SeqPAN achieves 5.86% absolute improvement by "R@1, IoU@0.7", and 1.40% by "R@1, IoU@0.5". Table 5.2 reports the results on ANetCaps. SeqPAN is superior to baselines and achieves the best performance on "R@1, IoU@0.7" and mean IoU. As reported in Table 5.3, similar observations hold



(a) Self Attention TRM (b) Cross Attention TRM (c) Self Guided Parallel Attention

FIGURE 5.4: The structures of standard transformer blocks and self-guided parallel attention module. Top: the structure of each module; Bottom: the parallel streams of encoding visual and textual inputs. (a) The standard transformer block with self-attention; (b) The standard transformer block with cross-attention; (c) The self-guided parallel attention (SGPA) module.

on TACoS. Note 2D-TAN [32] pre-processes the TACoS dataset, making it is slightly different from the original one. We also conduct experiments on their version for a fair comparison. SeqPAN outperforms the baselines over all evaluation metrics on both versions.

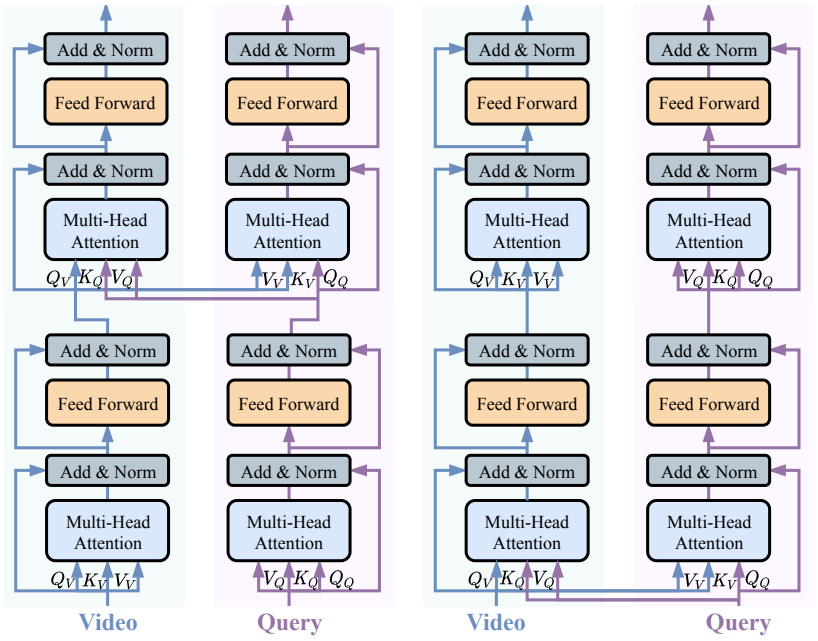
5.3.3 Analysis on Self-Guided Parallel Attention

The SGPA (see Figure 5.3) is a variant of transformer(TRM), designed for learning cross-modality interactions between visual and text feature. Here, we compare SGPA with standard TRMs. In general, two ways are mainly used to adopt the transformer block for multi-modal representation learning:

- Transformer block with the self-attention (Se-TRM), which encodes visual and textual inputs in separate streams, shown in Figure 5.4(a).
- Transformer block with the cross-attention (Co-TRM), which encodes both visual and textual inputs with interactions through co-attention, shown in Figure 5.4(b).

Several works [34, 91, 92] adopt Se-TRM to learn visual and textual representations in video grounding task. Se-TRM separately encodes each modality, it focuses on learning the refined unimodal representations within each modality for video and text respectively. Without any connection between two modalities, Se-TRM cannot use information from other modality to improve the representations. Co-TRM[6] is commonly used as a basic component in various

[6]It is also known as co-attentional, multi-modal or cross-modal transformer block in different works.



(a) SeCo-TRM Block (b) CoSe-TRM Block

FIGURE 5.5: The structures of SeCo-TRM and CoSe-TRM.

vision-language methods [19, 22]. Co-TRM relies on co-attention to learn the cross-modal representations for both visual and textual inputs. However, Co-TRM lacks the ability to encode self-attentive context within each modality.

The cascade of Se-TRM and Co-TRM is also used in recent vision-language models [19, 22, 23] to learn both unimodal and cross-modal representations. In general, there are two cascade forms: 1) stacking Co-TRM upon Se-TRM (SeCo-TRM) in Figure 5.5(a); and 2) stacking Se-TRM upon Co-TRM (CoSe-TRM) in Figure 5.5(b). These stacked TRMs learn the unimodal and cross-modal information in a sequence manner. Hence, their final outputs focus more on either the self-attentive contexts or cross-modal interactions. Our SGPA combines advantages of both Se-TRM and Co-TRM, but not through cascade. As shown in Figure 5.4(c), SGPA contains two parallel multi-head attention blocks. One block takes single modality as input and the other takes both modalities as inputs. Thus, SGPA is able to learn both unimodal and cross-modal representations simultaneously. Then, a cross-gating strategy is designed to fuse the self- and cross-attentive representations. We also employ a self-guided head to replace the feed forward layer in transformer block. This design implicitly emphasizes informative representations by measuring the confidence of each element.

To better reflect the performance of different TRMs, we remove the sequence matching component and only use a single block (*i.e.*, $N_{\text{SGPA}} = 1$) in the experiment. The results are reported in Table 5.4. Observe that SGPA is superior to TRMs on both datasets. Co-TRM

TABLE 5.4: Comparison between SGPA with standard transformer blocks on Charades-STA and ActivityNet Captions datasets, where Se-TRM is the transformer block with single modality inputs, and Co-TRM is with dual modality inputs, PA is the SGPA without self-guided head (*i.e.,* replaced by FFN). The scores in bracket denotes standard deviation.

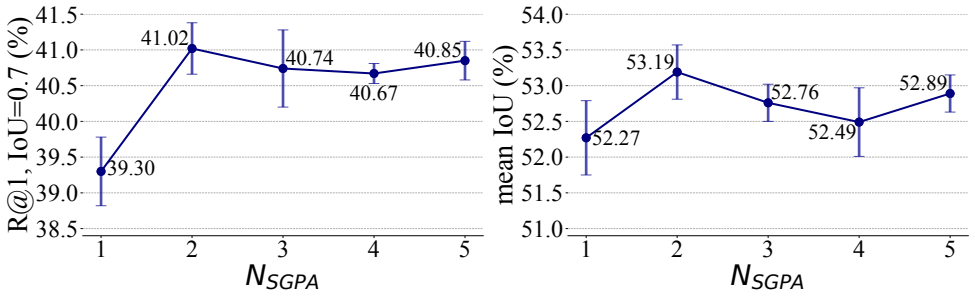
Methods	R@1, IoU $= \mu$		
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Charades-STA			
Se-TRM	68.84 (0.46)	51.92 (0.54)	34.58 (0.18)
Co-TRM	69.03 (0.49)	52.34 (0.50)	35.07 (0.32)
SeCo-TRM	69.11 (0.24)	52.63 (0.49)	35.17 (0.22)
CoSe-TRM	69.08 (0.26)	52.82 (0.43)	35.09 (0.50)
PA	69.21 (0.27)	54.37 (0.46)	36.22 (0.49)
SGPA	69.47 (0.32)	54.63 (0.43)	36.36 (0.24)
ActivityNet Captions			
Se-TRM	57.64 (0.38)	40.76 (0.35)	25.10 (0.30)
Co-TRM	57.39 (0.29)	40.55 (0.45)	24.85 (0.47)
SeCo-TRM	57.47 (0.38)	40.70 (0.24)	25.07 (0.21)
CoSe-TRM	57.72 (0.41)	40.85 (0.17)	25.16 (0.15)
PA	58.27 (0.13)	41.59 (0.24)	25.88 (0.28)
SGPA	58.40 (0.31)	41.72 (0.19)	26.07 (0.16)

TABLE 5.5: Ablation studies of sequence matching strategy in SeqPAN, where the values in bracket denote standard deviation.

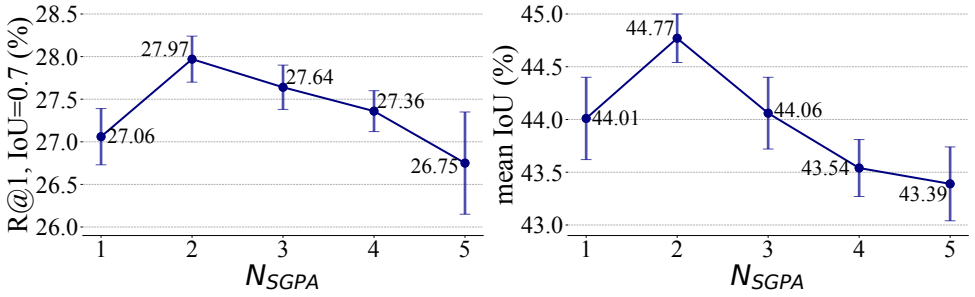
Method	sq-match		Charades-STA				ActivityNet Captions			
			R@1, IoU $= \mu$			mIoU	R@1, IoU $= \mu$			mIoU
	G	$\boldsymbol{E}_{\text{lab}}$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$		$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
SeqPAN w/ fb-match	-	-	70.27(0.75)	56.96(0.46)	38.95(0.27)	51.84(0.40)	*59.99*(0.25)	43.71(0.19)	26.72(0.29)	43.23(0.23)
SeqPAN w/o sq-match	✘	✘	69.62(0.54)	55.29(0.30)	36.71(0.48)	51.13(0.25)	59.03(0.35)	42.65(0.32)	26.29(0.13)	42.51(0.36)
SeqPAN w/ Gumbel	✔	✘	*71.64*(0.64)	*57.61*(0.26)	*39.26*(0.31)	*52.15*(0.45)	59.74(0.42)	*43.85*(0.35)	*27.12*(0.20)	*43.69*(0.24)
SeqPAN	✔	✔	**72.70**(0.51)	**60.15**(0.50)	**41.02**(0.36)	**53.19**(0.38)	**61.12**(0.39)	**45.09**(0.37)	**27.97**(0.27)	**44.77**(0.23)

performs better on Charades-STA but worse on ANetCaps comparing with Se-TRM. Compared to Se-TRM and Co-TRM, SGPA learns both self-modal contexts and cross-modal interactions, which is approximately equivalent to parallel connection of two modules.

Impact of SGPA block numbers N_{SGPA}. We now study the impact of SGPA block numbers on Charades-STA and ANetCaps. We evaluate five different values of N_{SGPA} from 1 to 5. The performance across the number of SGPA blocks in SeqPAN are plotted in Figures 5.6(a) and 5.6(b). Best performance is achieved at $N_{\text{SGPA}} = 2$ on both datasets. In general, along with increasing N_{SGPA}, the performance of SeqPAN first increases and then gradually decreases, on both datasets. We also note that performance on Charades-STA is not very sensitive to the setting of N_{SGPA}.



(a) Charades-STA



(b) ActivityNet Captions

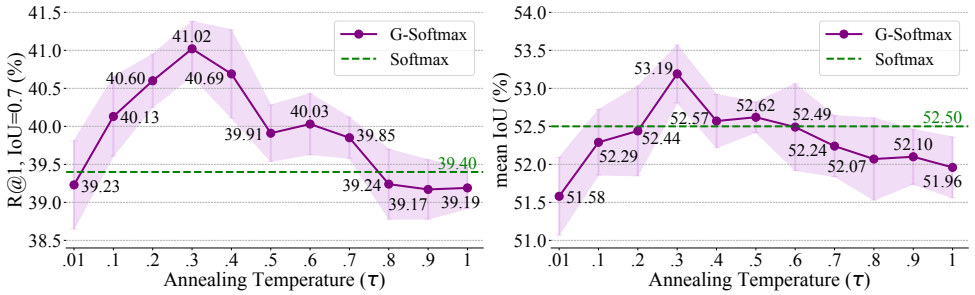
FIGURE 5.6: The impact of SGPA block numbers (N_{SGPA}) on the Charades-STA and ActivityNet Captions datasets.

5.3.4 Analysis on Sequence Matching

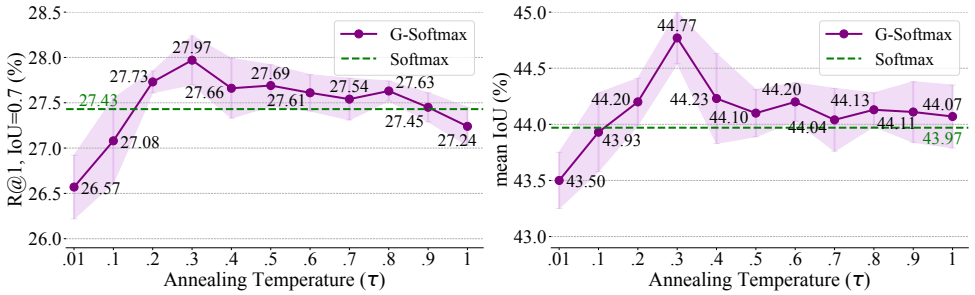
The conventional matching strategy [88, 91, 95] (denoted by fb-match) is to predict whether a frame is inside or outside of target moment, *i.e.,* foreground or background. In SeqPAN, we predict begin-, inside- and end-regions, and introduce label embeddings ($\boldsymbol{E}_{\text{lab}}$) to represent each region. The prediction process also uses the Gumbel-Max trick. In this experiment, we analyze the effects of label embeddings and Gumbel-Max trick in sequence matching.

Summarized in Table 5.5, both Gumbel-Max trick (denoted by G) and label embeddings contribute to the grounding performance improvement. In addition, consistent improvements are observed by incorporating G and $\boldsymbol{E}_{\text{lab}}$ into the model. SeqPAN is superior to SeqPAN w/ fb-match over all evaluation metrics. The performance improvements are more significant under more strict metrics. The results show that sq-match is more effective than the fb-match strategy. Regional indication of potential positions of start/end boundaries does help the model to produce accurate predictions.

Impact of Annealing Temperature τ. We then analyze the impact of annealing temperature τ of Gumbel-Softmax in sequence matching module. Gumbel-Softmax distributions are identical to a categorical distribution when $\tau \to 0^+$. With $\tau \to \infty$, its distribution is smooth. We evaluate 11 different τ values from 0.01 to 1.0, where 0.01 is used to approximate 0.0 since 0.0 is not



(a) Charades-STA



(b) ActivityNet Captions

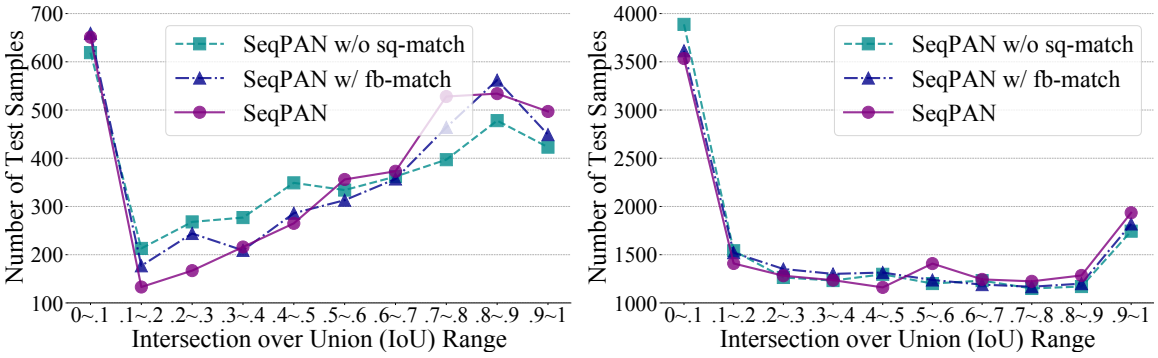
FIGURE 5.7: The impact of annealing temperature τ in sequence matching on Charades-STA and ActivityNet Captions datasets.

divisible. The results are compared against vanilla Softmax as a baseline. For vanilla Softmax, we multiply the probability distribution of labels with $\boldsymbol{E}_{\text{lab}}$, to aggregate label information into the visual representations.

Figure 5.7 plots the results of different τ's on Charades-STA and ANetCaps, respectively. We observe similar patterns on the four sets of results. The best performance is achieved when $\tau = 0.3$ over both metrics on both datasets. From Figure 5.7(a), when τ is too small or too large (*i.e.,* the probability distribution from Gumbel-Softmax becomes too sharp or too smooth), Gumbel-Softmax performs poorer than vanilla Softmax. This result suggests that a proper annealing temperature τ is crucial to achieve good performance. Similar observations hold on ANetCaps (see Figure 5.7(b)).

5.3.5 Qualitative Analysis

Figure 5.8 shows the number of predicted test samples within different IoU ranges on Charades-STA and ANetCaps. Here, we compare SeqPAN with two of its variants: (i) removal of sequence matching module, and (ii) replacement of sequence matching with fb-match. All three variants show similar patterns. Nevertheless, within the higher IoU ranges, *e.g.,* IoU ≥ 0.5 on both datasets, SeqPAN and the variant with fb-match outperform the variant without sequence matching. The results show that having auxiliary objectives (*e.g.,* foreground/background or



(a) Charades-STA (b) ActivityNet Captions

FIGURE 5.8: Plots of the number of predicted test samples within different IoU ranges on Charades-STA and ActivityNet Captions datasets.

sequential regions) is helpful in video grounding task. Results in Figure 5.8 also show that our sequence matching is more effective than fb-match, for highlighting the correction regions for predicting start/end boundaries.

Figure 5.9 depicts two video grounding examples from the ANetCaps dataset. From the two examples, the moments retrieved by SeqPAN are closer to the ground truth than that are retrieved by SeqPAN without utilizing the sq-match strategy. Besides, the start and end boundaries predicted by SeqPAN are roughly constrained within the pre-set potential start and end regions. In addition, the predicted sequence labels (PSL) in Figure 5.9 also reveal the weakness of sq-match strategy. The predicted labels by sq-match strategy are not continuous, where multiple start, inside, and end regions are generated. In consequence, the localizer may be affected by wrongly predicted regions and leads to inaccurate results. To further constrain the generated regions is part of our future work.

5.4 Summary

In this chapter, we have analyzed the importance of video-text interaction and the sparsity issue in the moment boundary prediction for many existing proposal-free methods. To address those issues, we presented a framework termed Parallel Attention Network with Sequence matching (SeqPAN). In SeqPAN, we first design a parallel attention module to improve the multimodal representation learning by capturing both self- and cross-modal attentive information simultaneously. On top of the parallel attention module, we ameliorate the standard context-query attention and propose a video-query integration module, which further enhances the cross-modal interactions between visual and textual features. In addition, we propose a sequence matching strategy, which splits the video into four different regions, *i.e.,* begin, inside, end and

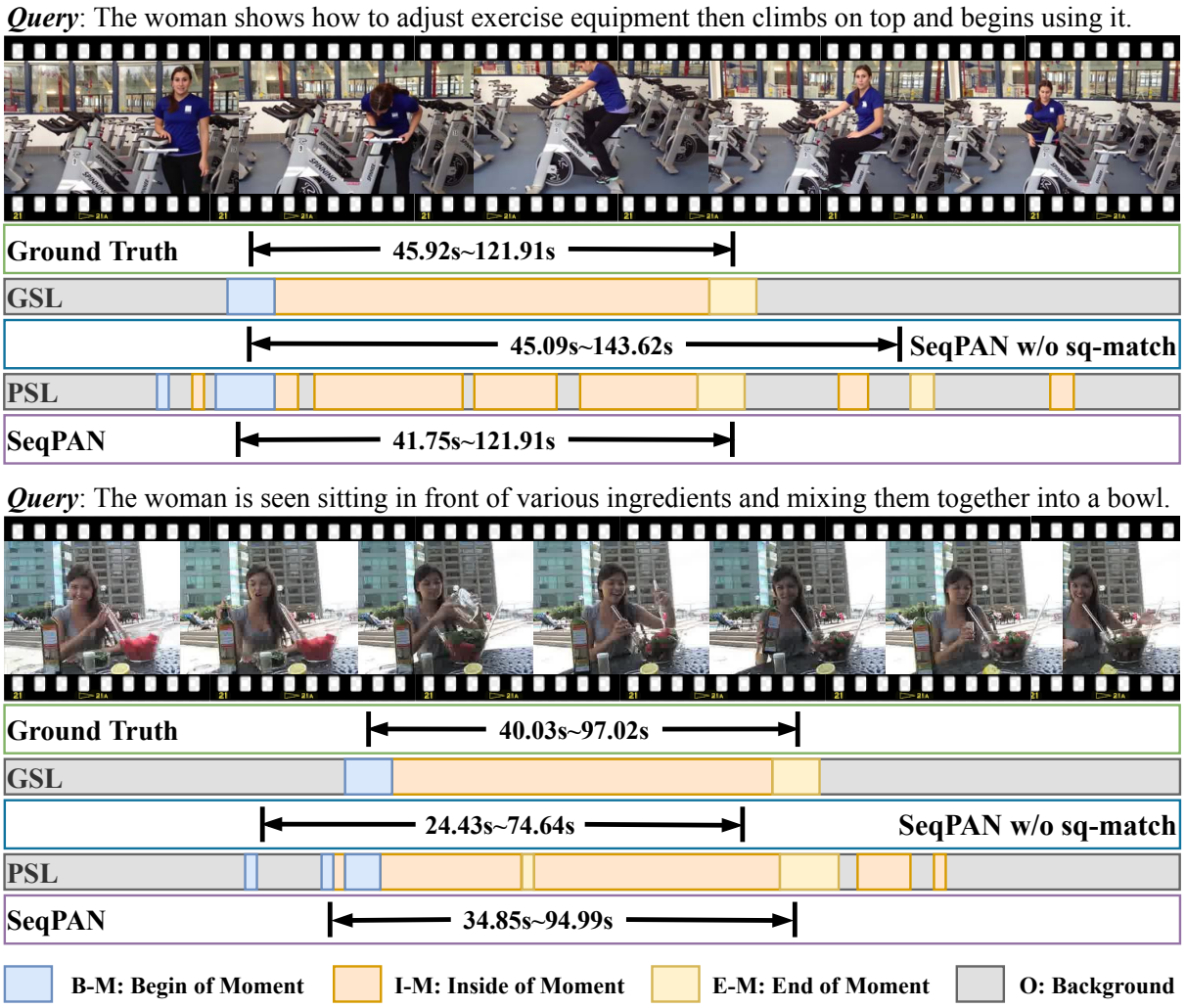


FIGURE 5.9: Qualitative results of SeqPAN and SeqPAN w/o sq-match on ANetCaps. "GSL" is the ground truth sequence labels; "PSL" is the predicted labels by sq-match of SeqPAN.

background, and explicitly indicates the potential start and end regions of the target moment to allow the localizer precisely predicting the boundaries. Compared to VSLBase and VSLNet, SeqPAN also follows the canonical span-based question answering framework, but it incorporate the concepts of named entity recognition and equip with more sophisticated multimodal interaction module to boost performance. Through extensive experimental studies, we show that SeqPAN outperforms the state-of-the-art methods on three benchmark datasets; and both the proposed parallel attention and sequence matching modules contribute to the grounding performance improvement.

Chapter 6

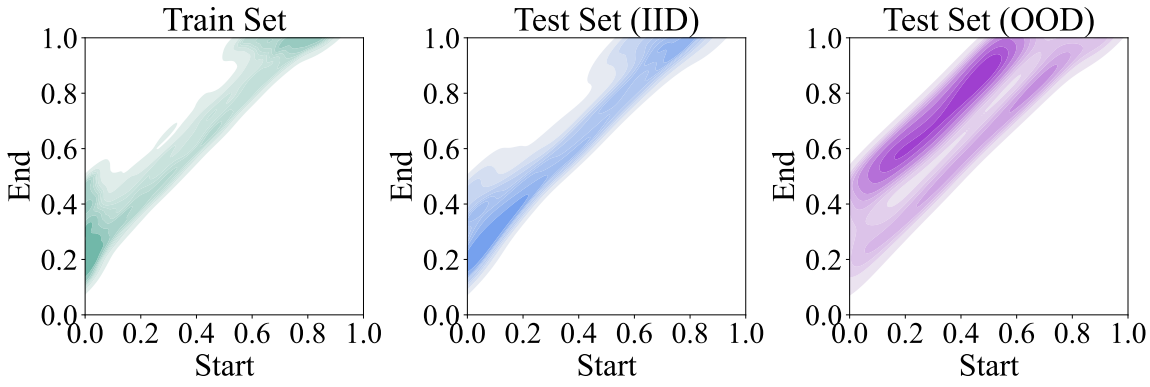
Towards Debiasing TSGV

6.1 Introduction

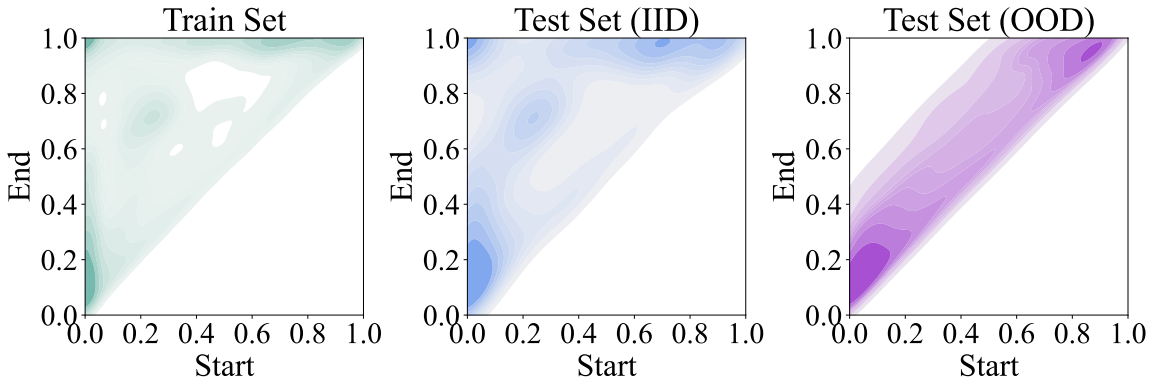
The previous chapters have explored to formulate TSGV as a multimodal span-based question answering problem, and developed a standard span-based QA framework to solve TSGV. However, recent studies [174, 193] reveal that: 1) Substantial distribution bias exists in the benchmark datasets; 2) Many TSGV models rely on exploiting the statistical regularities of annotation distribution for moment retrieval, to achieve good performance. To illustrate the annotation distribution bias in the benchmark datasets, we use Charades-CD dataset[1] as an example and visualize its annotation distribution, as shown in Figure 6.1(a). The "Start" and "End" axes denote the normalized start and end time points of the annotated moments in videos. Observe that many annotated moments in the train set of Charades-CD locate in the normalized temporal region of $0.2 \sim 0.4$ of the video length. Hence, a TSGV model could make a good guess of start/end time points, even without taking into consideration the input video and language query. Consequently, the existing TSGV methods achieve impressive performance on the independent-and-identical distribution (iid) test set, but fail to generalize on the out-of-distribution (ood) test set.

The aforementioned studies [174, 193] well analyze the issue of distribution bias in TSGV datasets and models. However, they do not attempt to address the bias. To solve this problem, Yang et al. [194] propose a causality-inspired TSGV framework that builds a structural casual model [119] to disentangle confounding effects of moment location from visual content for correct prediction. Luo et al. [159] devise a self-supervised approach to solve TSGV with pseudo label generation, which does not rely on the original biased annotations. Instead of

[1]Charades-CD dataset is built from Charades-STA dataset, Yuan et al. [174] redesign the Charades-STA to form Charades-CD dataset for the purpose of evaluating model biasing, with dedicated iid test set and ood test set. Details of Charades-CD and ActivityNet-CD are introduced in the experiments.



(a) Charades-CD



(b) ActivityNet-CD

FIGURE 6.1: Moment annotation distributions of the Charades-CD and ActivityNet-CD datasets, where "Start" and "End" axes represent the normalized start and end time points, respectively. Deeper color represents larger density (*i.e.,* more annotations) in the dataset.

designing sophisticated architectures, in this chapter[2], we aim to mitigate bias in TSGV models with simple yet effective strategies. Specifically, we propose two debiasing strategies, data debiasing and model debiasing. The data debiasing strategy is model-agnostic, which could be applied to various TSGV models. The model debiasing strategy, inspired from visual question answering debiasing [195, 196], is designed for proposal-free TSGV models.

From data perspective, bias is caused by the imbalanced distribution of moment annotations, *i.e.,* many annotations are concentrated in several regions as shown in Figure 6.1. To mitigate the bias, we artificially re-balance the moment annotations via data augmentation, *i.e.,* creating more samples through video truncation. Specifically, we partition each training video into multiple non-overlapping clips, and gradually cut background clips (*i.e.,* clips that do not contain target moment) to form new videos. Each newly created video hence has a different

[2]This chapter is submitted as Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "Towards Debiasing Temporal Sentence Grounding in Video". Under review [37].

start/end time point from the original training sample. Besides, each query corresponds to multiple videos of different lengths after data debiasing. This oversampling implicitly encourages a TSGV model to focus more on the interactions between the language query and target moment.

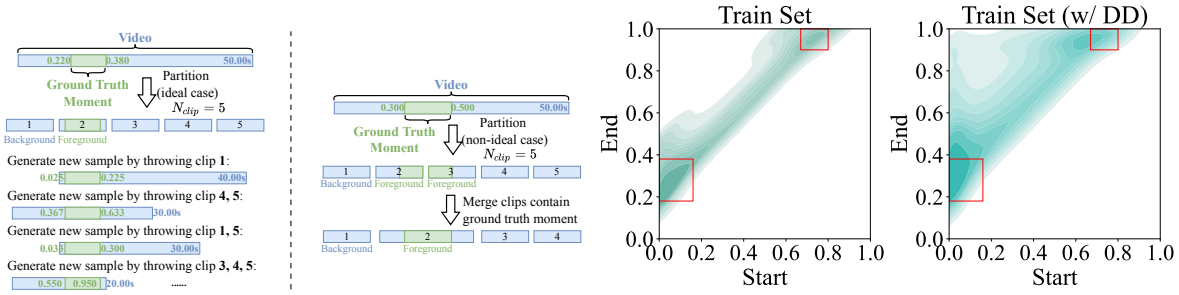
The model debiasing strategy is inspired by the question-only branch in visual question answering (VQA) debiasing [195, 196]. Bias in VQA comes from the suspicious correlations between answer occurrences and certain patterns of questions, *e.g.,* color of a banana is always answered as "yellow" regardless the input image contains a yellow or green banana. Bias in TSGV, on the other hand, is caused by the regularities in the moment boundaries. That is, a TSGV model with video- or query-only input could achieve fairly good performance by making a good guess of the distribution bias. Thus, our model debiasing is conceptually different from the strategies in VQA models. To debias, we add two unimodal models, *i.e.,* video- and query-only branches, in addition to the TSGV model. Both unimodal models learn to capture the bias, and to disentangle bias from the TSGV model by adjusting losses to compensate for biases dynamically. Specifically, the gradients backpropagated through TSGV model are reduced for biased examples and are increased for the less biased after loss adjustment.

We evaluate the proposed data and model debiasing strategies on the Charades-CD and ActivityNet-CD datasets, by using VSLNet as the base model. Experimental results demonstrate their ability of well generalization on out-of-distribution test set. With both debiasing strategies, VSLNet achieves further improvements against the base model.

6.2 Debiasing TSGV Framework

We first recall the formulation of TSGV task in feature space. Then we elaborate the proposed data and model debiasing strategies. The data debiasing strategy is model-agnostic, hence can be applied to any TSGV model in principle. The proposed model debiasing strategy is applicable to proposal-free models. Lastly, we use VSLNet [34] as a backbone to illustrate how to apply the two debiasing strategies.

We denote an untrimmed video with T frames as $V = [f_t]_{t=0}^{T-1}$, language query with m words as $Q = [q_i]_{i=0}^{m-1}$, τ_s and τ_e as the start and end time of ground truth moment. The video V is then split into n units and encoded into visual features $\boldsymbol{V} = [\boldsymbol{v}_i]_{i=0}^{n-1} \in \mathbb{R}^{n \times d_v}$ with pre-trained visual feature extractor. The query Q is initialized by the pre-trained word embeddings, such as GloVe [26], as $\boldsymbol{Q} = [\boldsymbol{w}_i]_{i=0}^{m-1} \in \mathbb{R}^{m \times d_q}$. The $\tau_{s(e)}$ are mapped to the corresponding indices $a_{s(e)}$ in the feature sequence, where $0 \leq a_s \leq a_e \leq n-1$. The goal of TSGV is to localize the moment starting at a_s and ending at a_e.



(a) An illustration of the data debiasing strategy with ideal and non-ideal cases.

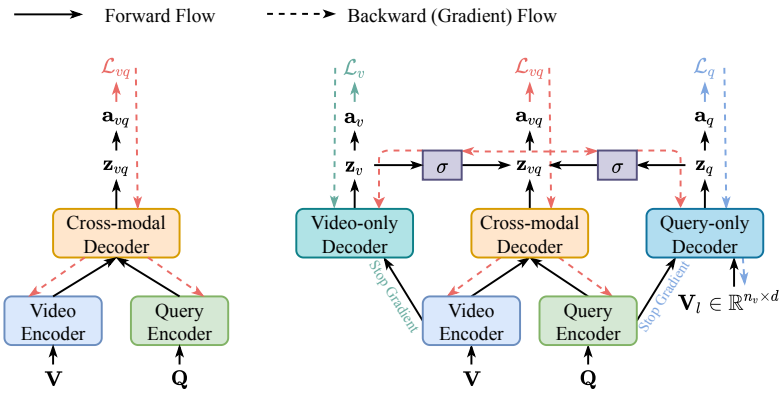
(b) The train set distribution before and after the data debiasing (DD) strategy.

FIGURE 6.2: The illustration of data debiasing (DD) strategy and an example of augmented train set distribution for Charades-CD dataset. Note the decimals with green color in (a) represent the normalized start and end time according to the position of the ground truth moment and the video length. The red rectangles in (b) highlight the regions of biased samples in Charades-CD train set before and after the data debiasing (DD) strategy.

6.2.1 Data Debiasing

As shown in Figure 6.1, the iid test set shares identical distribution of start/end positions with the train set, while the ood test has an entirely different distribution. From a data perspective, this issue is caused by the fact that the train set does not cover many moment annotations that start and end at different positions of the untrimmed video. Thus, we propose a simple data debiasing (DD) strategy to include more annotations with varying start/end positions.

Ideally, the moment annotations of the train set should uniformly distribute in the upper triangle area in the distribution plot (see Figure 6.1). To this end, we oversample annotations through video truncation. As illustrated in Figure 6.2(a), for each video-query pair, we first partition the video into multiple non-overlapping clips. If a clip overlaps with the ground truth moment, then it is regarded as foreground, otherwise as background. If the ground truth moment happens to be partitioned into multiple clips, then these clips are merged back to ensure the ground truth moment is unaffected. Then we gradually truncate the video to generate new videos by throwing background clips from both ends. The newly generated videos hence are sub-clips of the original video, with their overall length reduced, but all contain the ground truth moment. The relative positions of the start and end boundaries of the ground truth moment are then well distributed. Because the correspondence between the language query and ground truth moment does not change, this oversampling process makes each query correspond to multiple videos of different lengths. It is worth noting that a query is paired with multiple copies of the corresponding video (original/clipped versions) after DD, thus, there is mostly no information loss from the perspective of all samples as the dropped background still exists in the original or other clipped versions. In this way, data debiasing implicitly encourages the TSGV model



(a) Standard Proposal-free TSGV Model

(b) Proposal-free TSGV Model with Model Debiasing (MD)

FIGURE 6.3: A standard proposal-free VMR model (a), and model debiasing strategy (b).

to learn the multimodal matching between the query and the target moment, with different amounts of irrelevant content in videos. Note that, we do not rotate or permutate background clips, to ensure semantic continuity in the generated videos.

We use the train set of Charades-CD as an example to illustrate the effect of DD in Figure 6.2(b). After data debiasing, the moment annotation distribution of the augmented train set spreads to a much larger area than that of the original train set. Besides, the proportion of biased samples (highlighted in red rectangle in the figure) in the entire training samples reduces from 38.83% to 21.80%.

6.2.2 Model Debiasing

We propose model debiasing for proposal-free TSGV models, which directly learn cross-modal interactions between video and query, and directly predict boundaries of target moment. Next, we use the computation process of span-based TSGV model to present our debiasing strategy.

A typical span-based model consists of a video encoder $e_v : \boldsymbol{V} \in \mathbb{R}^{n\times d_v} \rightarrow \bar{\boldsymbol{V}} \in \mathbb{R}^{n\times d}$, a query encoder $e_q : \boldsymbol{Q} \in \mathbb{R}^{m\times d_q} \rightarrow \bar{\boldsymbol{Q}} \in \mathbb{R}^{m\times d}$, a cross-modal interaction module $m_{vq} : \bar{\boldsymbol{V}} \times \bar{\boldsymbol{Q}} \rightarrow \boldsymbol{H}_{vq} \in \mathbb{R}^{n\times d}$, and an answer predictor $g_{vq} : \boldsymbol{H}_{vq} \rightarrow \boldsymbol{z}^{vq} = \{\boldsymbol{z}_s^{vq}, \boldsymbol{z}_e^{vq}\} \in \mathbb{R}^{n\times 2}$. The overall architecture is shown in Figure 6.3(a), where we combine the cross-modal interaction module m_{vq} and answer predictor g_{vq} as cross-modal decoder $\varphi_{vq} : \bar{\boldsymbol{V}} \times \bar{\boldsymbol{Q}} \rightarrow \boldsymbol{z}^{vq} = \{\boldsymbol{z}_s^{vq}, \boldsymbol{z}_e^{vq}\} \in \mathbb{R}^{n\times 2}$ for better visualization, and $\boldsymbol{a}_{s(e)}^{vq} = \texttt{Softmax}(\boldsymbol{z}_{s(e)}^{vq})$. The training objective is defined as:

$$\mathcal{L}_{\text{vq}} = \frac{1}{2} \times \big[f_{\text{XE}}(\boldsymbol{a}_s^{vq}, \boldsymbol{y}_s) + f_{\text{XE}}(\boldsymbol{a}_e^{vq}, \boldsymbol{y}_e)\big], \tag{6.1}$$

where f_{XE} is cross-entropy function, $\boldsymbol{y}_s$ and $\boldsymbol{y}_e$ are the corresponding one-hot labels for start (a_s) and end (a_e) boundaries, respectively.

Due to the existence of substantial data distribution biases in TSGV datasets, TSGV models tend to rely on the statistical regularities to generate predictions even without having to consider the video and query inputs [193]. Hence, a TSGV model with unimodal input (*i.e.,* either video or query) could also achieve fair performance by capturing the distribution biases in TSGV datasets. Inspired by VQA debiasing [195, 196], we adapt unimodal models, *i.e.,* video-only module and query-only module, as separate branches and integrate them into TSGV model, illustrated in Figure 6.3(b). The unimodal models learn to explicitly capture the distributional bias from TSGV benchmarks, and force the TSGV model to focus on learning cross-modal interaction between video and query by removing the biased information from its answer predictions.

It is worth noting that VQA usually suffers from language prior bias. For example, whenever a query mentions "banana", systems will answer "yellow" regardless whether image contains "yellow" or "green" banana. In this sense, VQA methods only adopt the query-only branch for debiasing. In contrast, for TSGV, bias has negative effects on both video and query sides. Thus, we have to detect and remove the effects of bias from both sides.

Specifically, given the encoded visual ($\bar{\boldsymbol{V}}$) and textual ($\bar{\boldsymbol{Q}}$) features, the TSGV model retrieves the answer with cross-modal decoder as:

$$\boldsymbol{z}^{vq} = g_{vq}(m_{vq}(\bar{\boldsymbol{V}}, \bar{\boldsymbol{Q}})). \tag{6.2}$$

Because the video-only branch does not contain query input, the video-only decoder generate answer by:

$$\boldsymbol{z}^{v} = g_{v}(\bar{\boldsymbol{V}}). \tag{6.3}$$

In other words, the video-only branch directly takes video as input to predict the possible target moment without the assistance of language query. If the benchmark datasets are uniformly distributed, the video-only branch cannot learn correct patterns from the dataset, and degenerate to random guess. However, due to the existence of strong distributional bias, the video-only branch could easily capture the bias to encode the suspicious correlations and achieve fair performance via iteratively model updating.

For query-only branch, as TSGV task requires to retrieve start and end boundaries of target moment, we follow Otani et al. [193] to replace $\bar{\boldsymbol{V}}$ with a learnable feature sequence $\boldsymbol{V}_l \in \mathbb{R}^{n_v \times d}$ in the same shape, to simulate visual input. The answer is computed as:

$$\boldsymbol{z}^{q} = g_{q}(m_{q}(\boldsymbol{V}_l, \bar{\boldsymbol{Q}})). \tag{6.4}$$

Similar to the video-only branch, the query-only branch also captures the bias to obtain correct predictions through model learning. Note that, the structures of m_q and m_{vq} are the same, but their parameters are not shared. Also, g_v, g_q, and g_{vq} are the same.

Since both video-only and query-only branches encode the bias information through model learning, we can explicitly remove the learned bias by disentangling the biased representations from the TSGV model. Specifically, with the biased prediction $\boldsymbol{z}_q$ and $\boldsymbol{z}_v$, we modify the prediction of TSGV model $\boldsymbol{z}_{vq}$ as:

$$\hat{\boldsymbol{z}}_{vq} = \boldsymbol{z}^{vq} \odot \sigma(\boldsymbol{z}^{q}) \odot \sigma(\boldsymbol{z}^{v}), \tag{6.5}$$

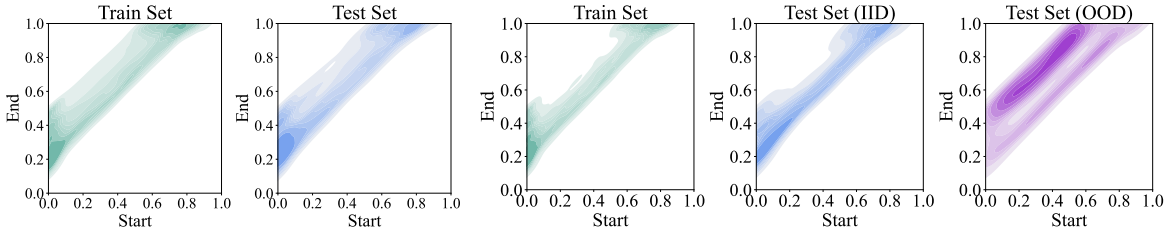
where $\odot$ denotes element-wise multiplication and σ is Sigmoid activation to map the biased prediction between 0 and 1. The key of Equation (6.5) is to dynamically alter the loss, by modifying predictions of the TSGV model, to prevent the model to pick up distribution biases from the dataset.

Given a biased sample, both unimodal branches and the TSGV model tend to generate high confidence to the correct answer and low confidence to others. The confidence score of the correct answer predicted by the TSGV model will be further increased with Equation (6.5). Thus, the loss from a biased sample is much smaller. Accordingly, since the gradients are generally proportional to loss, the gradient backpropagated through the TSGV model is very small, reducing the importance of this biased sample in training. On the contrary, the importance of a non-biased sample will be increased. The reason is, all the models tend to assign low confidence to the correct answer, hence the score is further decreased through Equation (6.5). With a large loss, the TSGV model is forced on learning non-biased samples. To be specific, the model debiasing is to minimize overall risk. During training, we further increase confidence of correct predictions for biased samples with help of unimodal branches. Thus, model will guess those samples are well learned and pay less attention to them. In contrast, for unbiased samples, we suppress confidence of correct predictions, which makes model pay more attention to learn them to increase confidence for reducing risk. In short, a higher confidence of sample leads to lower cross-entropy loss, *i.e.,* the gradients are smaller, thereby reducing their importance. Vice versa for the samples with lower confidences.

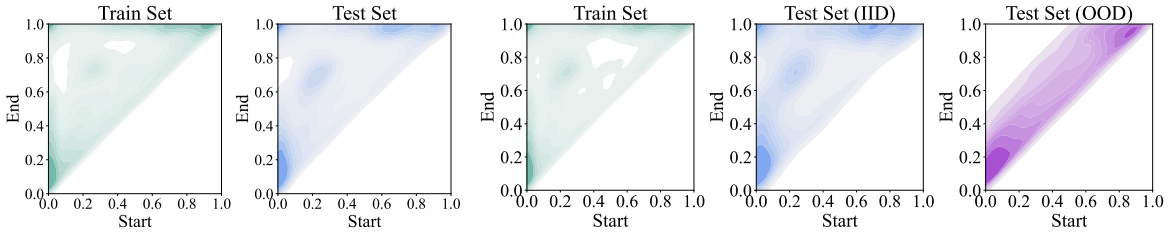
Finally, we jointly optimize the two unimodal models and the TSGV model as:

$$\mathcal{L}_{all} = \mathcal{L}_{vq} + \mathcal{L}_{q} + \mathcal{L}_{v}. \tag{6.6}$$

Note that, both $\mathcal{L}_q$ and $\mathcal{L}_v$ do not optimize the query (e_q) and video (e_v) encoders, *i.e.,* the "stop gradient" in Figure 6.3(b), to prevent them from directly learning distribution biases. $\mathcal{L}_{vq}$ is



(a) Charades-STA. (b) Charades-CD.



(c) ActivityNet Captions. (d) ActivityNet-CD

FIGURE 6.4: The moment annotation distributions of train and test sets for the Charades-STA, Charades-CD, ActivityNet Captions and ActivityNet-CD datasets, respectively.

TABLE 6.1: Statistics of the TSGV benchmark datasets.

Dataset	Split	# Videos	# Annotations
Charades-CD	train	4,564	11,071
	val	333	859
	test-iid (iid)	333	823
	test-ood (ood)	1,442	3,375
ActivityNet-CD	train	10,984	51,415
	val	746	3,521
	test-iid (iid)	746	3,443
	test-ood (ood)	2,450	13,578
Charades-STA	train	5,338	12,408
	test	1,334	3,720
ActivityNet Captions	train	10,009	37,421
	test	4,917	34,536

used to optimize all the modules except the learnable feature sequence for query-only decoder. During inference, only the TSGV model is used.

6.3 Experiments

6.3.1 Experimental Settings

Datasets. We conduct experiments on the Charades-CD and the ActivityNet-CD datasets, prepared by [174]. The two datasets originate from the Charades-STA [27] and the ActivityNet Captions [165] datasets, respectively. The train/test instances are reorganised by [174] into train, val, iid test and ood test sets. The moment annotation distributions of the evaluated datasets, *i.e.,* Charades-STA [27], Charades-CD [174], ActivityNet Captions [165] and ActivityNet-CD [174] are depicted in Figure 6.4. Robustness of TSGV models are measured by the performance gap between their iid and ood test sets. Note that, the number of samples in the iid test sets for both Charades-CD and ActivityNet-CD datasets are small, which lead to large variances and may not accurately reflect the performance of our proposed strategies. For this reason, we also conduct experiments on original Charades-STA and ActivityNet Captions datasets to further validate our strategies. Statistics of the datasets are summarized in Table 6.1.

Evaluation Metric. The measure R@n, IoU@μ denotes the percentage of test samples that have at least one result whose Intersection over Union (IoU) with ground truth is larger than μ in top-n predictions. The measure dR@n, IoU@μ, is the discounted-R@n, IoU@μ proposed by Yuan et al. [174]. This metric introduces two discount factors computed from the boundaries of predicted and ground truth moments to restrain over-long predictions. We also report mIoU, which is the average IoU over all test samples. We set $n = 1$ and $\mu \in \{0.3, 0.5, 0.7\}$.

Implementation. We use VSLNet [34] as the backbone model to evaluate both debiasing strategies, due to its simple architecture and prominent performance. VSLNet consists of a transformer-based module as video encoder e_v, a transformer-based module as query encoder e_q, a video-query co-attention layer as cross-modal reasoning module m_{vq}, and the stacked LSTMs as answer predictor g_{vq}. Here, we replace the stacked LSTMs with stacked transformer blocks [177] for faster training and inference. In our implementation, the VSLNet is regarded as the standard TSGV model. Then we initialize a new answer predictor g_v with stacked transformer blocks as video-only decoder. The query-only decoder is constructed by creating a new cross-modal reasoning module m_q and a new answer predictor g_q. As mentioned before, the m_q and m_{vq} have the same structure but their parameters are not shared, while g_v, g_q and g_{vq} are the same. Finally, the VSLNet, query-only decoder and video-only decoder are integrated together following Figure 6.3(b).

We utilize the 300d GloVe [26] vectors to initialize lowercase words in query Q. For video V, we follow Yuan et al. [174] with the pre-trained I3D features [25] for Charades-CD and

TABLE 6.2: The performance (%) of unimodal models and VSLNet on Charades-CD dataset.

Split	Method	R@1, IoU@μ		dR@1, IoU@μ		mIoU
		μ=0.5	μ=0.7	μ=0.5	μ=0.7	
iid	Q-only	29.65	17.89	26.98	17.06	34.67
	V-only	30.38	20.29	27.94	19.28	33.24
	VSLNet	60.51	41.07	56.12	39.25	54.39
ood	Q-only	15.17	5.99	12.96	5.54	20.87
	V-only	16.71	6.96	14.18	6.48	20.03
	VSLNet	48.18	28.89	43.29	27.20	45.56

Charades-STA, and the pre-trained C3D features [24] for ActivityNet-CD and ActivityNet Captions. For all benchmark datasets, we set the maximal visual sequence feature length as 128. For the model parameters, we follow Zhang et al. [34] and use the hidden size of 128 for all hidden layers, the head size of 8 for multi-head attention, and the kernel size of 7 for convolutions. For data debiasing, we empirically partition each video into $N_{clip} = 5$ clips. Adam [181] optimizer is used with batch size of 16 and learning rate of 0.0005 for training. The model is trained by 100 epochs in total with early stopping tolerance of 10 epochs.[3]

6.3.2 Bias in Backbone Model

We first study the performance of backbone model on Charades-CD dataset. In particular, we separately train the standard VSLNet, query-only model (Q-only), and video-only model (V-only). The results are summarized in Table 6.2. Observe that both Q-only and V-only models achieve fair performance on iid test set, but poorer on ood test set, *e.g.,* 17.06% on iid versus 5.54% on ood for Q-only model over the dR@n, IoU@0.7 measure. Similar observation holds on the standard VSLNet. We conclude that Q-only and V-only models can well capture the distributional bias, and the backbone model fails to generalize well on test set with ood samples.

6.3.3 Impact of data and model debiasing

The results of applying data debiasing (DD) and model debiasing (MD) strategies are summarized in Table 6.3, where "all" means all samples in both iid and ood test sets. On iid test sets, performance drop is observed after applying DD and MD strategies. One reason is that both DD and MD prevent the model from exploiting distribution bias in training. Another reason is that the number of samples in iid test sets are small, which leads to large variances and may not well reflect the accurate effects of the proposed debiasing strategies. On contrast, on ood

[3] All experiments are conducted on a workstation with dual NVIDIA GeForce RTX 3090 GPUs.

TABLE 6.3: The performance (%) of applying data debiasing (DD) and model debiasing (MD) strategies on top of VSLNet on the Charades-CD and ActivityNet-CD datasets.

Split	Method	Charades-CD							ActivityNet-CD						
		R@1, IoU@μ			dR@1, IoU@μ			mIoU	R@1, IoU@μ			dR@1, IoU@μ			mIoU
		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7	
iid	VSLNet	**75.09**	**60.51**	**41.07**	66.72	**56.12**	**39.25**	**54.39**	**63.18**	**49.16**	**32.37**	**52.41**	**43.40**	**31.37**	**47.57**
	+DD	74.73	59.54	38.40	65.79	55.01	36.78	53.42	62.08	47.95	30.10	51.81	42.45	28.28	45.90
	+MD	73.39	59.17	39.13	65.22	54.69	37.41	53.26	58.73	45.10	29.32	50.23	40.51	27.60	43.62
	+DD+MD	74.85	59.97	40.55	**67.19**	55.66	38.87	53.92	60.65	45.84	29.94	51.26	40.88	28.11	44.71
ood	VSLNet	65.93	48.18	28.89	55.85	43.29	27.20	45.56	41.71	23.31	12.06	32.99	20.96	11.59	29.95
	+DD	**71.88**	54.84	33.69	60.68	49.02	31.69	49.60	42.21	26.81	14.33	35.12	24.29	13.45	30.19
	+MD	70.79	54.90	33.30	60.59	49.26	31.64	49.49	42.43	**27.70**	15.01	36.08	**25.33**	14.41	30.55
	+DD+MD	70.10	**55.82**	**34.70**	**60.81**	**50.37**	**32.70**	**50.30**	**42.78**	27.33	**15.10**	**36.16**	25.05	**14.67**	**30.56**
all	VSLNet	67.72	50.60	31.28	57.98	45.81	29.56	47.30	46.02	28.49	16.14	36.89	25.47	15.66	33.19
	+DD	**72.44**	55.76	34.61	61.68	50.19	32.81	50.57	46.19	29.68	17.23	38.18	26.77	16.43	33.31
	+MD	70.25	55.79	35.30	60.93	50.63	33.33	50.40	45.70	**31.19**	17.88	38.92	**28.38**	17.05	33.17
	+DD+MD	71.03	**57.03**	**36.06**	**62.06**	**51.40**	**34.08**	**51.01**	**46.38**	31.09	**18.14**	**39.21**	28.25	**17.43**	**33.42**

TABLE 6.4: The performance (%) of applying data debiasing (DD) and model debiasing (MD) strategies on top of VSLNet on the Charades-STA and ActivityNet Captions datasets.

Method	Charades-STA							ActivityNet Captions						
	R@1, IoU@μ			dR@1, IoU@μ			mIoU	R@1, IoU@μ			dR@1, IoU@μ			mIoU
	μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7	
VSLNet	71.18	55.73	36.83	62.04	51.22	35.08	51.31	59.30	42.50	26.01	48.13	37.16	24.29	43.01
+DD	**71.75**	55.38	38.90	**62.49**	51.12	37.10	51.96	**59.68**	42.21	26.19	48.18	36.88	24.45	43.31
+MD	70.19	55.75	37.80	61.72	51.55	36.05	51.17	58.46	42.14	26.74	47.96	37.16	25.03	42.88
+DD+MD	70.05	**58.49**	**39.92**	62.27	**53.97**	**38.06**	**52.18**	59.66	**42.89**	**26.99**	**48.99**	**37.45**	**25.26**	**43.46**

test sets, both DD and MD significantly improve the performance. In addition, by combining DD and MD, further improvements are observed on ood test sets. Thus, the performance gaps between iid and ood sets are reduced further. On the combined test samples (iid + ood), DD and MD together bring performance increase on both datasets.

Table 6.4 reports results on original Charades-STA and ActivityNet Captions datasets. Note moment annotation distributions of train and test sets are almost the same for both Charades-STA and ActivityNet Captions datasets [174], and the number of test samples in both datasets is relatively large. Observe that small improvements are observed after applying DD and MD strategies on both datasets. The DD and MD encourage TSGV model to focus more on exploiting cross-modal reasoning between video and query, which contributes to the performance improvements.

In general, without debiasing, the standard TSGV model can easily capture shortcuts for good predictions. However, the shortcuts do not reveal correct cross-modal interactions, which leads to poor performance on ood. With debiasing, shortcuts are blocked; the results can no longer be easily predicted based on shortcuts, but learned cross-model interaction. Thus, the

FIGURE 6.5: Results (in %) of dR@1, IoU@0.7 and the performance gap (%) between iid and ood test sets for SOTA TSGV models, VSLNet and proposed debiasing strategies on Charades-CD and ActivityNet-CD.

overall generalization ability of TSGV model is improved with debiasing, leading to better performance on test sets.

6.3.4 Comparison with the State-of-the-Arts

Although two recent work proposes to solve bias issue of TSGV, Yang et al. [194] do not conduct experiments on Charades-CD or ActivityNet-CD datasets. Instead, they synthesize ood samples from original datasets, *i.e.,* Charades-STA and ActivityNet Captions, by inserting a random-generated video clip at the beginning of videos. While Luo et al. [159] solve bias issue with self-supervised learning under the semi-supervised setting. Due to the differences in datasets and settings, their methods are not directly comparable to our proposed method. Figure 6.5 depicts the dR@1, IoU@0.7 results of SOTA models on Charades-CD and ActivityNet-CD datasets. We compare with 1) proposal-based methods, CTRL [27], ACRN [61] and SCDM [31]; 2) proposal-free methods, ABLR [88], 2D-TAN [32] and DRN [94]; 3) RL-based method TSP-PRL [125]. Results of these models are reported by Yuan et al. [174]. In general, our backbone model, and the version with DD and MD strategies, are superior to the compared SOTA models on both datasets.

Although the performance of these models varies greatly, we are more interested in their performance gap between iid and ood test sets. Specifically, we define the performance gap as:

$$p_{gap} = \frac{|s_{ood} - s_{iid}|}{s_{iid}} \times 100\% \tag{6.7}$$

TABLE 6.5: The performance (%) of VSLNet with data debiasing (DD) strategy over different number of clip N_{clip} on the Charades-CD and ActivityNet-CD datasets.

Split	N_{clip}	Charades-CD							ActivityNet-CD						
		R@1, IoU@μ			dR@1, IoU@μ			mIoU	R@1, IoU@μ			dR@1, IoU@μ			mIoU
		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7	
iid	4	74.00	59.17	39.13	65.82	54.75	**37.46**	53.75	61.49	46.49	28.90	51.36	41.15	27.16	45.17
	5	**74.73**	59.54	38.40	**65.79**	55.01	36.78	53.42	**62.08**	**47.95**	30.10	**51.81**	**42.45**	28.28	**45.90**
	6	74.24	**61.24**	**38.52**	65.67	**56.46**	36.86	**53.86**	61.92	46.30	**30.45**	51.46	41.21	**28.61**	45.63
ood	4	70.43	54.28	33.24	59.89	48.63	31.31	48.86	42.56	26.47	14.07	**35.31**	24.04	13.45	**30.39**
	5	**71.88**	**54.84**	**33.69**	**60.68**	**49.02**	**31.69**	**49.60**	42.21	**26.81**	**14.33**	35.12	**24.29**	**13.45**	30.19
	6	69.66	54.52	32.71	59.45	48.81	30.80	49.33	**42.62**	26.04	13.68	35.08	23.73	13.16	30.33
all	4	71.13	55.55	34.40	61.05	50.09	32.51	49.82	46.36	29.49	17.05	**38.53**	26.47	16.24	33.35
	5	**72.44**	55.76	**34.61**	**61.68**	50.19	**32.81**	**50.57**	46.19	**29.68**	**17.23**	38.18	**26.77**	**16.43**	33.31
	6	70.56	**55.84**	33.85	60.67	**50.31**	31.99	50.22	**46.49**	29.10	17.05	38.36	26.23	16.26	**33.40**

where $|\cdot|$ denotes absolute value operation, s_{iid} and s_{ood} represent the score of a model on iid and ood test sets, respectively. Smaller p_{gap} means the model is more robust.

On Charades-CD dataset, CTRL and ACRN achieve comparable performance gap between iid and ood test sets. Other methods, including VSLNet, show large performance gap between iid and ood test sets. Compared to those SOTAs, the gap of VSLNet is moderate, with $p_{gap} = 30.70\%$. With data debiasing, p_{gap} of VSLNet decreases from 30.70% to 13.84%. DD strategy significantly improves the results on ood test set by balancing the distribution of train set to be more uniform. MD strategy also reduces the p_{gap} distinctly by disentangling the bias from the TSGV model with two unimodal branches during training. On ActivityNet-CD dataset, the performance gap between iid and ood test sets are more conspicuous than that on Charades-CD dataset. Specifically, p_{gap} of VSLNet is 63.05%, the second largest among all models. By applying DD and MD strategies, the results on ood set increase and the results on iid set slightly decrease. After debiasing, p_{gap} of VSLNet reduces significantly.

6.3.5 Analysis of Data Debiasing Strategy

We now study the effect of number of clips N_{clip} in data debiasing (DD) strategy on Charades-CD dataset. We evaluate $N_{clip} \in \{4, 5, 6\}$ and report results in Table 6.5. Observe that performances of VSLNet+DD with different N_{clip}s are comparable on iid test set. However, VSLNet+DD with $N_{clip} = 5$ generally outperforms the model with other N_{clip} values on ood test set. Besides, VSLNet+DD with $N_{clip} = 5$ also performs best over all test samples. Similar observations hold on the ActivityNet-CD dataset.

TABLE 6.6: The performance (%) of VSLNet with different model debiasing (MD) strategies on the Charades-CD and the ActivityNet-CD datasets.

Split	Method	Charades-CD							ActivityNet-CD						
		R@1, IoU@μ			dR@1, IoU@μ			mIoU	R@1, IoU@μ			dR@1, IoU@μ			mIoU
		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7		μ=0.3	μ=0.5	μ=0.7	μ=0.3	μ=0.5	μ=0.7	
iid	VSLNet	**75.09**	60.51	**41.07**	**66.72**	56.12	**39.25**	**54.39**	**63.18**	**49.16**	**32.37**	**52.41**	**43.40**	**31.37**	**47.57**
	+V-MD	73.63	**60.87**	39.61	65.68	**56.41**	38.04	54.11	61.27	46.88	30.16	51.80	41.94	28.41	45.09
	+Q-MD	73.51	60.39	40.83	65.56	55.95	38.95	53.41	60.88	46.14	30.94	51.67	41.37	29.15	45.00
	+MD	73.39	59.17	39.13	65.22	54.69	37.41	53.26	58.73	45.10	29.32	50.23	40.51	27.60	43.62
ood	VSLNet	65.93	48.18	28.89	55.85	43.29	27.20	45.56	41.71	23.31	12.06	32.99	20.96	11.59	29.95
	+V-MD	68.24	52.74	32.47	58.21	47.50	30.53	48.42	42.85	27.23	14.56	36.07	24.91	13.96	30.45
	+Q-MD	67.11	51.59	32.18	57.90	46.63	30.33	47.60	**42.89**	27.06	14.60	**36.14**	24.73	14.01	**30.58**
	+MD	**70.79**	**54.90**	**33.30**	**60.59**	**49.26**	**31.64**	**49.49**	42.43	**27.70**	**15.01**	36.08	**25.33**	**14.41**	30.55
all	VSLNet	67.72	50.60	31.28	57.98	45.81	29.56	47.30	46.02	28.49	16.14	36.89	25.47	15.66	33.19
	+V-MD	69.92	54.34	33.87	59.67	49.25	32.00	49.53	**46.54**	31.18	17.12	39.23	28.13	16.75	33.35
	+Q-MD	68.37	53.31	33.87	59.40	48.45	32.02	48.74	46.50	30.88	17.28	**39.26**	27.86	16.85	**33.47**
	+MD	**70.25**	**55.79**	**35.30**	**60.93**	**50.63**	**33.33**	**50.40**	45.70	**31.19**	**17.88**	38.92	**28.38**	**17.05**	33.17

TABLE 6.7: Data statistics of Charades-CD and ActivityNet-CD. $\bar{L}_V$/ $\bar{L}_M$ is the average video/moment length in seconds, $\bar{L}_Q$ is average number of words in query, $\bar{N}_{A/V}$ is average annotations per video, N_{vocab} is the vocabulary size and N_{act} is the size action verb. Note we only count the verb with occurrence larger than 5 for N_{act}.

Dataset	$\bar{L}_V$	$\bar{L}_M$	$\bar{L}_Q$	$\bar{N}_{A/V}$	N_{vocab}	N_{act}
Charades-CD	30.75s	8.12s	6.22	2.42	1,255	69
ActivityNet-CD	117.60s	37.14s	14.41	4.82	13,707	901

6.3.6 Analysis of Model Debiasing Strategy

Here we study the effect of each unimodal model, *i.e.,* video-only and query-only branches on VSLNet. The results are summarized in Table 6.6. V-MD and Q-MD denotes debiasing using video-only or query-only branch only. Both V-MD and Q-MD strategies lead to significant improvement on the ood test set, and slight degradation on the iid test set. We observe the same on the ActivityNet-CD dataset. This set of results indicate that model debiasing on both video and query sides are necessary for learning a robust TSGV model.

6.3.7 Performance Differences on Charades-CD and ActivityNet-CD

The data and model debiasing strategies show their strong generalization capability on the test set with out-of-distribution. However, we observe that the performance of the two strategies on ActivityNet-CD are inferior to that on Charades-CD. Based on data statistics of the two datasets (see Table 6.7), ActivityNet-CD dataset is more challenging than Charades-CD, because average video and query lengths are much larger. Besides, ActivityNet-CD contains more than 900 different action verbs while Charades-CD has 69 verbs only. That is, the activities/events of

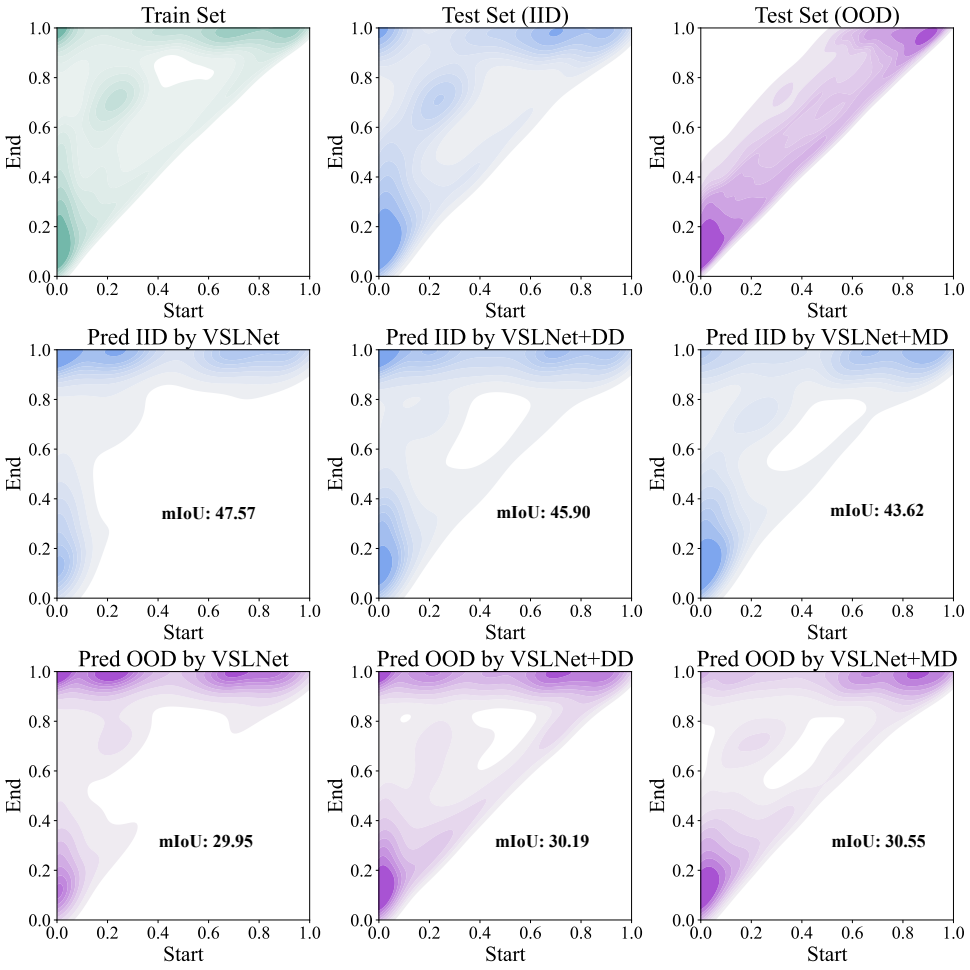


FIGURE 6.6: Visualization of moment annotation distributions of train, iid and ood test sets in ActivityNet-CD dataset, and that predicted by VSLNet and the proposed debiasing strategies.

ActivityNet-CD are many more diverse than that in Charades-CD. This could contribute to the difficulty of debiasing ActivityNet-CD. Figure 6.6 depicts the predicted annotation distributions of VSLNet and two proposed debiasing strategies on iid and ood test sets. Despite the improvements made by DD and MD, we observe that the predicted distributions of ood test set by VSLNet+DD/+MD remain similar to that of train set. In other words, the bias issue is more challenging to address on ActivityNet-CD dataset.

6.4 Summary

In this chapter, we have studied the annotation distribution bias issue existing in the commonly used benchmark datasets. Specifically, the bias denotes that the annotation distributions of train and test sets are independent-and-identical. In this sense, existing TSGV approaches tend to rely on exploiting the statistical regularities of annotation distribution for moment retrieval, to

achieve good performance. In other words, these methods could make a good guess of start/end time points, even without taking into consideration the input video and language query. To mitigate the suspicious correlations in the benchmark datasets, we propose two simple yet effective strategies, data debiasing and model debiasing. The data debiasing strategy is to balance the data sample distribution by oversampling with video truncation. While the model debiasing guides the TSGV model to learn accurate cross-modal interactions by explicitly eliminating the unimodal biases. With the help of two unimodal branches, we reduce the loss propagated to the TSGV model for biased samples and amplify the loss of non-biased samples. Thus, those non-biased samples are drawn more attention by the model during training. Through extensive experiments, we show that both data and model debiasing strategies contribute performance improvement on ood test sets.

Chapter 7

Video Corpus Moment Retrieval with Contrastive Learning

7.1 Introduction

TSGV aims to retrieve a short video segment from an untrimmed video that semantically corresponds to the given language query. In the previous chapters, we have explored to formulate TSGV as span-based question answering task and adopt the standard span-based QA framework to solve it. In this chapter[1], we study a new task, named video corpus moment retrieval (VCMR), which is an *extension* of TSGV task. Compared to TSGV which localizes relevant moments in a single video, VCMR is more challenging, since it aims to retrieve a temporal moment that semantically corresponds to a given text query from *a collection of untrimmed and unsegmented videos*, *i.e.,* a video corpus. Meanwhile, compared to TSGV, VCMR is more closer to the practical scenarios. Because acquiring a large collection of video-query pairs for TSGV is labor-intensive and time-consuming, and VCMR covers more applications like video surveillance, search, and navigation, within a video corpus. The comparison between TSGV and VCMR is depicted in Figure 7.1. VCMR was first extended from TSGV by Escorcia et al. [197], who devise a ranking-based clip-query alignment model. The model compares query features with uniformly partitioned video clips. Then several subsequent approaches [198, 199] are developed to solve this problem. *Note that TSGV also known as single video moment retrieval,* i.e., *SVMR. To better compare with the VCMR, we replace TSGV with SVMR in the following content.*

[1]This chapter is published as Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. "Video Corpus Moment Retrieval with Contrastive Learning". In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (Long Papers)*, pages: 685–695, Online, 2021 [38].

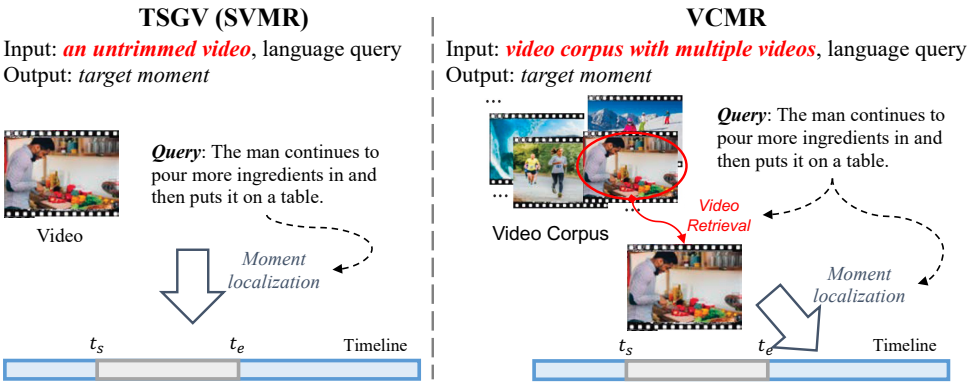


FIGURE 7.1: The comparison between TSGV (SVMR) and VCMR tasks.

Similar to SVMR, in order to perform query-based video moment retrieval, we need to learn the matching between query and video from training samples. In general, there are two approaches, illustrated in Figure 7.2. One is to encode video and text separately, and learn the matching through late feature fusion, known as ***unimodal encoding*** [197, 198]. With unimodal encoding, the text query is encoded to a d-dimensional feature vector. A video is encoded to a sequence of d-dimensional feature vectors, where each vector corresponds to a small fraction of the video, *e.g.,* a few frames. The other is ***cross-modal interaction learning***, which takes in a video as a sequence of visual features, and the query as a sequence of word features to learn their interactions [199]. The latter typically leads to better retrieval accuracy as the learned parameters capture the relevance between query and video at fine-grained granularity. However, in query evaluation, cross-modal encoding needs to be performed between query and *every video* in corpus (illustrated by "$\times N$" in Figure 7.2), leading to high computational cost. On the other hand, with unimodal encoding, visual features can be pre-encoded and stored. In query evaluation, we only need to encode query and then perform video retrieval and moment localization. The challenge becomes to refine two separate encoders during training process, such that the encoded features are well aligned for accurate retrieval. Thus, there is a contradiction between high efficiency and high-quality retrieval for existing two approaches. That is, unimodal encoding approaches are with high efficiency but low retrieval accuracy, while cross-modal encoding approaches are with high retrieval accuracy but low efficiency.

Compared to SVMR which focuses on moment localization, VCMR contains two objectives, *i.e.,* video retrieval and moment localization (see Figure 7.2). Thus, given a language query, VCMR requires to first retrieve the best matched video from video corpus, then perform moment localization in the retrieved video. Besides, for both approaches, video retrieval and moment localization are performed jointly, *i.e.,* the model is trained with a joint objective. An earlier study [198] has shown that joint learning outperforms two-stage learning where video retrieval and moment localization are treated as two separate subtasks and performed in stages.

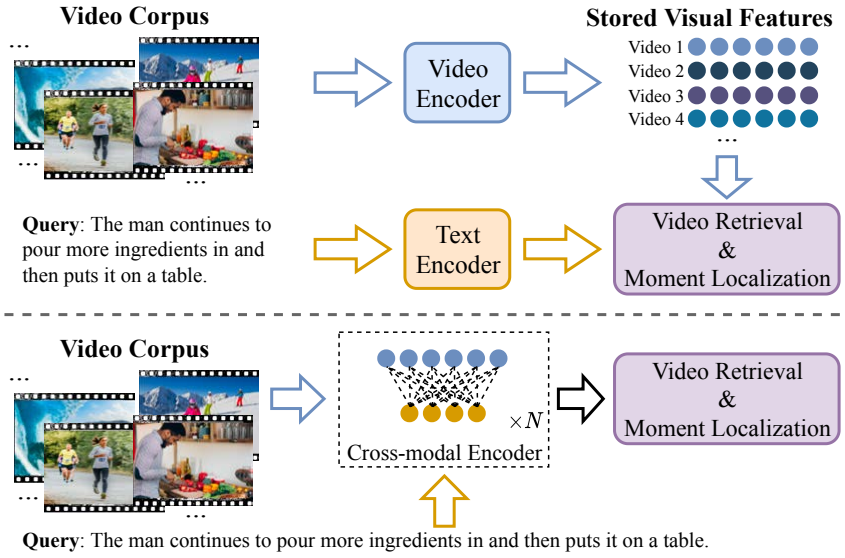


FIGURE 7.2: Two approaches to VCMR: unimodal encoding vs. cross-modal interaction learning.

We aim to remedy the contradiction between high efficiency and high-quality retrieval in VCMR, and achieve the advantages of both unimodal and cross-modal encoding approaches. The essence of cross-modal interaction is to highlight the relevant and important information from both modalities via co-attention mechanisms. Meanwhile, contrastive learning [200–202] is a strategy to maximize the mutual information (MI) [203, 204] of positive pairs and to minimize the MI of negative pairs. In our context, a pair of matching video and query is a positive pair and a non-matching pair is a negative pair in training. We consider that both cross-modal interaction learning and contrastive learning share the same objective of emphasizing the relevant information of input pairs. Hence, we can apply contrastive learning to refine encoders in unimodal encoding to achieve similar effectiveness.

Our Approach. We develop a Retrieval and Localization Network (ReLoNet) as a base network to separately encode video and query representations, *i.e.,* in an unimodal encoding style, and to (late) fuse them for joint retrieval. We then introduce contrastive learning to ReLoNet to simulate cross-modal interactions between video and query, and propose ReLoCLNet. Build on top of ReLoNet, ReLoCLNet is trained with two contrastive learning objectives: VideoCL and FrameCL. The VideoCL objective aims to learn video and text features such that the semantically related videos and queries are close to each other, and far away otherwise. The FrameCL works at frame-level for moment localization, which simulates fine-grained cross-modal interactions between visual and textual features within a video. In FrameCL, we regard the features within target moment as foreground (positive samples), while the remaining as background (negative samples). Thus, FrameCL enhances the MI between query and foreground, while

suppresses the MI between query and background. Once trained, the learned parameters in video encoder and text encoder can be used to encode video and text features separately and independently. Accordingly, all videos in a given corpus can be pre-encoded by the learned video encoder and stored, as illustrated in Figure 7.2, for efficient retrieval. Experiments on two benchmarks to demonstrate that ReLoCLNet achieves comparable accuracy with cross-modal interaction learning, with much faster retrieval speed.

7.2 ReLoCLNet Framework

To ensure retrieval efficiency, we follow the unimodal encoding approach and aim to develop video encoder and text encoder for effective feature encoding separately. To achieve high-quality retrieval results, we aim to simulate the cross-modal interaction learning to better align the encoded video and text features. To this end, we introduce contrastive learning to our model. Conceptually, contrastive learning and cross-modal interaction learning share a similar objective of highlighting the relevant information of input pairs, *i.e.,* matching video-query pairs in our setting. Different from cross-modal interaction learning, contrastive learning is only engaged in the training phase. Once trained, the learned parameters ensure the alignment between the encoded video features and text features even though the two features are encoded separately. The task objective, *i.e.,* video retrieval and moment localization, can then be easily achieved through late feature fusion. In this section, we first develop the ReLoNet as a base model, to separately encode video and query inputs and fuse them for prediction. Then we design two contrastive learning objectives: (i) Video-level Contrastive Learning (VideoCL) for video retrieval, and (ii) Frame-level Contrastive Learning (FrameCL) for moment localization. During training phrase, VideoCL and FrameCL simulate the cross-modal interaction to enhance the representation learning. During inference (*i.e.,* retrieval) phrase, the model separately encodes video and query to maintain retrieval efficiency. The overall architecture of the proposed model is shown in Figure 7.3. Next, we formally formulate the research problem, then detail the components in ReLoNet and ReLoCLNet.

7.2.1 Problem Formulation

We denote a video corpus as $\mathcal{V} = \{V^1, V^2, \ldots, V^M\}$, where M is the number of videos and $V^k = [f_i]_{i=0}^{T-1}$ represents the k-th video with T frames.[2] Given a text query $Q = [q_i]_{i=0}^{m-1}$, we aim to retrieve the temporal moment (starting from τ_s and ending at τ_e) in V^* that semantically

[2]Videos in $\mathcal{V}$ could be of different lengths; we simply use T to represent the length (in number of frames) of an arbitrary video.

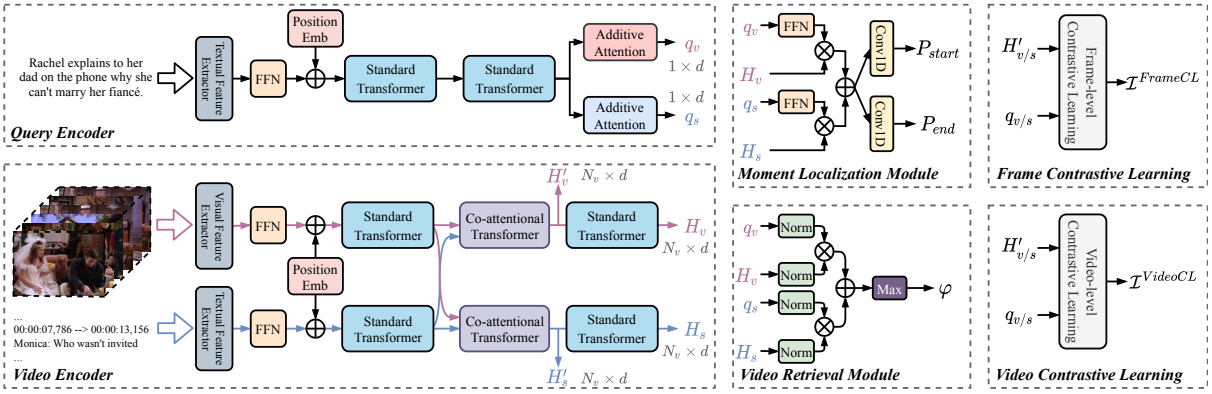


FIGURE 7.3: In both ReLoNet and ReLoCLNet, query is encoded to $\boldsymbol{q}_{v/s}$ and video is encoded to $\boldsymbol{H}_{v/s}$, for video retrieval and moment localization. ReLoCLNet adds contrastive learning objectives through $\boldsymbol{q}_{v/s}$ and $\boldsymbol{H}'_{v/s}$ to refine query and video encoders.

corresponds to Q from video corpus $\mathcal{V}$. Here V^* denotes a video that contains the ground truth moment and $\tau_{s/e}$ are the start/end time points of target moment in V^*. Thus, VCMR has two objectives: (i) video retrieval, *i.e.,* finding V^* from $\mathcal{V}$; and (ii) moment localization, *i.e.,* locating the target moment in V^*.

For words in Q, the initial encoding is obtained from pre-trained word embeddings or language models as $\boldsymbol{Q} = [\boldsymbol{q}_i]_{i=0}^{m-1} \in \mathbb{R}^{n \times d_q}$, where d_q is the word feature dimension. For each video $V \in \mathcal{V}$, we split it into n clip units, and use pre-trained feature extractor to encode them into visual features $\boldsymbol{V} = [\boldsymbol{v}_i]_{i=0}^{n-1} \in \mathbb{R}^{n \times d_v}$, where d_v is the visual feature dimension. Then, $\tau_{s(e)}$ are mapped to the corresponding indices $a_{s(e)}$ in the visual feature sequence, and the target moment is represented as $\boldsymbol{m}^* = \{\boldsymbol{v}_i | i = a_s, \cdots, a_e\}$, where $0 \leq a_s \leq a_e \leq n-1$. That is, in term of visual feature space, $\boldsymbol{m}^*$ may correspond to a sequence of $\boldsymbol{v}_i$'s of any length within n starting from any index. The best matching $\boldsymbol{m}^*$ can be estimated by:

$$\boldsymbol{m}^* = \arg \max_{\boldsymbol{m} \in \boldsymbol{V}, \boldsymbol{V} \in \mathcal{V}} p(\boldsymbol{m}|\boldsymbol{V}, \boldsymbol{Q}) p(\boldsymbol{V}|\boldsymbol{Q}). \tag{7.1}$$

Given M videos in $\mathcal{V}$ with average video feature length n, the search space is $\mathcal{O}(M \times n_v^2)$. It is infeasible to compute $\boldsymbol{m}^*$ in such a large space. Hence, we approximate Equation (7.1) by:

$$\boldsymbol{V}^* = \arg \max_{\boldsymbol{V}} p(\boldsymbol{V}|\boldsymbol{Q}) \quad \text{and} \quad \boldsymbol{m}^* \approx \arg \max_{\boldsymbol{m} \in \boldsymbol{V}^*} p(\boldsymbol{m}|\boldsymbol{V}^*, \boldsymbol{Q}). \tag{7.2}$$

Equation (7.2) is consistent with two objectives of VCMR, *e.g.,* video retrieval and moment localization. The search space reduces to $\mathcal{O}(M + M' \times n^2)$, where M' is the top-M' retrieved videos ($M' \ll M$) from the video corpus. In addition to visual features, a video may contain its own multi-modality features, such as subtitle and audio. For instance, videos in TVR

dataset [198] come with subtitles. We denote the subtitle of a video by S, and the features extracted from subtitle by $\boldsymbol{S} \in \mathbb{R}^{d_w \times n_v}$. For easy presentation, we assume all videos come with subtitles and simply use "video" to refer "video + subtitle".

7.2.2 Query Encoder

The structure of query encoder is shown in Figure 7.3. Given a text query Q with m words, we first apply textual feature extractor to covert words in the query to corresponding features $\boldsymbol{Q} = [\boldsymbol{q}_i]_{i=0}^{m-1} \in \mathbb{R}^{m \times d_q}$. Then we project the obtained features into dimension d with a feed-forward layer as $\hat{\boldsymbol{Q}} = \boldsymbol{Q} \cdot \boldsymbol{W}_q + \boldsymbol{b}_q \in \mathbb{R}^{m \times d}$, where $\boldsymbol{W}_q \in \mathbb{R}^{d_q \times d}$ and $\boldsymbol{b}_q \in \mathbb{R}^d$ are the learnable weight and bias, respectively.

Positional embedding is incorporated to each feature of the query sequence $\hat{\boldsymbol{Q}}$ before they are fed to the transformer blocks [177]. We adopt the transformer block to better capture the contextual representations of the query, for its proven effectiveness [34, 91, 92]. Specifically, the transformer block consists of a multi-head attention layer and a feed-forward layer. Residual connection [179] and layer normalization [178] strategies are applied to each layer in the transformer block. The encoded contextual representations of the query after the transformer block become $\widetilde{\boldsymbol{Q}}$:

$$\widetilde{\boldsymbol{Q}} = \texttt{Transformer}_q(\hat{\boldsymbol{Q}}). \tag{7.3}$$

We use *two* transformer blocks in the query encoder. Then we apply additive attention mechanism [13] to compute the attention scores of each query word. The scores computed are utilized to aggregate the information of $\widetilde{\boldsymbol{Q}} = [\tilde{\boldsymbol{q}}_0, \tilde{\boldsymbol{q}}_2, \ldots, \tilde{\boldsymbol{q}}_{m-1}]$ to compute the modularized query vector, *i.e.,* the sentence representation of $\widetilde{\boldsymbol{Q}}$:

$$\boldsymbol{\alpha}^q = \texttt{Softmax}\big(\widetilde{\boldsymbol{Q}} \cdot \boldsymbol{W}_{m,\alpha}\big) \in \mathbb{R}^{n_q}, \quad \boldsymbol{q}_m = \sum_{i=0}^{m-1} \alpha_i^q \times \boldsymbol{q}_i \in \mathbb{R}^d, \tag{7.4}$$

where $\boldsymbol{q}_m \in \mathbb{R}^d$ denotes the modularized query vector. $m \in \{v, s\}$ means two modularized query vectors, $\boldsymbol{q}_v$ and $\boldsymbol{q}_s$, are computed for matching with visual and subtitle features, respectively. Both $\boldsymbol{q}_v$ and $\boldsymbol{q}_s$ are d-dimensional vectors as shown in Figure 7.3. If the videos to be retrieved do not contain subtitles, then only $\boldsymbol{q}_v$ is computed.

7.2.3 Video Encoder

We detail the video encoder with the assumption that the videos come with subtitles, as shown in Figure 7.3. Given a video with its subtitle, we first use visual and textual feature extractors to obtain the corresponding visual and subtitle features $\boldsymbol{V} \in \mathbb{R}^{n \times d_v}$ and $\boldsymbol{S} \in \mathbb{R}^{n \times d_q}$, respectively.

Then both $\boldsymbol{V}$ and $\boldsymbol{S}$ are projected into dimension d with two feed-forward layers as $\hat{\boldsymbol{V}} = \boldsymbol{V} \cdot \boldsymbol{W}_v + \boldsymbol{b}_v \in \mathbb{R}^{n \times d}$ and $\hat{\boldsymbol{S}} = \boldsymbol{S} \cdot \boldsymbol{W}_s + \boldsymbol{b}_s \in \mathbb{R}^{n \times d}$, where $\boldsymbol{W}_v \in \mathbb{R}^{d_v \times d}$ and $\boldsymbol{b}_v \in \mathbb{R}^d$ are the weight and bias for video feed-forward layer; $\boldsymbol{W}_s \in \mathbb{R}^{d_q \times d}$ and $\boldsymbol{b}_s \in \mathbb{R}^d$ are the weight and bias for subtitle feed-forward layer.

Similar to the query encoder, we add positional embeddings to both $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{S}}$, and feed them to the transformer block. The encoded contextual representations for video and subtitle are:

$$\widetilde{\boldsymbol{V}} = \texttt{Transformer}_v(\hat{\boldsymbol{V}}), \text{ and } \widetilde{\boldsymbol{S}} = \texttt{Transformer}_s(\hat{\boldsymbol{S}}), \tag{7.5}$$

where $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{n \times d}$ and $\widetilde{\boldsymbol{S}} \in \mathbb{R}^{n \times d}$.

Different from the query encoder, we do not use two transformer blocks here. Instead, after the first transformer blocks, we use co-attentional transformer blocks [19, 22, 23, 198]. Because the visual content in a video and its subtitle are well aligned, through co-attentional transformers, we are able to better capture the cross-modal representations of video and subtitle within a video. Given $\widetilde{\boldsymbol{V}}$ and $\widetilde{\boldsymbol{S}}$, the cross-modal representations are encoded as:

$$\boldsymbol{H}'_v = \texttt{Co-Transformer}_{vs}(\widetilde{\boldsymbol{V}}, \ \widetilde{\boldsymbol{S}}), \text{ and } \boldsymbol{H}'_s = \texttt{Co-Transformer}_{sv}(\widetilde{\boldsymbol{S}}, \ \widetilde{\boldsymbol{V}}), \tag{7.6}$$

where $\boldsymbol{H}'_v \in \mathbb{R}^{n \times d}$ and $\boldsymbol{H}'_s \in \mathbb{R}^{n \times d}$ are the learned cross-modal representations of video and subtitle, respectively. Finally, we refine the encoded cross-modal representations of $\boldsymbol{H}'_v$ and $\boldsymbol{H}'_s$ with standard transformer blocks by learning the self-attentive contexts, respectively. The final output is calculated as:

$$\boldsymbol{H}_v = \texttt{Transformer}_v(\boldsymbol{H}'_v), \text{ and } \boldsymbol{H}_s = \texttt{Transformer}_s(\boldsymbol{H}'_s), \tag{7.7}$$

where $\boldsymbol{H}_v \in \mathbb{R}^{n \times d}$ and $\boldsymbol{H}_s \in \mathbb{R}^{n \times d}$ are the final output representations of video and subtitle, respectively.

If videos do not come with subtitles, then the feature encoding pipeline for subtitle will be removed. Accordingly, the co-attentional transformer becomes the standard transformer, and the final output is $\boldsymbol{H}_v$ only.

7.2.4 Video Retrieval Module

Through query encoding, a query is encoded to two d-dimensional vectors $\boldsymbol{q}_m \in \mathbb{R}^d$, $m \in \{v, s\}$, for matching with visual and subtitle features from a video. Recall that, with video encoding, each video is encoded to $\boldsymbol{H}_m = [\boldsymbol{h}_m^0, \boldsymbol{h}_m^2, \ldots, \boldsymbol{h}_m^{n-1}] \in \mathbb{R}^{n \times d}$, *i.e.,* a sequence of $\boldsymbol{h}_m$'s each represents two d-dimensional vectors for visual and subtitle features extracted from

a small fraction of a video. We estimate the matching between the query and a video by cosine similarities computed on $\boldsymbol{q}_m$ and $\boldsymbol{H}_m$, *i.e.,* a simple late feature fusion. Specifically, we compute the cosine similarities between $\boldsymbol{q}_m$ and each element of $\boldsymbol{H}_m$ as:

$$\boldsymbol{\varphi}_m = \texttt{norm}(\boldsymbol{H}_m^\top) \cdot \texttt{norm}(\boldsymbol{q}_m), \tag{7.8}$$

where $m \in \{v, s\}$, $\boldsymbol{\varphi}_m \in \mathbb{R}^n$, and `norm` denotes the $l2$ normalization operation. Then we select the maximum score in $\boldsymbol{\varphi}_m$ to represent the matching between query and video:

$$\varphi_m = \max(\boldsymbol{\varphi}_m) = \max([\varphi_m^0, \varphi_m^1, \ldots, \varphi_m^{n_v-1}]), \tag{7.9}$$

where φ_m is a scalar. If videos come with subtitles, then $\varphi = \frac{1}{2}(\varphi_v + \varphi_s)$, otherwise $\varphi = \varphi_v$.

We adopt the hinge loss as training objective for video retrieval, similar to [198, 205–207]. We first sample two sets of negative pairs $\{(Q_i^-, V)\}_{i=1}^N$ and $\{(Q, V_i^-)\}_{i=1}^N$ for each positive pair (Q, V), where Q^- and V^- denote the negative (*i.e.,* non-matching) query and video, respectively.[3] Suppose the computed similarity scores of both sets of negative pairs are $\boldsymbol{\varphi'}$ and $\boldsymbol{\varphi''}$, the hinge loss is calculated as:

$$\mathcal{L}^{VR} = \max(0, \Delta + \frac{1}{N}\sum \boldsymbol{\varphi'} - \varphi) + \max(0, \Delta + \frac{1}{N}\sum \boldsymbol{\varphi''} - \varphi), \tag{7.10}$$

where Δ is the pre-defined margin value and we set $\Delta = 0.1$.

7.2.5 Moment Localization Module

For efficiency purpose, moment localization is also computed based on the encoded query features $\boldsymbol{q}_m$ and video features $\boldsymbol{H}_m$, through late feature fusion, following [184, 198]. Specifically, $\boldsymbol{q}_m$ is further encoded with a feed-forward layer as $\boldsymbol{q}'_m = \boldsymbol{W}_m \cdot \boldsymbol{q}_m + \boldsymbol{b}_m \in \mathbb{R}^d$. Then we compute video-query similarity scores as:

$$\boldsymbol{\mathcal{S}}_{mq} = \boldsymbol{H}_m^\top \cdot \boldsymbol{q}'_m \in \mathbb{R}^{n_v}, \quad \text{where} \quad m \in \{v, s\}. \tag{7.11}$$

Again, if subtitle is available, $\boldsymbol{\mathcal{S}} = \frac{1}{2}(\boldsymbol{\mathcal{S}}_{vq} + \boldsymbol{\mathcal{S}}_{sq})$, otherwise $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}}_{vq}$. The start and end scores for target moment are generated by convolutional start-end boundary predictor [198]:

$$\boldsymbol{\mathcal{S}}_{\text{start}} = \texttt{Conv1D}_{\text{start}}(\boldsymbol{\mathcal{S}}), \text{ and } \boldsymbol{\mathcal{S}}_{\text{end}} = \texttt{Conv1D}_{\text{end}}(\boldsymbol{\mathcal{S}}), \tag{7.12}$$

[3] We simply use V to represent a video with its subtitle if available.

where $\boldsymbol{\mathcal{S}}_{\text{start/end}} \in \mathbb{R}^{n_v}$. Then, the probability distributions of start and end boundaries are computed by:

$$\boldsymbol{P}_{\text{start}} = \texttt{Softmax}(\boldsymbol{\mathcal{S}}_{\text{start}}), \text{ and } \boldsymbol{P}_{\text{end}} = \texttt{Softmax}(\boldsymbol{\mathcal{S}}_{\text{end}}). \tag{7.13}$$

For a video-query pair, the predicted start and end boundaries of the target moment are derived by maximizing the joint probability:

$$\begin{aligned} (\hat{a}_s, \hat{a}_e) &= \arg\max_{a_s, a_e} \boldsymbol{P}_{\text{start}}(a_s) \times \boldsymbol{P}_{\text{end}}(a_e), \\ P^{se} &= \boldsymbol{P}_{\text{start}}(\hat{a}_s) \times \boldsymbol{P}_{\text{end}}(\hat{a}_e), \end{aligned} \tag{7.14}$$

where $0 \leq \hat{a}_s \leq \hat{a}_e \leq n-1$, and P^{se} is the score of best boundaries $(\hat{a}_s, \hat{a}_e)$. The training objective of moment localization is:

$$\mathcal{L}^{ML} = \frac{1}{2} \times \Big(f_{\text{XE}}(\boldsymbol{P}_{\text{start}}, \boldsymbol{Y}_{\text{start}}) + f_{\text{XE}}(\boldsymbol{P}_{\text{end}}, \boldsymbol{Y}_{\text{end}}) \Big), \tag{7.15}$$

where f_{XE} is the cross-entropy function, $\boldsymbol{Y}_{\text{start}}$ and $\boldsymbol{Y}_{\text{end}}$ are one-hot labels for start (a_s) and end (a_e) boundaries of the ground truth moment, respectively.

We now have the full picture of the base architecture ReLoNet with four modules: query encoder (Section 7.2.2), video encoder (Section 7.2.3), video retrieval and moment localization modules (Sections 7.2.4 and 7.2.5). Next, we incorporate contrastive learning objectives into ReLoNet to develop ReLoCLNet.

7.2.6 Video and Frame Contrastive Learning

In ReLoNet, video retrieval and moment localization are fully based on the encoded query features $\boldsymbol{q}_m$ and video features $\boldsymbol{H}_m$. They are both computed by simple late future fusion. Quality of the final moment retrieval hence heavily relies on the effectiveness of the two separate encoders, query encoder and video encoder. In ReLoCLNet, we aim to guide the two encoders to simulate cross-modal interaction learning in the training phase. To this end, we introduce two contrastive learning objectives, VideoCL and FrameCL. VideoCL guides the two encoders to better distinguish matching video-query pairs from non-matching pairs. FrameCL guides the two encoders to better distinguish the matching moment to the query from the non-matching moments.

Video Contrastive Learning (VideoCL). VideoCL guides the encoders to learn a joint feature space where the semantically related videos and queries are close to each other, and far away otherwise. In other words, VideoCL aims to reduce the distance of matching video-query pairs, and to increase the distance of non-matching pairs, in the joint feature space.

We encode the latent representation of video $\boldsymbol{H}'_m \in \mathbb{R}^{n\times d}$ from Equation (7.6) (illustrated as $\boldsymbol{H}'_v$ and $\boldsymbol{H}'_s$ in Figure 7.3) into its modularized video representation $\boldsymbol{c}_m$. Similar to modular component in query encoder, we adopt additive attention mechanism to compute $\boldsymbol{c}_m$:

$$\boldsymbol{\alpha}^m = \texttt{Softmax}\big(\boldsymbol{H}'_m \cdot \boldsymbol{W}_{m,\alpha}\big) \in \mathbb{R}^{n_v}, \quad \boldsymbol{c}_m = \sum_{i=0}^{n_v-1} \alpha_i^m \times \boldsymbol{h}'_{m,i}, \tag{7.16}$$

where $\boldsymbol{c}_m \in \mathbb{R}^d$, $\boldsymbol{W}_{m,\alpha} \in \mathbb{R}^{d\times 1}$ and $m \in \{v, s\}$.

Given a set of positive (*i.e.,* matching) video-query pairs $\mathcal{P} = \{(\boldsymbol{c}_m, \boldsymbol{q}_m)\}$ and the sampled set of negative (*i.e.,* non-matching) video-query pairs $\mathcal{N} = \{(\boldsymbol{c}'_m, \boldsymbol{q}'_m)\}$, we adopt the noise-contrastive estimation (NCE) [20, 208–210] to compute the VideoCL score:

$$\mathcal{I}_m^e = \log\left(\frac{\sum\limits_{(\boldsymbol{c}_m,\boldsymbol{q}_m)\in\mathcal{P}} e^{f(\boldsymbol{c}_m)^\top \cdot g(\boldsymbol{q}_m)}}{\sum\limits_{(\boldsymbol{c}_m,\boldsymbol{q}_m)\in\mathcal{P}} e^{f(\boldsymbol{c}_m)^\top \cdot g(\boldsymbol{q}_m)} + \sum\limits_{(\boldsymbol{c}'_m,\boldsymbol{q}'_m)\sim\mathcal{N}} e^{f(\boldsymbol{c}'_m)^\top \cdot g(\boldsymbol{q}'_m)}}\right), \tag{7.17}$$

where the exponential term, $e^{f(\boldsymbol{c})^\top \cdot g(\boldsymbol{q})}$, computes the mutual information (MI) between $\boldsymbol{c}$ and $\boldsymbol{q}$. $f(\cdot)$ and $g(\cdot)$ denote the parametrized mappings, which project video and query representations into the same embedding space. Again, $\mathcal{I}^e = \frac{1}{2}(\mathcal{I}_v^e + \mathcal{I}_s^e)$ if subtitle is available, otherwise $\mathcal{I}^e = \mathcal{I}_v^e$. The objective of NCE is to optimize $\max_{f,g}(\mathcal{I}^e)$, which is equivalent to maximizing the ratio of the summed MI's of all samples in $\mathcal{P}$ and the summed MI's of all samples in $\mathcal{N}$ [20]. The loss of VideoCL is defined as:

$$\mathcal{L}^{VideoCL} = -\mathcal{I}^e. \tag{7.18}$$

Frame Contrastive Learning (FrameCL). FrameCL focuses on moment localization within a given pair of video-query, where the video retrieval module predicts the video contains a matching moment to the query. We regard the video features that reside within boundaries of the target moment as foreground or positive samples, and the rest as background or negative samples. Then we compute the contrastive loss by measuring MI between the query and the positive/negative video features. For this purpose, we utilize a discriminative approach based on mutual information maximization [211, 212].

The structure of FrameCL module is shown in Figure 7.4. The inputs $\boldsymbol{H}'_v$, $\boldsymbol{H}'_s$, $\boldsymbol{q}_v$, and $\boldsymbol{q}_s$ are outputs illustrated in Figure 7.3. Given the latent representation of video $\boldsymbol{H}'_m \in \mathbb{R}^{n\times d}$, we first split it into two parts by boundaries of target moment. The positive/foreground video features are $\boldsymbol{H}'_{m,F} = \{\boldsymbol{h}'_{m,i} | i = i^s, \ldots, i^e\} \in \mathbb{R}^{n_t\times d}$, which are features within the target moment.[4] The

[4] $n_t = i^e - i^s + 1$, and it denotes the length of target moment.

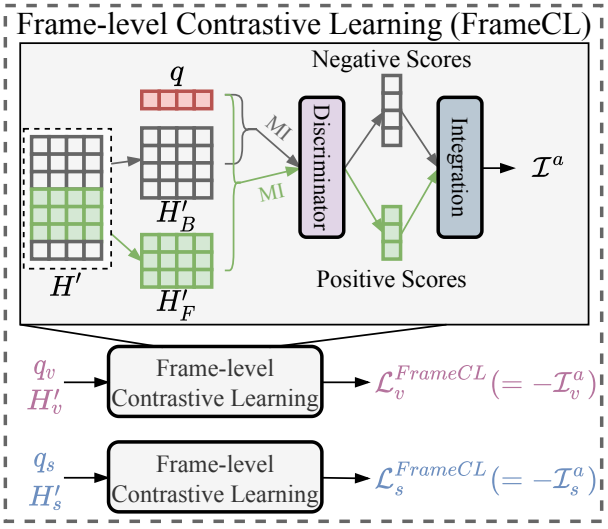


FIGURE 7.4: Structure of the FrameCL module.

negative/background features $\boldsymbol{H}'_{m,B} = \{\boldsymbol{h}'_{m,i} | i = 0, \ldots, i^s - 1, i^e + 1, \ldots, n_v - 1\} \in \mathbb{R}^{(n-n_t)\times d}$, are not in the target moment.

With query representation $\boldsymbol{q}_m$, foreground representation $\boldsymbol{H}'_{m,F}$, and background representation $\boldsymbol{H}'_{m,B}$, our goals are to maximize the MI between the query and the foreground, as well as to minimize the MI between the query and the background. Since MI estimation is in general intractable for continuous and random variables, we choose to maximize the value over lower bound estimators of MI, through Jensen-Shannon MI estimator [211] as:

$$\mathcal{I}_m^a = \mathbb{E}_{\boldsymbol{H}'_{m,F}}\Big[-\mathtt{sp}\big(-\mathcal{C}_\theta(\boldsymbol{q}, \boldsymbol{H}'_{m,F})\big)\Big] - \mathbb{E}_{\boldsymbol{H}'_{m,B}}\Big[\mathtt{sp}\big(\mathcal{C}_\theta(\boldsymbol{q}, \boldsymbol{H}'_{m,B})\big)\Big], \tag{7.19}$$

where $\mathtt{sp}(x) = \log(1 + ex)$ is the Softplus activation. $\mathcal{C}_\theta : d \times d \to \mathbb{R}$ refers to a discriminator. Similarly, $\mathcal{I}^a = \frac{1}{2}(\mathcal{I}_v^a + \mathcal{I}_s^a)$ if subtitle is available, otherwise $\mathcal{I}^a = \mathcal{I}_v^a$. The contrastive loss of FrameCL is:

$$\mathcal{L}^{FrameCL} = -\mathcal{I}^a. \tag{7.20}$$

Note that, both VideoCL and FrameCL are training objectives, and their losses are used to update video and query encoders. Although the two objectives are designed for video retrieval and moment localization respectively, they mutually affect each other, because both video and query encoders are adjusted based on the loss from both VideoCL and FrameCL, together with other losses.

7.2.7 Training and Inference

The overall training loss for ReLoCLNet is:

$$\mathcal{L} = \lambda_1 \times \mathcal{L}^{VR} + \lambda_2 \times \mathcal{L}^{ML} + \lambda_3 \times \mathcal{L}^{VideoCL} + \lambda_4 \times \mathcal{L}^{FrameCL}, \tag{7.21}$$

where λ_i's are hyperparameters to balance the contribution of each loss. We set $\lambda_1 = 1.0$ and $\lambda_{2,3,4} = 0.01$ to keep all losses at the same order of magnitude *i.e.,* equal contributions from the four components. Note that each video contains a large number of candidate moments.

During inference for VCMR, given a text query and a video corpus with M videos, we first use Equation (7.8) and Equation (7.9) to compute the similarity between the query and each of the M videos, leading to $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_M]$. The top-$K$ most relevant videos are retrieved based on $\boldsymbol{\varphi}$ ($K = 100$ in our implementation). For each retrieved video, we compute the scores of a few candidate predicted moments by Equation (7.14). Let P^{se} be the score of one predicted moment in the video. The final VCMR score is computed by:

$$\delta = P^{se} \times e^{\gamma \cdot \varphi}. \tag{7.22}$$

The exponential term and the hyperparameter γ are used to balance the importance of video retrieval and moment localization scores.

7.3 Experiments

7.3.1 Experimental Settings

Datasets. We conduct experiments on two benchmark datasets: ActivityNet Captions [165] and TVR [198]. ActivityNet Captions contains around 20K videos taken from the ActivityNet [166] dataset. The average video duration is about 120 seconds, the average query length is around 14.78 words, the average moment duration is about 36.18 seconds, and each video contains 3.68 annotations on average. This dataset is originally designed for SVMR task, then adapted to VCMR by Escorcia et al. [197]. We follow the setup in [197, 199] with $10,009$ and $4,917$ videos (*i.e.,* $37,421$ and $17,505$ annotations) for train and test, respectively. **TVR** is collected by Lei et al. [198], which contains 21.8K videos and 109K queries in total. The average video duration is 76.2 seconds, the query contains 13.4 words on average, the average moment duration is 9.1 seconds, and each video contains 5 annotations on average. We follow Zhang et al. [199] with $17,435$ and $2,179$ videos for train and test, respectively. Same as Lei et al. [198]

TABLE 7.1: The hyper-parameters for TVR and ActivityNet Captions.

Hyperparameter Name	TVR	ANetCaps
n_v (max video sequence)	128	
n_q (max query sequence)	30	64
d_v (visual feature dim)	3072 2048(ResNet)+1024(I3D)	1024 (I3D)
d_w (word feature dim)	768 (RoBERTa)	300 (GloVe)
d (hidden size)	384	
γ	30	20
# negative samples in VR: 10		Optimizer: AdamW [10]
Dropout rate: 0.1	Weight decay rate: 0.01	Batch size: 128
Learning rate (lr): 0.0001		lr warmup proportion: 0.01
Early stop tolerance: 10		# total training epochs: 100

and Zhang et al. [199], we utilize both video and subtitle features in the TVR dataset for train and test.

Evaluation Metrics. We evaluate the models for the VCMR task as well as its two subtasks: video retrieval (VR) and SVMR. For VR, we use "**Recall**@k" ($k \in \{1, 5, 10, 100\}$) as the evaluation metric following [198, 199]. Note that we do not use "Precision@k" because each query only corresponds to one ground truth video, in both datasets. For SVMR and VCMR, we use "**Recall**@k, **IoU**=μ" as the evaluation metric, which denotes the percentage of test samples that have at least one predicted moment whose *intersection over union* (IoU) with the ground-truth moment is larger than μ in the top-k predictions. We set $k \in \{1, 10, 100\}$ and $\mu \in \{0.5, 0.7\}$. A prediction is correct if (i) the predicted video matches the ground truth video, and (ii) the predicted moment has high overlap with the ground truth moment, where temporal IoU is used to measure the overlap [198].

Implementation. For the ActivityNet Captions, we use I3D [25] pre-trained on the Kinetics dataset [213] as the visual feature extractor following Zhang et al. [199], and adopt GloVe embeddings [26] as the textual feature extractor for query words. For TVR, we directly use the visual and textual features provided by Lei et al. [198]. The visual feature is the concatenation of appearance feature extracted by ResNet152 [179] pre-trained on ImageNet [214] and temporal feature extracted by I3D. The textual feature of query and subtitle is extracted by 12-layer pre-trained RoBERTa [41]. The negative sets of video retrieval and VideoCL modules are sampled within each mini-batch during training. The hyperparameters are summarized in Table 7.1. Other hyperparameters are given when describing the corresponding model components.[5]

[5]Our model is implemented in PyTorch `1.7.0` with CUDA `11.1` and cudnn `8.0.5`. All experiments are conducted on a workstation with dual NVIDIA GeForce RTX 3090 GPUs.

TABLE 7.2: Results of VCMR on TVR and ActivityNet Captions datasets.

Dataset	Method	Recall@k, IoU $= 0.5$			Recall@k, IoU $= 0.7$		
		R1	R10	R100	R1	R10	R100
TVR	XML [198]	-	-	-	2.62	9.05	22.47
	HERO [215]	-	-	-	2.98	10.65	18.25
	FLAT [199]	*8.45*	21.14	30.75	*4.61*	11.29	16.24
	HAMMER [199]	**9.19**	*21.28*	31.25	**5.13**	*11.38*	16.71
	ReLoNet	5.46	16.65	*35.08*	2.71	9.37	*22.87*
	ReLoCLNet	8.03	**21.37**	**44.10**	4.15	**14.06**	**32.42**
ActivityNet	MCN [46]	0.02	0.18	1.26	0.01	0.09	0.70
	CAL [197]	0.21	1.32	6.82	0.12	0.89	4.79
	FLAT [199]	2.57	*13.07*	*30.66*	1.51	*7.69*	17.67
	HAMMER [199]	*2.94*	**14.49**	**32.49**	*1.74*	**8.75**	**19.08**
	ReLoNet	2.16	9.96	24.54	1.26	5.64	17.43
	ReLoCLNet	**3.09**	11.28	25.95	**1.82**	6.91	*18.33*

7.3.2 Overall Performance

We compare our models with MCN [46], CAL [197], XML [198], HERO [215], FLAT [199] and HAMMER [199]. Among them, MCN, CAL, XML, and HERO follow unimodal encoding approaches, while FLAT and HAMMER belong to cross-modal interaction learning approaches (see Figure 7.2). FLAT is a variant of HAMMER without using hierarchical structure. In all tables, results of the compared models are reported in their corresponding papers.[6] The best results are in **boldface** and the second bests are in *italic*.

The results of VCMR on TVR and ActivityNet Captions datasets are reported in Table 7.2. On TVR dataset, ReLoNet is comparable to XML with slightly better performance. ReLoCLNet outperforms all baselines over Recall@10 and Recall@100 metrics. Observe that the performance of ReLoCLNet is lower than FLAT and HAMMER over Recall@1. Since both FLAT and HAMMER adopt fine-grained cross-modal interaction learning, they are more adequate to align video and query for precise moment retrieval. Compared with ReLoNet, ReLoCLNet achieves significant improvements over all evaluation metrics, which demonstrate the effectiveness of the proposed contrastive learning objectives.

On ActivityNet Captions dataset, ReLoNet surpasses the ranking-based methods, MCN and CAL, by large margins over all evaluation metrics. Similarly, ReLoCLNet is superior to ReLoNet thanks to the contrastive learning components. Compared with FLAT and HAMMER, ReLoCLNet outperforms both over Recall@1 but is poorer over Recall@10 and Recall@100. This observation is contrary to that on TVR dataset. Recall that FLAT and HAMMER adopt

[6]Two sets of results are reported for HERO in [215], with and without large-scale pre-training. We choose the version without pre-training as all other models compared here do not use pre-training.

TABLE 7.3: Retrieval efficiency on the TVR dataset.

Method	Retrieval Efficiency	
	Total Time	Average Per Query
XML [198]	39.34 seconds	3.61 milliseconds
HAMMER [199]	2,378.67 seconds	218.33 milliseconds
ReLoNet ReLoCLNet	42.07 seconds	3.86 milliseconds

cross-modal interactions learning between video and query, and we have separate encoders for video and query. In addition, FLAT and HAMMER utilize pre-trained RoBERTa to extract textual features for query, while we simply adopt GloVe embeddings. All these contribute the differences between our results and that of FLAT and HAMMER. Overall, we consider ReLoCLNet achieves comparable effectiveness with FLAT and HAMMER.

7.3.3 Retrieval Efficiency

Here we compare the retrieval efficiency among different methods. We consider VCMR in the validation set of TVR dataset containing $2,179$ videos with $10,895$ queries. The retrieval efficiency is summarized in Table 7.3. The time spent on data pre-processing and feature extraction by pre-trained extractor are not counted since the same process applies to all methods. We used the XML code released by the authors, and re-implemented HAMMER according to their paper as its code is not released. Observe that the retrieval efficiency of our models are comparable to XML, and our models are far more efficient than HAMMER. Although HAMMER performs better on more strict metrics (*e.g.,* Recall@1, IoU=0.7), our models are around 56.71 times faster than HAMMER in retrieval. Note that, ReLoCLNet and ReLoNet have the same retrieval efficiency, because neither VideoCL nor FrameCL introduces additional parameters; and all additional computations of ReLoCLNet happen in training stage.

7.3.4 Ablation Study

Now we study the performance of our models on VR and SVMR subtasks, and the effects of different components.

Video Retrieval Subtask. Table 7.4 reports the results on TVR and ActivityNet Captions datasets. Observe that ReLoNet performs slightly better than XML on TVR, and significantly better than HAMMER on ActivityNet Captions. ReLoCLNet outperforms all baselines by large margins on both datasets. In particular, ReLoCLNet achieves 5.59% improvement in Recall@1

TABLE 7.4: Results of VR subtask on TVR and ActivityNet Captions datasets

Dataset	Method	Recall@k			
		$k=1$	$k=5$	$k=10$	$k=100$
TVR	MCN [46]	0.05	0.38	0.66	3.59
	CAL [197]	0.28	1.02	1.68	8.55
	MEE [216]	7.56	20.78	29.88	73.07
	XML [198]	16.54	38.11	50.41	88.22
	ReLoNet	*16.96*	*39.28*	*51.34*	*88.46*
	ReLoCLNet	**22.13**	**45.85**	**57.25**	**90.21**
ActivityNet	FLAT [199]	5.37	-	29.14	71.64
	HAMMER [199]	5.89	-	30.98	73.38
	ReLoNet	*7.02*	*24.42*	*35.24*	*78.08*
	ReLoCLNet	**9.64**	**28.02**	**40.26**	**79.13**

TABLE 7.5: Results of SVMR subtask on TVR and ActivityNet Captions datasets

Dataset	Method	Recall@1, IoU = μ		
		$\mu=0.3$	$\mu=0.5$	$\mu=0.7$
TVR	MCN [46]	-	13.08	5.06
	CAL [197]	-	12.07	4.68
	ExCL [89]	-	*31.34*	*14.19*
	XML [198]	-	30.75	13.41
	ReLoNet	*48.14*	29.49	13.13
	ReLoCLNet	**49.87**	**31.88**	**15.04**
ActivityNet	FLAT [199]	*57.58*	*39.60*	*22.59*
	HAMMER [199]	**59.18**	**41.45**	**24.27**
	ReLoNet	39.27	23.67	14.55
	ReLoCLNet	42.65	28.54	17.76

comparing with XML on TVR dataset. On ActivityNet Captions dataset, ReLoCLNet obtains 9.64% absolute score in Recall@1, compared with 5.89% of HAMMER.

Temporal Sentence Grounding in Videos Subtask. The results of SVMR on both datasets are reported in Table 7.5. On TVR, ReLoCLNet achieves best performance, and obtains significant improvements against ReLoNet. Compared with ExCL, ReLoCLNet only outperforms by a small margin. ExCL is specially designed for SVMR, with fine-grained cross-modal interactions learning. On ActivityNet Captions, ReLoCLNet is superior to ReLoNet by large margins, which again shows the effectiveness of contrastive learning. However, ReLoCLNet performs worse than FLAT and HAMMER. Because both FLAT and HAMMER inherit their architectures designed for SVMR, which contain sophisticated and computational expensive cross-modal interactions for high-quality moment retrieval. In contrast, ReLoCLNet only relies on simple late fusion of separately encoded query and video features.

TABLE 7.6: The effects of different objectives on TVR dataset (VR=Video Retrieval, ML=Moment Localization, VideoCL=Video Contrastive Learning, and FrameCL=Frame Contrastive Learning)

Objective				VCMR						VR			SVMR					
				Recall@k, IoU=0.5			Recall@k, IoU=0.7			Recall@k			Recall@k, IoU=0.5			Recall@k, IoU=0.7		
VR	ML	VideoCL	FrameCL	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100
✔	✘	✘	✘	-	-	-	-	-	-	16.23	49.33	87.38	-	-	-	-	-	-
✘	✔	✘	✘	-	-	-	-	-	-	-	-	-	30.21	59.81	83.43	13.91	41.55	68.51
✔	✔	✘	✘	5.46	16.65	35.08	2.71	9.37	22.87	16.96	51.34	88.46	29.49	54.06	75.89	13.13	35.46	58.84
✔	✔	✔	✘	6.63	18.16	39.69	3.24	11.78	27.69	*20.69*	*55.70*	*89.71*	29.52	57.32	78.65	13.76	38.26	64.27
✔	✔	✘	✔	*7.21*	*20.04*	*42.45*	*3.75*	*12.77*	*30.32*	19.81	54.38	88.96	*31.75*	*62.20*	*85.99*	*14.73*	*44.60*	*71.44*
✔	✔	✔	✔	**8.03**	**21.37**	**44.10**	**4.15**	**14.06**	**32.42**	**22.13**	**57.25**	**90.21**	**31.88**	**63.89**	**86.67**	**15.04**	**45.24**	**72.12**

Analysis on the Learning Objectives. Table 7.6 reports the contributions of different training objectives on TVR dataset. Note ReLoNet equals to VR+ML objectives, and ReLoCLNet is with all the four objectives. We first analyze the video retrieval (VR) and moment localization (ML) objectives. ReLoNet jointly trains VR and ML objectives for the VCMR task. Comparing VR with ReLoNet, the performance of ReLoNet on video retrieval is slightly better than that of VR, which means the ML objective also contributes to refine video retrieval learning process. In contrast, compared to ML only, ReLoNet underperforms ML on moment localization with marginal performance degradation, which implies that VR objective has negligible negative impact on moment localization.

Now we analyze the effects of VideoCL and FrameCL objectives. Observe that VideoCL contributes to performance improvements on both VCMR and VR, while it achieves marginal improvements on SVMR. Recall that VideoCL adopts noise-contrastive estimation to enlarge the similarities of matched video-query pairs, and reduce similarities between unpaired videos and queries; this is in line with video retrieval objective. Thus, it is beneficial to video retrieval learning. ReLoNet with FrameCL outperforms ReLoNet on all the three tasks. FrameCL aims to distinguish the matching moment from non-matching moment within a video. In this case, FrameCL guides the model to search for boundaries of target moment for precise moment localization. In fact, the matching between query and video is largely based on the matching moment in the video. In this sense, by highlighting matching moment, FrameCL does contribute to video retrieval task as well. Combining VideoCL and FrameCL, ReLoCLNet further boosts the performances on all three tasks by incorporating the advantages of both VideoCL and FrameCL.

Qualitative Analysis. Figure 7.5 plots Recall@1 and Recall@10 of VCMR performances on TVR dataset over different IoU thresholds. We evaluate 9 different IoU(μ) values, from 0.1 to 0.9. ReLoCLNet consistently outperforms ReLoNet, and relative performance improvements of ReLoCLNet are larger under more strict metrics. For instance, compared with ReLoNet,

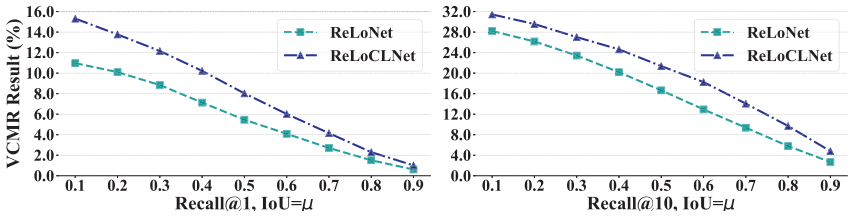


FIGURE 7.5: Recall@1 and Recall@10 of VCMR on TVR dataset over different IoU thresholds.

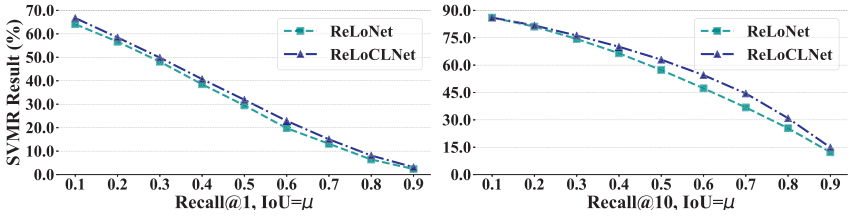


FIGURE 7.6: Recall@1 and Recall@10 of SVMR on TVR dataset over different IoU thresholds.

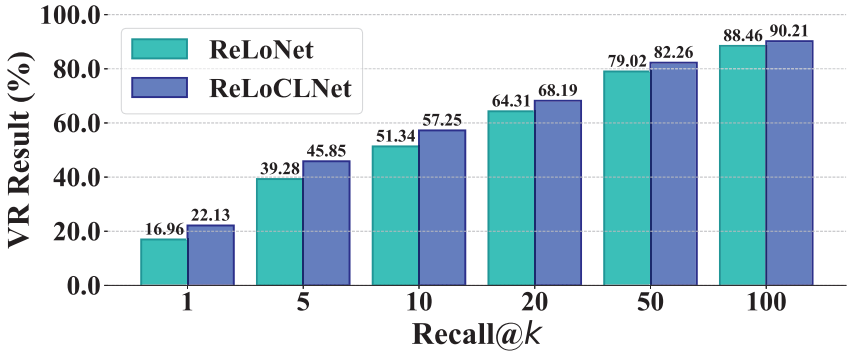


FIGURE 7.7: Recall@K of VR on TVR dataset over different K.

ReLoCLNet achieves 47.07% relative gains (8.03 vs 5.46) in Recall@1, IoU=0.5 versus 28.35% relative gains (21.37 vs 16.65) in Recall@10, IoU=0.5.

Figure 7.6 plots Recall@1 and Recall@10 of SVMR over different IoU thresholds, and similar observations hold on this task. Figure 7.7 plots the video retrieval (VR) results of ReLoNet and ReLoCLNet over different recall thresholds on TVR dataset. Similarly, ReLoCLNet surpasses ReLoNet over all thresholds, and the relative performance improvement ratio is larger under more strict metrics.

Finally, we show two retrieval examples in Figure 7.8 from ActivityNet Captions dataset. The figure shows the predicted moments by ReLoCLNet and ReLoNet+FrameCL are closer to ground truth than that by ReLoNet and ReLoNet+VideoCL, which demonstrates the effectiveness of FrameCL module. Note FrameCL is designed to maximize the mutual information between query and frames within the target moment, and to minimize the MI between the

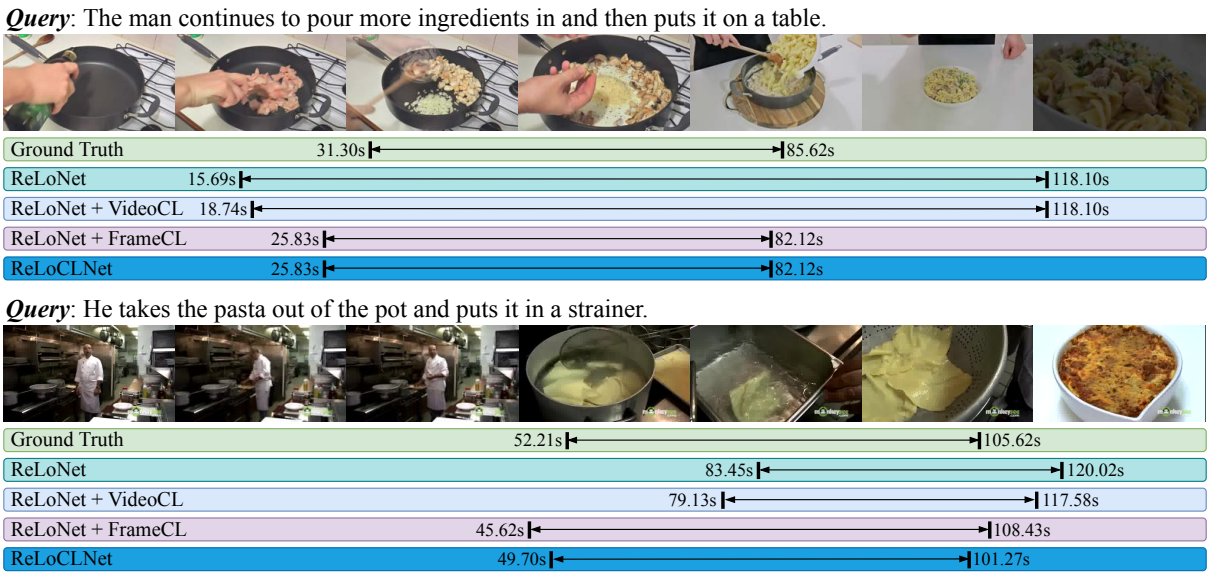


FIGURE 7.8: Visualization of moment localization predictions by ReLoNet, ReLoCLNet, and ReLoNet with VideoCL or FrameCL, for two queries on ActivityNet Captions dataset.

query and frames outside target moment. With FrameCL, the model is guided to search for the boundaries within the region of target moment.

7.4 Summary

VCMR aims to retrieve a temporal moment that semantically corresponds to a given text query from a collection of untrimmed and unsegmented videos, *i.e.,* a video corpus. As an extension of TSGV (SVMR), VCMR is more challenging and closer to practical application scenarios. In this chapter, we propose a Retrieval and Localization Network with Contrastive Learning (ReLoCLNet) for video corpus moment retrieval (VCMR) task. Specifically, we introduce two contrastive learning objectives (VideoCL and FrameCL) on top of a unimodal encoding approach, ReLoNet, to address the contradiction between retrieval efficiency and retrieval quality. The VideoCL objective guides the video and query encoders to shorten the distance of matching videos and queries while enlarge the non-matching pairs. The FrameCL objective works at frame-level to simulate the fine-grained cross-modal interactions between visual and textual features within a video. Through extensive experimental studies, we show that ReLoCLNet addresses VCMR with high efficiency, and its retrieval accuracy is comparable with state-of-the-art methods which are much costly in terms of computation. Compared with the expensive cross-model interaction learning, we show that unimodal encoding with contrastive learning is a promising direction to explore for video corpus moment retrieval. Nevertheless, the performance of existing approaches, including our proposed method, are still inferior and far apart

to be deployed in the real-world applications. Thus, this task needs more attention to move it forward.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis, we have presented several new ideas to effectively utilize the concepts, *i.e.,* question answering and named entity recognition in NLP, to reformulate the TSGV problem. We also investigate the annotation distributional bias issue among TSGV benchmark datasets. Besides, we explore an extension of TSGV, video corpus moment retrieval (VCMR), and devise an effective solution to solve it. In the first chapter, Chapter 1 - *Introduction*, we highlight motivations of our research problem and its applications in video question answering, video grounded dialogue, intelligent video surveillance, and intelligent video creation/edit. We also discuss the main challenges regarding the large discrepancy between video and language modality, the cross-modal reasoning between them, as well as the sparsity issue of moment boundary predictions. Chapter 2 - *Literature Review* provides a summary of fundamental concepts in TSGV and current research status. The next five chapters detail our key contributions, in which three chapters serve TSGV, one chapter studies the bias issue in TSGV, and one chapter associates with VCMR.

Chapter 3 - *Span-based Question Answering for TSGV* presents a new idea which formulates TSGV as a multimodal span-based question answering task. We investigate the similarities and differences between TSGV and span-based QA. Based on the observations, we devise a span-based QA baseline (VSLBase) and its improved version (VSLNet) to solve TSGV and mitigate the differences between TSGV and span-based QA, respectively. Experiments demonstrate that formulating TSGV as span-based QA is a promising research direction, and the proposed models are simple yet effective. Nevertheless, existing TSGV methods, including VSLNet, suffer from a common defect, that is, the performance degradation along with the increase of video length. To tackle this issue, Chapter 4 - *Multi-Paragraph Question Answering for TSGV*

introduces the concepts of MPQA by regarding a long video as a document, and splitting the long video into multiple short clip segments (*i.e.,* short videos), where each short video is treated as a paragraph. Due to the different data nature between document and video, how to properly split long video into clip segments is challenging. To this end, we present a multiscale split-and-concatenation strategy to partition long video into clips of different lengths. Equipped with the strategy, we extend VSLNet to VSLNet-L to solve the performance degradation issue and boost the TSGV performance. Furthermore, our analysis reveals that the target moment is usually a very small portion of the video, making positive and negative samples imbalanced. As TSGV is to start/end boundaries of target moment. Thus, the sparsity is a major concern of TSGV, *e.g.,* catching two frames among thousands. Chapter 5 - *Parallel Attention Network with Sequence Matching* aims to tackle the moment boundary prediction issue in TSGV. Specifically, we introduce the concepts of NER by emphasizing the "sequence" nature of frames. We devise a loose region label annotation strategy and a parallel attention network with sequence matching to alleviate the sparsity issue of TSGV and boost its performance.

In addition, Chapter 6 - *Towards Debiasing TSGV* studies the bias issue in TSGV. Our analysis reveals that commonly used benchmark datasets contain substantial distributional bias in moment annotations, and TSGV models tend to capture these statistical regularities to achieve spurious success on evaluation and result in poor generalization ability. To mitigate the bias issue, we develop two strategies, data debiasing and model debiasing, to "force" a TSGV model to focus on cross-modal interactions by correcting the dataset distribution via video truncation (data debiasing) and altering biased predictions via unimodal branch learning (model debiasing). Experiments show that both strategies effectively suppress the bias and improve the model generalization ability. Chapter 7 - *Video Corpus Moment Retrieval with Contrastive Learning* further explores an extension task of TSGV, *i.e.,* video corpus moment retrieval (VCMR). We investigate two common VCMR structures and show the contradiction between high efficiency and high-quality retrieval in these VCMR solutions. We then devise a contrastive learning-based framework to remedy such contradiction, and demonstrate its effectiveness via extensive experiments.

In conclusion, TSGV has been a fruitful research problem because of its significant impact on a wide range of downstream applications. The problem not only relates to CV and NLP, but also requires the sophisticated interactions between these two domains. As such, existing models have to consider techniques in both CV and NLP. In this thesis, we have walked the readers through different ideas to solve TSGV problem and improve the TSGV performance. However, in a bigger picture, as vision and languages are changing and evolving, more new challenges for TSGV and its related tasks will arise. To conclude, we will briefly discuss

several potential directions for future work.

8.2 Future Work

8.2.1 Effective Feature Extractor(s)

Feature quality directly affects TSGV performance. Existing solutions extract visual and textual features independently using corresponding pre-trained visual and textual extractors. Thus there is a large gap between extracted visual and textual features for in different feature spaces. Although TSGV methods attempt to project them into the same feature space, the natural gap between them is hard to be eliminated. There may be also differences between TSGV datasets and the datasets used to pre-train feature extractors, which leads to information loss or inaccurate representations.

Recently, Zhang et al. [56] develop a single-stream feature extraction framework for TSGV task, following BERT [10]. Visual and textual features are concatenated and jointly encoded with stacked transformer blocks. However, the visual and textual features remain separately generated by different pre-trained extractors. Xu et al. [137] propose a pre-training strategy for TSGV by constructing a large-scale synthesized dataset with TSGV annotations. Inspired by ViT [217], Cao et al. [138] develop a video cubic embedding module to extract 3D visual tokens and learn video content from scratch without reliance on pre-trained visual feature extractor. Although they adopt GloVe [26] embeddings for query, the issues of feature gap are well alleviated. Considering recent advances in video-based vision-language pre-training (*e.g.,* BVET [218], ActBERT [23], ClipBERT [219], and VideoCLIP [220]), dedicated or more effective feature extractors for TSGV are much expected.

8.2.2 TSGV with Multiple Answers

Existing TSGV benchmark datasets generally hold an implicit assumption that there is only one ground-truth moment exist in input video for a query. In reality, a query may describe multiple disjoint moments in a video. Lei et al. [221] present a unified benchmark dataset named QVHighlights for both TSGV and highlight detection tasks. Given a query, QVHighlights provides one or multiple disjoint moments in a video, enabling a more realistic evaluation of TSGV methods. The authors then propose Moment-DETR to solve the TSGV as a direct set prediction problem, inspired by DETR [5]. From query perspective, Bao et al. [135] convert TSGV to a dense events grounding task, which aims to jointly localize multiple moments described in a paragraph, *i.e.,* multiple queries. Jointly localizing multiple moments could help to alleviate

the bias issue in TSGV. This joint localization may also help to improve accuracy as moments are semantically correlated and temporally coordinated by their order in a video. In general, multiple answers for a query is a novel task inherited from the standard TSGV task. The setting is more realistic and less-biased.

8.2.3 Spatio-Temporal Sentence Grounding in Videos

Spatio-temporal sentence grounding in videos (STSGV) is another extension of TSGV. STSGV aims to sequentially localize the referring instances from a sequence of continuous frames in a video, *i.e.,* a spatio-temporal tube, that semantically correspond to the given sentence query. STSGV is more complicated, since the task requires to not only localize the temporal boundaries of the event in video, but also detect bounding boxes among frames within the temporal boundaries. Recently, a series of work [222–228] is proposed for this problem. A number of datasets are available, including VID-sentence [229] which is based on ImageNet video object detection, ActivityNet-SRL [222] from existing caption and grounding datasets, VidSTG [223], and HC-STVG [227].

Despite multiple datasets being made available, annotating spatio-temporal tubes from video is even difficult and labor-intensive, compared with TSGV annotation. Thus, many methods [229–233] seek to solve STSGV under weakly-supervised setting, which do not require fully annotated dataset. Although some promising results are achieved, STSGV remains under explored and is far apart to be addressed.

8.2.4 Multi-modal Temporal Grounding in Video

TSGV is a form of temporal video grounding using text sentence as query, *i.e.,* language modality. Other modalities, such as audio, image, and short video clip, may also serve as queries for temporal video grounding. In fact, audio-visual event localization (AVEL) [234–240] is to retrieve the synchronized video segment for a given audio content from an untrimmed video. The task of Image-to-Video Retrieval (IVR) [241–244] localizes video segments that contain similar activity as in a query image. Similarly, given a query video and a reference video, video re-localization (VRL) [245–248] localizes a segment in the reference video that semantically corresponds to the query video. Conceptually, the query is in the form of audio in AVEL, appearance vision in IVR, and motion vision in VRL, respectively. Compared to TSGV, temporal video grounding with such modalities are not fully exploited.

Different query modalities could provide extra guidance to boost performance for moment localization in videos. For instance, audio signals (*e.g.,* dog bark, noise in kitchen) offer auxiliary clues [100, 249] for precise localization. Converting voice in video (if exists) into subtitle texts using ASR [250] could provide relevant information for the cross-modal alignment between video and query. From the perspective of query, different modalities of query (*e.g.,* audio, sentence, and image) that describe the same event could perform cross-validation of the retrieved results. Although TSGV, AVEL, IVR, and VRL accept different input modalities as query, there is lack of a unified framework, which is suitable for all settings.

8.2.5 Video Corpus Moment Retrieval

Video corpus moment retrieval (VCMR) extends video sources for TSGV. Instead of localizing moments in a single video, VCMR aims to retrieve a matching moment to a query from a collection of untrimmed and unsegmented videos. Escorcia et al. [197] first extend TSGV to VCMR, and devise clip-query alignment model to compare query features with uniformly partitioned video segments. Lei et al. [198] construct the TVR dataset, where videos come with associated textual subtitles. Then Lei et al. [251] further extend TVR to a multilingual version named mTVR, containing both English and Chinese queries.

A few recent methods [38, 199, 215, 252–254] tackle the VCMR problems. Zhang et al. [199] develop a hierarchical multi-modal encoder to learn multimodal interactions at both coarse- and fine-grained granularities. Hou et al. [254] develop a two-step multimodal fusion for precise and efficient moment retrieval. In Chapter 7, we propose to introduce contrastive learning to replace the time-consuming multimodal interaction strategy in VCMR to achieve a balance between efficiency and retrieval accuracy. In general, VCMR contains two sub-tasks, *i.e.,* video retrieval and moment localization. If a TSGV model is directly adapted, the query needs to interact with every video in the corpus, which is infeasible. However, VCMR is more closer to the practical scenarios as videos are ubiquitous.

List of Author's Publications[1]

Publications covered in the thesis

- **Hao Zhang**, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "The Elements of Temporal Sentence Grounding in Videos: A Survey and Future Directions". Under review.
- **Hao Zhang**, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "Towards Debiasing Temporal Sentence Grounding in Video". Under review.
- **Hao Zhang**, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. "Natural Language Video Localization: A Revisit in Span-based Question Answering Framework". In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages: 1-15, 2021.
- **Hao Zhang**, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Siow Mong Rick Goh. "Parallel Attention Network with Sequence Matching for Video Grounding". In *Findings of the Association for Computational Linguistics: ACL-IJCNLP (Long Papers)*, pages: 776–790, Online, 2021.
- **Hao Zhang**, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. "Video Corpus Moment Retrieval with Contrastive Learnin"g. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (Long Papers) (SIGIR)*, pages: 685–695, Virtual Event, Canada, 2021.
- **Hao Zhang**, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. "Span-based Localizing Network for Natural Language Video Localization". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pages 6543–6554, Online, 2020.

[1]The superscript * indicates joint first authors.

Publications not covered in the thesis

- Sicheng Yu, Jing Jiang, **Hao Zhang**, Yulei Niu, Qianru Sun, and Lidong Bing. "Interventional Training for Out-Of-Distribution Natural Language Understanding". Under review.

- Sicheng Yu, Qianru Sun, **Hao Zhang**, and Jing Jiang. "Translate-Train Embracing Translationese Artifacts". In *The 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL)*, 2022.

- En Yen Puang, **Hao Zhang**, Hongyuan Zhu, Wei Jing. "Hierarchical Point Cloud Encoding and Decoding with Lightweight Self-Attention based Model". In *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, pages: 4542-4549, 2022.

- Fuzhao Xue, Aixin Sun, **Hao Zhang**, Jinjie Ni, Eng-Siong Chng. "An Embarrassingly Simple Model for Dialogue Relation Extraction". In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages: 1-5, Singapore 2022.

- Sicheng Yu, **Hao Zhang**, Wei Jing, Jing Jiang. "Context Modeling with Evidence Filter for Multiple Choice Question Answering". In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages: 1-5, Singapore 2022.

- Sicheng Yu, **Hao Zhang**, Yulei Niu, Qianru Sun, and Jing Jiang. "COSY: COunterfactual SYntax for Cross-Lingual Understanding". In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL-IJCNLP)*, pages: 577–589, Online, 2021.

- Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, **Hao Zhang**, and Wei Lu. "Interventional Video Grounding with Dual Contrastive Learning". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages: 2765-2775, Online, 2021.

- Fuzhao Xue, Aixin Sun, **Hao Zhang**, and Eng Siong Chng. "GDPNet: Refining Latent Multi-View Graph for Relation Extraction". In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages: 14194-14202, Virtual Event, California USA, 2021.

- Joey Tianyi Zhou*, **Hao Zhang***, Di Jing, and Xi Peng. "Dual Adversarial Transfer for Sequence Labeling". In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 43, pages: 434-446, 2021.

- Ming Yan, **Hao Zhang**, Di Jin, and Joey Tianyi Zhou. "Multi-source Meta Transfer for Low Resource Multiple-Choice Question Answering". In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pages: 7331–7341, Online, 2020.

- Tianying Wang*, Wei Qi Toh*, **Hao Zhang***, Xiuchao Sui, Shaohua Li, Yong Liu, and Wei Jing. "RoboCoDraw: Robotic Avatar Drawing with GAN-based Style Transfer and Time-efficient Path Optimization". In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages: 10402-10409, New York, USA, 2020.

- Tianying Wang, **Hao Zhang**, Wei Qi Toh, Hongyuan Zhu, Cheston Tan, Yan Wu, Yong Liu, and Wei Jing. "Efficient Robotic Task Generalization Using Deep Model Fusion Reinforcement Learning". In *Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages: 148-153, Dali, China, 2019.

- Joey Tianyi Zhou*, **Hao Zhang***, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. "Dual Adversarial Neural Transfer for Low-Resource Named Entity Recognition". In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pages: 3461–3471, Florence, Italy, 2019.

- Joey Tianyi Zhou*, **Hao Zhang***, Di Jing, Xi Peng, Yang Xiao, and Zhiguo Cao. "RoSeq: Robust Sequence Labeling". In *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 31, pages: 2304-2314, 2019.

- Joey Tianyi Zhou, Meng Fang, **Hao Zhang**, Chen Gong, Xi Peng, Zhiguo Cao, and Rick Siow Mong Goh. "Learning With Annotation of Various Degrees". In *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 30, pages: 2794-2804, 2019.

Bibliography

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 14

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 1, 31, 125

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019. 1, 2

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1, 2

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[9] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019. 1, 3, 5, 13, 115, 125

[11] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5754–5764, 2019. 1

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015. 1, 47, 59, 74, 108

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252, 2017. 1

[15] H. Huang, Chenguang Zhu, Y. Shen, and W. Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *Proceedings of the International Conference on Learning Representations*, 2018. 1, 2, 27, 28, 42

[16] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations*, 2017. 1, 2, 27, 28, 42, 44

[17] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, 2017. 1

[18] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, 2019. 1

[19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32, pages 13–23, 2019. 1, 70, 80, 109

[20] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 112

[21] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 70

[22] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 5100–5111, 2019. 70, 80, 109

[23] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 80, 109, 125

[24] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 3, 4, 14, 96

[25] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3, 4, 14, 43, 48, 61, 77, 95, 115

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014. 3, 5, 13, 44, 48, 61, 77, 89, 95, 115, 125

[27] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ramakant Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5277–5285, 2017. 3, 16, 21, 35, 36, 47, 49, 52, 62, 63, 70, 95, 98

[28] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 245–253, 2019. 16, 22, 49, 52, 62, 63, 77

[29] Huijuan Xu, Kun He, Bryan A. Plummer, L. Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9062–9069, 2019. 17, 23, 49, 62, 63

[30] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 162–171, 2018. 17, 24, 49, 62, 63, 77

[31] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *Advances in Neural Information Processing Systems*, pages 536–546, 2019. 17, 24, 62, 63, 70, 77, 98

[32] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks formoment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877, 2020. 3, 17, 25, 34, 37, 62, 63, 77, 79, 98

[33] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855, 2018. 6, 28, 57

[34] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6543–6554, 2020. 7, 28, 41, 63, 77, 79, 89, 95, 96, 108

[35] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 7, 28, 38, 57

[36] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Siow Mong Rick Goh. Parallel attention network with sequence matching for video grounding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 776–790, 2021. 8, 28, 69

[37] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Towards debiasing temporal sentence grounding in video. *ArXiv*, abs/2111.04321, 2021. 8, 88

[38] Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Video corpus moment retrieval with contrastive learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 685–695, 2021. 8, 103, 127

[39] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. The elements of temporal sentence grounding in videos: A survey and future directions. *ArXiv*, abs/2201.08071, 2022. 8, 11

[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ArXiv*, abs/1301.3781, 2013. 13

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. 13, 35, 115

[42] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28, pages 3294–3302, 2015. 14

[43] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017. 14

[44] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, 2019. 14

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ArXiv*, abs/1409.1556, 2014. 14

[46] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5803–5812, 2017. 16, 21, 35, 52, 116, 118

[47] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *Proceedings of the 26th ACM International Conference on Multimedia*, pages 843–851, 2018. 22

[48] Aming Wu and Yahong Han. Multi-modal circulant fusion for video-to-language and backward. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1029–1035, 2018. 16, 22

[49] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8199–8206, 2019. 17, 23, 49, 62

[50] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. *ArXiv*, abs/1804.05113, 2018. 23

[51] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. Boundary proposal network for two-stage natural language video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2986–2994, 2021. 23

[52] Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. Adaptive proposal generation network for temporal sentence localization in videos. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9292–9301, 2021. 17, 23

[53] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5783–5792, 2017. 17, 23

[54] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1247–1257, 2019. 17, 24, 49, 62, 77

[55] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. Cross-modal interaction networks for query-based moment retrieval in videos. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 655–664, 2019. 17, 25

[56] Mingxing Zhang, Yang Yang, Xinghan Chen, Yanli Ji, Xing Xu, Jingjing Li, and Heng Tao Shen. Multi-stage aggregated transformer network for temporal language localization in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12669–12678, 2021. 17, 26, 125

[57] Hao Wang, Zheng-Jun Zha, Liang Li, Dong Liu, and Jiebo Luo. Structured multi-level interaction network for video moment localization via language query. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2021. 26

[58] Jialin Gao, Xin Sun, Mengmeng Xu, Xi Zhou, and Bernard Ghanem. Relation-aware video reading comprehension for temporal language grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3978–3988, 2021. 26

[59] Junyu Gao and Changsheng Xu. Fast video moment retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1523–1532, 2021. 17, 26

[60] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1380–1390, 2018. 21, 35

[61] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 15–24, 2018. 22, 47, 49, 52, 62, 63, 98

[62] Songyang Zhang, Jinsong Su, and Jiebo Luo. Exploiting temporal relationships in video moment localization with natural language. In *Proceedings of the 27th ACM International Conference on Multimedia*, page 1230–1238, 2019. 22

[63] Ke Ning, Ming Cai, Di Xie, and Fei Wu. An attentive sequence to sequence translator for localizing video clips by natural language. *IEEE Transactions on Multimedia*, 22(9): 2434–2443, 2020. 22

[64] Bin Jiang, Xin Huang, Chao Yang, and Junsong Yuan. Cross-modal video moment retrieval with spatial and language-temporal attention. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, page 217–225, 2019. 22

[65] Yawen Zeng, Da Cao, Xiaochi Wei, Meng Liu, Zhou Zhao, and Zheng Qin. Multi-modal relational graph for cross-modal video moment retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2215–2224, 2021. 22

[66] Ke Ning, Lingxi Xie, Jianzhuang Liu, Fei Wu, and Qi Tian. Interaction-integrated network for natural language moment localization. *IEEE Transactions on Image Processing*, 30:2538–2548, 2021. 22

[67] Shaoning Xiao, Long Chen, Jian Shao, Yueting Zhuang, and Jun Xiao. Natural language video localization with learnable moment proposals. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4008–4017, 2021. 23, 24

[68] Yupeng Hu, Meng Liu, Xiaobin Su, Zan Gao, and Liqiang Nie. Video moment localization via deep cross-modal hashing. *IEEE Transactions on Image Processing*, 30: 4667–4677, 2021. 24

[69] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12168–12175, 2020. 25, 62, 63, 77

[70] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Fine-grained iterative attention network for temporal language localization in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 4280–4288, 2020. 25

[71] Ziyang Ma, Xianjing Han, Xuemeng Song, Yiran Cui, and Liqiang Nie. Hierarchical deep residual reasoning for temporal moment localization. In *Proceedings of the ACM Multimedia Asia Conference*, 2021. 25

[72] Zongmeng Zhang, Xianjing Han, Xuemeng Song, Yan Yan, and Liqiang Nie. Multimodal interaction graph convolutional network for temporal language localization in videos. *IEEE Transactions on Image Processing*, 30:8265–8277, 2021. 25

[73] Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. Jointly cross- and self-modal graph attention network for query-based moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 4070–4078, 2020. 25

[74] Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1841–1851, 2020. 25

[75] Daizong Liu, Xiaoye Qu, and Pan Zhou. Progressively guide to attend: An iterative alignment framework for temporal sentence grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9302–9311, 2021. 25

[76] Wen Wang, Jian Cheng, and Siyu Liu. Dct-net: A deep co-interactive transformer network for video temporal grounding. *Image and Vision Computing*, 110:104183, 2021. 25

[77] Bingbin Liu, Serena Yeung, Edward Chou, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Temporal modular networks for retrieving complex compositional activities in videos. In *Proceedings of the European Conference on Computer Vision*, pages 552–568, 2018. 25

[78] Songyang Zhang, Houwen Peng, Jianlong Fu, Yijuan Lu, and Jiebo Luo. Multi-scale 2d temporal adjacency networks for moment localization with natural language. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 26

[79] Qi Zheng, Jianfeng Dong, Xiaoye Qu, Xun Yang, Shouling Ji, and Xun Wang. Progressive localization networks for language-based moment localization. *ArXiv*, abs/2102.01282, 2021. 26

[80] Yupeng Hu, Liqiang Nie, Meng Liu, Kun Wang, Yinglong Wang, and Xian-Sheng Hua. Coarse-to-fine semantic alignment for cross-modal moment localization. *IEEE Transactions on Image Processing*, 30:5933–5943, 2021. 26

[81] Zixi Jia, Minglin Dong, Jingyu Ru, Lele Xue, Sikai Yang, and Chunbo Li. Stcm-net: A symmetrical one-stage network for temporal language localization in videos. *Neurocomputing*, 471:194–207, 2022. 26

[82] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. Vlg-net: Video-language graph matching network for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3224–3234, 2021. 26

[83] Qingbao Huang, Jielong Wei, Yi Cai, Changmeng Zheng, Junying Chen, Ho-fung Leung, and Qing Li. Aligned dual channel graph convolutional network for visual question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020. 26

[84] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 26

[85] Ziyue Wu, Junyu Gao, Shucheng Huang, and Changsheng Xu. Diving into the relations: Leveraging semantic and visual structures for video moment retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2021. 26

[86] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255, 2019. 26

[87] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. *ArXiv*, abs/2109.04872, 2021. 26

[88] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9159–9166, 2019. 27, 47, 49, 62, 63, 70, 82, 98

[89] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. ExCL: Extractive Clip Localization Using Natural Language Descriptions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Short Papers)*, pages 1984–1990, 2019. 27, 28, 45, 49, 62, 77, 118

[90] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *Proceedings of the International Conference on Learning Representations*, 2018. 27, 28, 44, 55, 73

[91] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. DEBUG: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5144–5153, 2019. 27, 49, 62, 63, 70, 77, 79, 82, 108

[92] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. Rethinking the bottom-up framework for query-based video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10551–10558, 2020. 27, 62, 63, 77, 79, 108

[93] Binjie Zhang, Yu Li, Chun Yuan, Dejing Xu, Pin Jiang, and Ying Shan. A simple yet effective method for video temporal grounding with cross-modality attention. *ArXiv*, abs/2009.11232, 2020. 27

[94] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10287–10296, 2020. 27, 62, 63, 70, 77, 98

[95] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10810–10819, 2020. 62, 63, 70, 77, 82

[96] Kun Li, Dan Guo, and Meng Wang. Proposal-free video grounding with contextual pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1902–1910, 2021.

[97] Hao Zhou, Chongyang Zhang, Yan Luo, Yanjun Chen, and Chuanping Hu. Embracing uncertainty: Decoupling and de-bias for robust temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8454, 2021. 27

[98] Xinfang Liu, Xiushan Nie, Junya Teng, Li Lian, and Yilong Yin. Single-shot semantic matching network for moment localization in videos. *ACM Trans. Multimedia Comput. Commun. Appl.*, 17(3), 2021. 27

[99] Shaoxiang Chen and Yu-Gang Jiang. Hierarchical visual-textual graph for temporal activity localization via language. In *Proceedings of European Conference on Computer Vision*, pages 601–618, 2020. 28

[100] Shaoxiang Chen, Wenhao Jiang, Wei Liu, and Yu-Gang Jiang. Learning modality interaction for temporal sentence localization and event captioning in videos. In *Proceedings of European Conference on Computer Vision*, pages 333–351, 2020. 28, 127

[101] Yi-Wen Chen, Yi-Hsuan Tsai, and Ming-Hsuan Yang. End-to-end multi-modal video temporal grounding. In *Advances in Neural Information Processing Systems*, volume 34, pages 1–12, 2021. 28

[102] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, volume 29, 2016. 28

[103] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8175–8182, 2019. 28, 49

[104] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, HONGDONG LI, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2464–2473, 2020. 28, 77

[105] Yang Zhao, Zhou Zhao, Zhu Zhang, and Zhijie Lin. Cascaded prediction network via segment tree for temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4197–4206, 2021.

[106] Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. Interventional video grounding with dual contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2765–2775, 2021. 28

[107] Guoqiang Liang, Shiyu Ji, and Yanning Zhang. Local-enhanced interaction for temporal moment localization. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, page 201–209, 2021.

[108] Xinli Yu, Mohsen Malmir, Xin He, Jiangning Chen, Tong Wang, Yue Wu, Yue Liu, and Yang Liu. Cross interaction network for natural language guided video moment retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1860–1864, 2021. 28

[109] Haoyu Tang, Jihua Zhu, Lin Wang, Qinghai Zheng, and Tianwei Zhang. Multi-level query interaction for temporal language grounding. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[110] Haoyu Tang, Jihua Zhu, Meng Liu, Zan Gao, and Zhiyong Cheng. Frame-wise cross-modal matching for video moment retrieval. *IEEE Transactions on Multimedia*, 2021.

[111] Zijian Zhang, Zhou Zhao, Zhu Zhang, Zhijie Lin, Qi Wang, and Richang Hong. Temporal textual localization in video via adversarial bi-directional interaction networks. *IEEE Transactions on Multimedia*, 23:3306–3317, 2021. 29

[112] Shanshan Qi, Luxi Yang, Chunguo Li, and Yongming Huang. Collaborative spatial-temporal interaction for language-based moment retrieval. In *Proceedings of the 13th International Conference on Wireless Communications and Signal Processing*, 2021.

[113] Cristian Rodriguez-Opazo, Edison Marrese-Taylor, Basura Fernando, Hongdong Li, and Stephen Gould. Dori: Discovering object relationships for moment localization of a natural language query in a video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1079–1088, 2021. 29

[114] Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11235–11244, 2021. 29

[115] Lingyu Zhang and Richard J. Radke. Natural language video moment localization through query-controlled temporal convolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1–9, 2022. 28

[116] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074, 2016. 28, 70

[117] Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471, 2019.

[118] Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, 2020. 28

[119] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 28, 87

[120] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8393–8400, 2019. 29, 49, 62, 63

[121] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 30

[122] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003. 30

[123] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang. A2-rl: Aesthetics aware reinforcement learning for image cropping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8193–8201, 2018. 30

[124] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 334–343, 2019. 30, 49, 62, 63

[125] Jie Wu, Guanbin Li, Si Liu, and Liang Lin. Tree-structured policy based progressive reinforcement learning for temporally language grounding in video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12386–12393, 2020. 30, 62, 63, 77, 98

[126] Da Cao, Yawen Zeng, Xiaochi Wei, Liqiang Nie, Richang Hong, and Zheng Qin. Adversarial video moment retrieval by jointly modeling ranking and localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 898–906, 2020. 30

[127] Da Cao, Yawen Zeng, Meng Liu, Xiangnan He, Meng Wang, and Zheng Qin. Strong: Spatio-temporal reinforcement learning for cross-modal video moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 4162–4170, 2020. 30

[128] Meera Hahn, Asim Kadav, James M Rehg, and Hans Peter Graf. Tripping through time: Efficient localization of activities in videos. In *Proceedings of the British Machine Vision Conference*, 2020. 30

[129] Xiaoyang Sun, Hanli Wang, and Bin He. Maban: Multi-agent boundary-aware network for natural language moment retrieval. *IEEE Transactions on Image Processing*, 30: 5589–5599, 2021. 30

[130] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *Proceedings of the European Conference on Computer Vision*, pages 200–216, 2018. 31

[131] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 31

[132] Hao Wang, Zheng-Jun Zha, Xuejin Chen, Zhiwei Xiong, and Jiebo Luo. Dual path interaction network for video moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 4116–4124, 2020. 31

[133] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *International Conference on Learning Representations*, 2021. 31

[134] Xinpeng Ding, Nannan Wang, Shiwei Zhang, De Cheng, Xiaomeng Li, Ziyuan Huang, Mingqian Tang, and Xinbo Gao. Support-set based cross-supervision for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11573–11582, 2021. 31

[135] Peijun Bao, Qian Zheng, and Yadong Mu. Dense events grounding in video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 920–928, 2021. 31, 125

[136] Wenbo Gou, Wen Shi, Jian Lou, Lijie Huang, Pan Zhou, and Ruixuan Li. Sneak: Synonymous sentences-aware adversarial attack on natural language video localization. *ArXiv*, abs/2112.04154, 2021. 31

[137] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7220–7230, 2021. 31, 125

[138] Meng Cao, Long Chen, Mike Zheng Shou, Can Zhang, and Yuexian Zou. On pursuit of designing multi-modal transformer for video grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9810–9823, 2021. 31, 125

[139] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2019. 32

[140] Mingfei Gao, Larry Davis, Richard Socher, and Caiming Xiong. WSLLN:weakly supervised natural language localization networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1481–1487, 2019. 32

[141] Zhenfang Chen, Lin Ma, Wenhan Luo, Peng Tang, and Kwan-Yee Kenneth Wong. Look closer to ground better: Weakly-supervised temporal grounding of sentence in video. *ArXiv*, abs/2001.09308, 2020. 33

[142] Minuk Ma, Sunjae Yoon, Junyeong Kim, Youngjoon Lee, Sunghun Kang, and Chang D Yoo. Vlanet: Video-language alignment network for weakly-supervised video moment retrieval. In *Proceedings of the European Conference on Computer Vision*, pages 156–171, 2020. 33

[143] Jie Wu, Guanbin Li, Xiaoguang Han, and Liang Lin. Reinforcement learning for weakly supervised temporal grounding of natural language in untrimmed videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1283–1291, 2020. 33

[144] Zhu Zhang, Zhou Zhao, Zhijie Lin, jieming zhu, and Xiuqiang He. Counterfactual contrastive learning for weakly-supervised vision-language grounding. In *Advances in Neural Information Processing Systems*, volume 33, pages 18123–18134, 2020. 33

[145] Cheng Da, Yanhao Zhang, Yun Zheng, Pan Pan, Yinghui Xu, and Chunhong Pan. Asynce: Disentangling false-positives for weakly-supervised video grounding. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 1129–1137, 2021. 33

[146] Zheng Wang, Jingjing Chen, and Yu-Gang Jiang. Visual co-occurrence alignment learning for weakly-supervised video moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 1459–1468, 2021. 33

[147] Yuechen Wang, Wengang Zhou, and Houqiang Li. Fine-grained semantic alignment network for weakly supervised temporal language grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 89–99, 2021. 33

[148] Jiabo Huang, Yang Liu, Shaogang Gong, and Hailin Jin. Cross-sentence temporal and semantic relations in video activity localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7199–7208, 2021. 33

[149] Wenfei Yang, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Local correspondence network for weakly supervised temporal sentence grounding. *IEEE Transactions on Image Processing*, 30:3252–3262, 2021. 33

[150] Junya Teng, Xiankai Lu, Yongshun Gong, Xinfang Liu, Xiushan Nie, and Yilong Yin. Regularized two granularity loss function for weakly supervised video moment retrieval. *IEEE Transactions on Multimedia*, 2021. 33

[151] Yuechen Wang, Jiajun Deng, Wengang Zhou, and Houqiang Li. Weakly supervised temporal adjacent network for language grounding. *IEEE Transactions on Multimedia*, 2021. 33

[152] Reuben Tan, Huijuan Xu, Kate Saenko, and Bryan A Plummer. Logan: Latent graph co-attention network for weakly-supervised video moment retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2083–2092, 2021. 32, 33

[153] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 33, 34

[154] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. Weakly-supervised video moment retrieval via semantic completion network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11539–11546, 2020. 34

[155] Yijun Song, Jingwen Wang, Lin Ma, Zhou Yu, and Jun Yu. Weakly-supervised multi-level attentional reconstruction network for grounding textual queries in videos. *ArXiv*, abs/2003.07048, 2020. 34

[156] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019. 34

[157] Shaoxiang Chen and Yu-Gang Jiang. Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8425–8435, 2021. 34

[158] Zhu Zhang, Zhijie Lin, Zhou Zhao, Jieming Zhu, and Xiuqiang He. Regularized two-branch proposal networks for weakly-supervised moment retrieval in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 4098–4106, 2020. 34

[159] Fan Luo, Shaoxiang Chen, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Self-supervised learning for semi-supervised temporal language grounding. *ArXiv*, abs/2109.11475, 2021. 34, 87, 98

[160] Jinwoo Nam, Daechul Ahn, Dongyeop Kang, Seong Jong Ha, and Jonghyun Choi. Zero-shot natural language video localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1470–1479, 2021. 34

[161] Junyu Gao and Changsheng Xu. Learning video moment retrieval without a single annotated video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 35

[162] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 35

[163] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision*, pages 510–526, 2016. 35

[164] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014. 36

[165] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 706–715, 2017. 36, 42, 95, 114

[166] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 36, 37, 114

[167] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. 37

[168] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. Script data for attribute-based recognition of composite activities. In *Proceedings of the European Conference on Computer Vision*, pages 144–157, 2012. 37

[169] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba Heilbron, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. *ArXiv*, abs/2112.00431, 2021. 37

[170] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 38

[171] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[172] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. 38

[173] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 39

[174] Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In *Proceedings of the 2nd International Workshop on Human-Centric Multimedia Analysis*, page 13–21, 2021. 39, 40, 87, 95, 97, 98

[175] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 189–198, 2017. 42

[176] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016. 43

[177] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. 44, 70, 95, 108

[178] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016. 44, 108

[179] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 44, 108, 115

[180] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. In *International Conference on Learning Representations*, 2017. 44

[181] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 49, 62, 77, 96

[182] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 49, 62, 77

[183] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, 2016. 70

[184] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018. 74, 110

[185] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000. 75

[186] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, volume 28, pages 3528–3536, 2015. 75

[187] Yoshua Bengio, N. Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013. 75

[188] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. 75, 76

[189] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. 75, 76

[190] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954. 75

[191] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27, pages 3086–3094, 2014. 75

[192] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 76

[193] Mayu Otani, Yuta Nakahima, Esa Rahtu, and Janne Heikkilä. Uncovering hidden challenges in query-based video moment retrieval. In *Proceedings of the British Machine Vision Conference*, 2020. 87, 92

[194] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. Deconfounded video moment retrieval with causal intervention. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1–10, 2021. 87, 98

[195] Remi Cadene, Corentin Dancette, Hedi Ben younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems*, volume 32, pages 839–850, 2019. 88, 89, 92

[196] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4069–4082, 2019. 88, 89, 92

[197] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *ArXiv*, abs/1907.12763, 2019. 103, 104, 114, 116, 118, 127

[198] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *The European Conference on Computer Vision*, 2020. 103, 104, 108, 109, 110, 114, 115, 116, 117, 118, 127

[199] Bowen Zhang, Hexiang Hu, Joonseok Lee, Ming Zhao, Sheide Chammas, Vihan Jain, Eugene Ie, and Fei Sha. A hierarchical multi-modal encoder for moment localization in video corpus. *ArXiv*, abs/2011.09046, 2020. 103, 104, 114, 115, 116, 117, 118, 127

[200] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1735–1742, 2006. 105

[201] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[202] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 105

[203] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. 105

[204] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000. 105

[205] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 110

[206] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019.

[207] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Proceedings of the European Conference on Computer Vision*, pages 214–229, 2020. 110

[208] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 297–304, 2010. 112

[209] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *ArXiv*, abs/1602.02410, 2016.

[210] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *ArXiv*, abs/1906.05743, 2019. 112

[211] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 112, 113

[212] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. 112

[213] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 115

[214] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 115

[215] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2046–2065, 2020. 116, 127

[216] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *ArXiv*, abs/1804.02516, 2018. 118

[217] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 125

[218] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. *ArXiv*, abs/2112.01529, 2021. 125

[219] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learningvia sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 125

[220] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *ArXiv*, abs/2109.14084, 2021. 125

[221] Jie Lei, Tamara L Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, pages 1–13, 2021. 125

[222] Arka Sadhu, Kan Chen, and Ram Nevatia. Video object grounding using semantic roles in language description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10417–10427, 2020. 126

[223] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli 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*, pages 10668–10677, 2020. 126

[224] Zhu Zhang, Zhou Zhao, Zhijie Lin, Baoxing Huai, and Jing Yuan. Object-aware multi-branch relation networks for spatio-temporal video grounding. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 1069–1075, 2020.

[225] Kai Shen, Lingfei Wu, Fangli Xu, Siliang Tang, Jun Xiao, and Yueting Zhuang. Hierarchical attention based spatial-temporal graph-to-sequence learning for grounded video description. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 941–947, 2020.

[226] Qianyu Feng, Yunchao Wei, Mingming Cheng, and Yi Yang. Decoupled spatial temporal graphs for generic visual grounding. *ArXiv*, abs/2103.10191, 2021.

[227] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 126

[228] Rui Su, Qian Yu, and Dong Xu. Stvgbert: A visual-linguistic transformer based framework for spatio-temporal video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1533–1542, 2021. 126

[229] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee Kenneth Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1884–1894, 2019. 126

[230] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5948–5957, 2018.

[231] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10444–10452, 2019.

[232] Junwen Chen, Wentao Bao, and Yu Kong. Activity-driven weakly-supervised spatio-temporal grounding from untrimmed videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 3789–3797, 2020.

[233] Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 126

[234] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision*, pages 247–263, 2018. 126

[235] Yu Wu, Linchao Zhu, Yan Yan, and Yi Yang. Dual attention matching for audio-visual event localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6292–6300, 2019.

[236] Haoming Xu, Runhao Zeng, Qingyao Wu, Mingkui Tan, and Chuang Gan. Cross-modal relation-aware networks for audio-visual event localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 3893–3901, 2020.

[237] Hanyu Xuan, Zhenyu Zhang, Shuo Chen, Jian Yang, and Yan Yan. Cross-modal attention network for temporal inconsistent audio-visual event localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 279–286, 2020.

[238] Bin Duan, Hao Tang, Wei Wang, Ziliang Zong, Guowei Yang, and Yan Yan. Audio-visual event localization via recursive fusion by joint co-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4013–4022, 2021.

[239] Hanyu Xuan, Lei Luo, Zhenyu Zhang, Jian Yang, and Yan Yan. Discriminative cross-modality attention network for temporal inconsistent audio-visual event localization. *IEEE Transactions on Image Processing*, 30:7878–7888, 2021.

[240] Cheng Xue, Xionghu Zhong, Minjie Cai, Hao Chen, and Wenwu Wang. Audio-visual event localization by learning spatial and semantic co-attention. *IEEE Transactions on Multimedia*, 2021. 126

[241] Noa Garcia and George Vogiatzis. Asymmetric spatio-temporal embeddings for large-scale image-to-video retrieval. In *Proceedings of the British Machine Vision Conference*, 2018. 126

[242] Zhu Zhang, Zhou Zhao, Zhijie Lin, Jingkuan Song, and Deng Cai. Localizing unseen activities in video via image query. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4390–4396, 2019.

[243] Ruicong Xu, Li Niu, Jianfu Zhang, and Liqing Zhang. A proposal-based approach for activity image-to-video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12524–12531, 2020.

[244] Liu Liu, Jiangtong Li, Li Niu, Ruicong Xu, and Liqing Zhang. Activity image-to-video retrieval by disentangling appearance and motion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2145–2153, 2021. 126

[245] Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. Video re-localization. In *Proceedings of the European Conference on Computer Vision*, pages 51–66, 2018. 126

[246] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Spatio-temporal video re-localization by warp lstm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1288–1297, 2019.

[247] Yung-Han Huang, Kuang-Jui Hsu, Shyh-Kang Jeng, and Yen-Yu Lin. Weakly-supervised video re-localization with multiscale attention model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11077–11084, 2020.

[248] Chen Jiang, Kaiming Huang, Sifeng He, Xudong Yang, Wei Zhang, Xiaobo Zhang, Yuan Cheng, Lei Yang, Qing Wang, Furong Xu, Tan Pan, and Wei Chu. Learning segment similarity and alignment in large-scale content based video retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 1618–1626, 2021. 126

[249] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 127

[250] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *National Academy of Sciences*, 117(14):7684–7689, 2020. 127

[251] Jie Lei, Tamara Berg, and Mohit Bansal. mTVR: Multilingual moment retrieval in videos. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 726–734, 2021. 127

[252] Sho Maeoki, Yusuke Mukuta, and Tatsuya Harada. Video moment retrieval with text query considering many-to-many correspondence using potentially relevant pair. *ArXiv*, abs/2106.13566, 2021. 127

[253] Sudipta Paul, Niluthpol Chowdhury Mithun, and Amit K. Roy-Chowdhury. Text-based localization of moments in a video corpus. *IEEE Transactions on Image Processing*, 30: 8886–8899, 2021.

[254] Zhijian Hou, Chong-Wah Ngo, and W. K. Chan. Conquer: Contextual query-aware ranking for video corpus moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 3900–3908, 2021. 127