Language-guided visual retrieval

He, Su

2021

He, S. (2021). Language-guided visual retrieval. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/151040>

<https://hdl.handle.net/10356/151040>

<https://doi.org/10.32657/10356/151040>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 18:34:21 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

LANGUAGE-GUIDED VISUAL RETRIEVAL

**HE SU
G1901225D**

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2021

LANGUAGE-GUIDED VISUAL RETRIEVAL

HE SU
G1901225D

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering (M.Eng)

2021

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.

16 / 01 / 2021

He Su

.....
Date

.....
HE SU

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.

18/1/2021

.....
Date

.....
Guosheng Lin

Authorship Attribution Statement

This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

16 / 01 / 2021

.....

Date

He Su

.....

HE SU

Acknowledgments

I would like to express my sincere gratitude to Prof. Guosheng Lin and Dr. Xiaofeng Yang for their guidance and help during my graduate career. Also, thanks for the support from my girlfriend, my family, and all the friends in NTU.

Table of Contents

Acknowledgments	6
Table of Contents	6
Abstract	I
List of Figures	II
List of Tables	IV
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Objectives	4
1.4 Organization of the Thesis	5
2 Literature Review	6
2.1 Common Techniques	6
2.1.1 Attention Mechanism	6
2.1.2 Graph Neural Network	8
2.1.3 Modular Network	9
2.2 Review of Visual Grounding	9
2.3 Review of Natural Language Video Localization	10
2.4 Chapter Summary	12
3 Visual Grounding	13
3.1 Introduction	13
3.2 The SVMN Model	15
3.2.1 Language Module	17

3.2.1.1	Language Pre-processing	17
3.2.1.2	Residual Attention Parser	17
3.2.2	Symbolic Module	19
3.2.3	Visual Modules	21
3.2.3.1	Subject Module	21
3.2.3.2	Location Module	21
3.2.3.3	Relation Module	22
3.2.4	Matching and Loss	23
3.3	Experiments	25
3.3.1	Datasets	25
3.3.2	Implementation Details	26
3.3.3	Quantitative results	26
3.3.4	Ablation experiment	28
3.3.4.1	Language Parser	28
3.3.4.2	Symbolic Module	28
3.3.4.3	Relation Module	29
3.3.5	Visualization	29
3.4	Chapter Summary	29
4	Natural Language Video Localization	32
4.1	Introduction	32
4.2	Model	36
4.2.1	Problem Formulation	37
4.2.2	Hybrid Graph Network	37
4.2.3	Temporal Encoder	41
4.2.4	Span Predictor	42
4.2.5	Effective Data Augmentation	43
4.3	Experiments	44
4.3.1	Datasets	44
4.3.2	Experimental settings	45
4.3.3	Comparison with the state-of-the-art	46
4.3.4	Ablation Experiments	47
4.3.5	Visualization	49

TABLE OF CONTENTS	9
4.4 Chapter Summary	49
5 Conclusion	51
References	52

Abstract

Language-guided Visual Retrieval (LGVR) is an important direction of cross-modality learning. It aims to retrieve or localize the objective message from the untrimmed visual information under the guidance of a linguistic description. In this thesis we study two popular sub-tasks of LGVR, one is Visual Grounding (VG) which aims to locate an object in the image, and the other is Natural Language Video Localization (NLVL) which aims to locate a targeted video clip from a long video span.

For VG, we propose a novel modular network learning to match both the object’s symbolic feature and visual feature extracted by CNN with the linguistic information to achieve a better cross-modality alignment. Besides, a residual attention parser is raised to leverage the difficulty of understanding language expressions.

For NLVL, we utilize the fine-grained semantic features of the sparse frames in the video. To organize the discrete features, we propose a network called Hybrid Graph Network to capture both the spatial and locally temporal relationships between objects in the frames and also apply semantically matching between objects and words. To model the long-span relationships between activities in the two modalities, we implement a temporal encoder based on the attentive model. Finally, we formulate the prediction as a binary classification task rather than regressing the specific boundaries.

We conduct extensive experiments on popular datasets on the two tasks to validate the effectiveness of our proposed models.

List of Figures

1.1	Illustration of the VG task.	3
1.2	Illustration of the NLVL task. Figure source: [1].	4
2.1	Illustration of the attention mechanism in NMT task. Figure source: [2].	7
2.2	Illustration of the multi-head attention used in [3]. Figure source: [3]. . .	8
3.1	Overall view of the SVMN model. The whole model mainly consists of three modules, namely the Residual Attention Parser, the hybrid visual module, and the matching module. The language parser decomposes an expression by reasoning operations into the subject, location, and relation phrases with their corresponding weights. The visual module fetches and organizes visual information also in three aspects. Finally, modular linguistic and visual features are matched by the multi-head matching module. Different modular features are represented in different colors. . .	16
3.2	Illustration of the computation of subject information by the Residual Attention language parser. The computation of the location and relation information is processed in the same way.	18
3.3	Illustration of the representations of visual and linguistic information. The left side is the showcases of the different representations for color, shape, and position information in visual space. And the right side is the showcase of the representation for all linguistic/symbolic information.	20
3.4	Illustration of relation network. The network aims to encode an object's relation features with its nearest neighbors.	23
3.5	Illustration of the Multi-head Matching Module. L2-N denotes L2 normalization applied for each feature before inner-product. This module aims to match visual and linguistic features in three components. Each component is composed of multiple channels' features. Cross-modality matching is first applied on each channel and then concluded into a final matching score.	24

3.6	Visualizations of some successful referring instances. The solid blue box is the ground-truth object and the red dashed box is the predicted one. Line A illustrates successful instances with phrases' attention distributions below. For the distribution, the darker the color is, the larger the attention value is, and vice versa. Line B and C provide more successful referring instances.	30
4.1	A showcase of annotations in ActivityNet. Figure source: [4].	33
4.2	Illustration of common scenarios in the video. (a) is a static scene that could be described with the spatial information purely. (b) is a movement description that involves the interactions of objects in different scenes. (c) demonstrates a scenario that highlights the importance of semantically objects-and-words matching.	34
4.3	The overall structure of the proposed HyGNet. The model includes three main sub-modules, namely the Hybrid Graph Network aiming to capture the spatial and local dependencies of scenes, the Temporal Encoder to model the temporal dependency and interact with the linguistic information, and the Span Predictor which localizes the target area with a single classifying loss.	36
4.4	Illustration of the structure of one layer's graph operation.	41
4.5	Illustration of some localization results. (a) to (c) are three successful instances with the iou larger than 0.99. While (d) is an unsatisfactory prediction, where the green bars indicates the scope of prediction.	50

List of Tables

3.1	Comparison with recent state-of-the-art approaches on three popular datasets RefCOCO, RefCOCO+, and RefCOCOg with ground-truth information (Acc%). We achieve state-of-the-art performance on all datasets, demonstrating the effectiveness of our model.	27
3.2	Comparison with state-of-the-art approaches on automatically detected proposals of Faster R-CNN (Acc%). SVMN also achieves state-of-the-art performance, which proves the robustness of our model.	27
3.3	Ablation studies of the main modules of SVMN (Acc%). ‘ <i>w/o</i> ’ represents replacing the module by a straightforward replacement or removing it directly. The results of erased models are compared with that of the full model and prove the effectiveness of the Residual Attention Parser, the symbolic module, and the relation network in the visual module.	28
4.1	Performance comparisons of our approach with existing state-of-the-art methods.	46
4.2	Ablation results of the layer composition analysis.	47
4.3	Comparison of results of sub-graphs ablation in Hybrid Graph Network. .	48
4.4	Results of ablation experiments of the feature analysis.	48

Chapter 1

Introduction

This chapter introduces the background of the task Language-Guided Visual Retrieval (LGVR), explains the motivation and objectives of the research, and also provides an overview of the organization of this thesis.

1.1 Background

In recent years, thanks to the liberation of computing power and the easy access of massive data on the Internet, the rapid development in Machine Learning has been comparable to the Cambrian Explosion in history. According to an OpenAI 2020 report¹, the algorithmic efficiency of Machine Learning doubles every 16 months, outpacing Moore’s Law. To be detailed, such explosive development mainly reflects in the field of Computer Vision (CV) and Natural Language Processing (NLP). For example, surpassing human-level results on the image classification task was just achieved in the year 2015; today, the autopilot system based on the Computer Vision algorithm has been realized to some extent. And machine translation and intelligent assistant systems from the NLP community have been providing service in our daily life for a long time. Machine Learning is showing its promising future in the coming technological revolution.

However, current achievements and applications in CV and NLP mainly lie in the mis-

¹<https://openai.com/blog/ai-and-efficiency/>

sions of recognition and classification. The extraordinary human intelligence is not only the perception and recognition of the surrounding environment through visual, auditory, gustatory, and tactile information but more importantly, the ability to converge various information to achieve the purpose of thinking and decision-making. Thereby, the realization of the human-like multi-modality interaction and collaboration capabilities is an important focus of the Machine Learning community because this is the basis to achieve true Artificial Intelligence. For the machine, visual and auditory information is the most common and useful modality of information, so the cross-modality learning studied in the community mainly refers to cross visual and linguistic information.

1.2 Motivation

Motivated by current industry needs and technical hot spots, this thesis takes research on Language-guided Visual Retrieval (**LGVR**) which is a concrete direction under cross-modality learning. LGVR aims to localize target information from untrimmed visual information under the guidance of a language description. Similar scenes are so commonly seen in human intelligence when thinking and making decisions that we even ignore its importance and sophistication. Nevertheless, when we want to achieve more intelligent and interactive Artificial Intelligence, this research becomes crucial. Compared with common vision and language tasks, LGVR not only needs to comprehend the visual and linguistic information respectively but also requires to find the association between the two modality information. The former has been adequately studied with the help of Convolution Neural Network (CNN) and Sequence Network. Therefore, the focus of LGVR mainly lies on the latter part, i.e. how to associate two types of information, which is also the emphasis of this thesis.

It is not easy to sweep this field completely because it contains a lot of content. Based on the difference in visual representation, LGVR can be divided into Image Retrieval and Video Retrieval which both have several detailed sub-tasks for different retrieving purposes. For example, for Image Retrieval, popular tasks include Visual Grounding, localizing a target object in one image matching the language description, and Text

Language query: A man wearing a blue jacket holding a controller on the right.

Figure 1.1: Illustration of the VG task.

Image Retrieval, retrieving a target image from the database according to the language; for the Video Retrieval, main tasks include Natural Language Video Localization aiming to localize a video clip from an untrimmed video based on the language, Video Retrieval, retrieving a video from a database with many videos according to a description, and Video Object Referring, referring an object in the frames matching with the description. However, although the purpose of each task is different, they do have some common requirements, that is, to extract the information of the two modalities using the existing technology, and to fuse the two kinds of information. Due to this coverage and similarity, this thesis delves into two sub-tasks namely Visual Grounding under Image Retrieval and Natural Language Video Localization under Video Retrieval.

To have an intact definition of the two tasks, Visual Grounding (denoted as **VG** in following), also called Image Referring and Referring Expression Comprehension, is a task of locating a specific target object in an image according to the description of a given language expression. A graphic example is shown in Fig. 1.1. Expressions in VG come from life or imitate human expressions, which are highly various and trivial. For example, the selected man in Fig. 1.1 can be described either simply as “the man”, “man in blue” or complicatedly as “a man wearing a blue jacket holding a controller standing on the right of

Figure 1.2: Illustration of the NLVL task. Figure source: [1].

a woman” which includes the relationship descriptions with surroundings. This could be under wide applications in the real world like the assistant robot and autopilot systems. Illustrated as Fig. 1.2, Natural Language Video Localization (denoted as **NLVL**) aims to retrieve or localize a temporal moment in an untrimmed video semantically corresponding to a given language expression. These expressions could be descriptions of a simple action like ”opening the door”, ”dancing”, or a series of coherent actions, or a scene like ”a group of people playing basketball”. This will be of great help in the search engine or recommendation system with a huge amount of videos.

1.3 Objectives

The objectives of this thesis are two-fold. For the first one, we aim to conduct thorough research on the aforementioned topics, including versatile and useful techniques and popular model designs. The purpose of this is to have a basic knowledge and even a high-level view of the current cross-modality learning progress. Founded on the achievements of previous work, the second objective is to design models with our novelty that can achieve competitive or even surpassing the state-of-the-art (SOTA) performance.

1.4 Organization of the Thesis

The thesis is organized as follows:

1. Chapter 1 presents the background, motivation, and objectives of this thesis.
2. Chapter 2 reviews previous work and analyzes the pros and cons.
3. Chapter 3 shows our work about Visual Grounding.
4. Chapter 4 demonstrates our model of Natural Language Video Localization.
5. Chapter 5 makes conclusions.

Chapter 2

Literature Review

This chapter first reviews some common functional techniques in cross-modality learning. Then the development and recent work about Visual Grounding and Natural Language Video Localization is given respectively.

2.1 Common Techniques

2.1.1 Attention Mechanism

The attention mechanism is an important technique that is widely used in both CV and NLP. It was inspired by humans paying different attention to surrounding information and first introduced in [2] to help memorize the information in long sentences for neural machine translation (NMT). To generate an output state, the decoder of the NMT model considers both the previous output state and the weighted combination of all input information, illustrated as Fig. 2.1.1. Weighted score or alignment score here is computed by comparing the difference between the key information (here is the optional output) and the query information (here is the input information), and the process of computing is the attention mechanism. Taking into account the improvement in NLP, it was soon extended into CV [5]. Through this biased selection of information, the attention mechanism can effectively help increase the complexity and compatibility of the model, and it is also a

Figure 2.1: Illustration of the attention mechanism in NMT task. Figure source: [2].

practical manner to associate two types of information.

Based on the origination of query information, attention can be summarized as self-attention and cross-attention. Self-attention, also called intra-attention, computes the alignment score with information from the same sequence but on different positions. And cross-attention, in contrast, is with the query information from other sources. As for the mechanism of computing the attention score, there are various operations explored in [2, 3, 6, 7], e.g.,

$$score = \sigma(W[s_t; h_i]), \quad (2.1a)$$

$$score = s_t^T W h_i, \quad (2.1b)$$

$$score = s_t^T h_i, \quad (2.1c)$$

$$score = \frac{s_t^T h_i}{\sqrt{n}}, \quad (2.1d)$$

where W is trainable parameters and σ denotes activation function, and s_t and h_i denote key information and query information respectively.

The great master of the attention mechanism is the “Attention is All you Need” [3], which exploits the multi-head scaled dot-product attention (illustrated in Fig. 2.1.1) to capture the dependencies between words in the sentence. Due to its high parallelization and effectiveness, it has been widely used in NLP [3, 8] and cross-modality learning [9, 10]

Figure 2.2: Illustration of the multi-head attention used in [3]. Figure source: [3].

recently and is gradually taking place of traditional sequence models in NLP and even CNN model in CV [11].

2.1.2 Graph Neural Network

Graph Neural Network (GNN) [12] has shown its advantage in processing irregular information in various domains including social networks, biological networks, and molecular architecture. In this thesis, we adopt it to model the relationships between semantic objects in the video sequence. At a high level, similar to Convolutional Neural Network (CNN), GNN also aims to build a comprehensive representation of the input data in the context. A simple update layer l of the GNN could be represented as

$$X^{l+1} = \sigma \left((A + I)X^l\theta^l \right), \quad (2.2)$$

where X^l is the feature vectors of nodes, A is the adjacent matrix indicating the connection of nodes and I means the self-loops of themselves. θ^l is a learnable parameter in layer l . σ is the activation function. During operations, $(A + I)$ is usually normalized.

Based on the sources of the adjacent matrix A , there are different variants of GNN. A vanilla version is all ones, which mean every node in the graph is connected, i.e. a fully connected graph. For some data with prior knowledge, the adjacent matrix would

be acquired. In most cases, A is computed using various mechanisms, e.g. attention mechanism (GAN) [13] or gating mechanism [14]. The operation could be reformulated as

$$h_i^{l+1} = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} h_j^l \right), \quad (2.3)$$

where $\mathcal{N}_i$ are the neighbors of one node i and h_j^l is the feature of its neighbor j . The collection of h_j^l corresponds to X^l in Eq. 2.2. α_{ij} is a computed linkage score of the neighboring node j to i , and $\mathbf{W}$ is the learnable parameter. σ is the activation function.

2.1.3 Modular Network

Modular network was first used in the control system a long time ago. With the development of the neural network, network architecture is becoming more and more complicated, and thereby modular network is introduced to improve the versatility and interpretability of the model. [15] first exploits neural module networks in Visual Question Answering (VQA). In VQA, neural module networks are dynamically constructed with different reusable-and-compositional candidate modules based on the questions' content. Candidate modules are jointly trained across all constructed structures during each inference. After that, more and more advancement of the neural module network is achieved including [16–20]. [19] first introduces the neural module network into the VG task, which uses a fixed structure of the compositional modular network with the components of subject, relationship, and object. [21] further extends the compositional modular network into (subject, location, and relation) triples, which is more inclined to human's speaking habits: firstly what the object is/looks like, then where it is, lastly you may find it above/under/next to another object.

2.2 Review of Visual Grounding

The task of Visual Grounding aims to locate a corresponding region of an image by the instruction of a language expression. The key point of this task is how to detect the relation between the language expression and the image, and transform this relation into

an indication in the image. A common idea is to adapt the encoder-decoder structure in semantic segmentation by introducing the instruction of linguistic features, e.g., [22–27]. Since these approaches use image features, segmenting masks could be directly generated end to end. However, because of the complexity and vagueness of the overall image and language features, the performance of this encoder-decoder approach is limited. To obtain more accurate pixel-prediction, some post-processing approaches, e.g., building graph structure to capture context information [28, 29], training model in a generative adversarial fashion [30], are explored.

Another idea is to first generate proposals from the image by existing CNN models like faster R-CNN [31], then locate the proposal satisfying the description of expression with the detailed proposals’ feature rather than the whole image’s feature. [19, 21, 32–35] are the representatives of this two-stage approach. Hu et al. [19] first exploits modular networks (subject, relation, object) to decompose a proposal’s visual features in different aspects to match with the corresponding expression phrases. Yu et al. [21] further improves modular networks into subject, location, and relation partitions to better utilize features. Rather than matching visual features with linguistic features in different aspects, Yang et al. [34, 35] suggest to construct a scene graph for all proposals and then mix and refine original features in multiple reasoning steps to compute a comprehensive representation for matching, which has a good performance on complex relation instances. These methods normally take ground-truth or detected proposals for referring. Since the detection technique is already matured and the aforementioned methods all achieved outstanding performance, we believe that referring based on proposals and segmenting after proposal localization is more suitable for this task.

2.3 Review of Natural Language Video Localization

Natural Language Video Localization (NLVL) is a typical and popular video-text task that aims to localize a temporal moment semantically matching a given expression in the video. This task was first introduced in [36, 37]. Similar to the practice of traditional object detection in 2D images, intuitive methods for NLVL also have two styles, i.e.

anchor-based and anchor-free. Anchor-based methods including [37–45] treat NLVL as a ranking task, which divides the untrimmed video into many candidate clips and composes the visual feature of each clip through different methods. A simple way of this is to generate consecutive clips through a sliding window and compose features via the pooling technique, e.g., [37–39, 43]. To better produce proper candidates, [44, 45] utilize a sparse sampling strategy to generate candidate moments, and [41] introduces an advanced query-based attentive approach to generate clip proposals. Recently, to capture the temporal dependencies among candidate moments/clips, [43, 45] model the relational information of moments through the graph network and [44] models all video moments in different start and end times by a two-dimensional map and applies convolution in the map. Each clip’s feature is then processed with the linguistic feature to predict an alignment score and regress distances to the real boundaries. The final prediction is the clip with the largest alignment score after the adjustment of boundaries. However, since the length and location of the target span have a large variance, generating a large number of candidate clips in different locations and scales is usually required, which incurs high computation and memory costs.

To overcome the drawback of the anchor-based method, some anchor-free schemes are proposed. These methods [1, 46–50] usually organize a video as a continuous sequence holistically and process it with the sequence network like [8, 51] to capture the temporal dependency. Then the interaction between the visual and linguistic sequences is usually applied in different attention operations. The processed sequence is used to predict the start and end boundaries directly. For example, [46, 47] first perform a binary classification for each frame in the sequence, then densely regress the distances from boundaries for all positive frames. And [1, 48–50] directly predict three probability scores for each frame being the foreground annotation, the start, and end boundaries with two kinds of classification losses. These methods greatly reduce the redundant computation bringing by candidate clips, and at the same time make good use of the temporal dependence in the video, which achieves appealing performance.

However, the aforementioned approaches normally exploit the 3D-Conv network to extract the visual feature but overlook the importance of understanding relationships of semantic objects in the scene. Recently, [52] proposes obtaining contextualized activity

representations in the video based on a language-conditioned message-passing algorithm where the activities are the composition of interactions between humans and objects in the scene. [53] also utilizes graph networks to model interactions between the objects and words as well as among the objects. These two jobs are similar to our model as they also utilize semantic features and devise graph operation in the network, but the organization and composition of their graphs are different from ours and they also fail to recognize the correspondence of objects in the time domain.

2.4 Chapter Summary

In this chapter, we first introduce the Attention Mechanism, which is an important biased learning technique in both CV and NLP. GNN is conducted in our NLVL network to model the relationship in the video and language. And we also review the development and important work of VG and NLVL respectively.

Chapter 3

Visual Grounding

In this section, we first introduce the current progress of the VG task and analyze the pros and cons of previous work, then give our proposed model and describe each module in detail. Dense experiments are conducted on three popular datasets to validate the effectiveness of the proposed model.

3.1 Introduction

Visual Grounding (VG) [19, 54, 55], also called Referring Expression Grounding, Referring Expression Comprehension, and Image Referring is a task of locating a specific region in an image according to the instruction of a linguistic expression. Though tasks in Computer Vision (CV) such as image classification [56, 57], object detection [31], and techniques in Natural Language Processing (NLP) develop rapidly in recent years, Visual Grounding, similar to Visual Question Answering (VQA) [58, 59] and Image Captioning [60], is still challenging for the difficulty in cross-modality processing. The information in images and languages is organized in different fashions that the former is high order continuous information organized in a grid, while the latter is represented as sequential or tree-structured labels. The difference in organizing structure sets up obstacles when seeking the connection of two modality information.

To tackle the challenge of cross-modality mapping, several methods are proposed in

previous work. One practice is to decompose the intricate linguistic and visual features into different pieces and seek the connection of each piece. For example, [19] decomposes information into (subject, relation, object) tuple to handle the relationships, and [21] proposes a new partition, i.e. (subject, location, and relation) which covers more comprehensive information. Another beneficial technique is the attention mechanism, which can be used to select relevant information from the untrimmed representation. For example, [61] attends overall image feature, local object feature, and expression feature mutually and circularly to extract related information, and then seeks their correlation. Besides, inspired by the work in VQA, reasoning network provides another good solution for this task, e.g., [35] tries to build a scene graph to represent visual information and then attends the feature by expression information in the graph. These methods have an advantage in analyzing complicated scene relations.

Besides, referring expressions of VG come from life, so they are highly various and trivial. For example, the selected man in Fig. 3.1 can be described either simply as “the man” or complicatedly as “a man wearing a blue jacket holding a controller standing on the right of a woman”. To solve the problem of language understanding, off-the-shelf language parsers were first introduced to process diverse expressions by their syntax rules [54,62], but the performance on oral expression is far from satisfactory. Then [19,21] learn to parse expressions automatically through the self-attention scheme, which is more light-weighted and also achieves appealing performance.

To address the difficulty in cross-modality mapping, we propose a module called Symbolic Module to apply the language-to-symbol mapping. Pure vision features are noisy and continuous. Given a region’s feature, the background information and intrusive objects’ information are also incorporated in the feature space. Information like color, shape, and position is expressed continuously in visual space, which is different from discrete linguistic information. Symbolic representation, on the other hand, is tidy, with noisy information removed and discretized into labels as the representation of language. Besides, to fully utilize the rich information of visual features, we also introduce a visual module to work in parallel with the symbolic module. The design of the visual module is based on the implementation in [21], which decomposes information into subject, location, and relation.

To settle the problem of language understanding, we propose a novel Residual Attention Parser to analyze different components and their weights in an expression. Residual Attention Parser computes a componential distribution with the expression’s context information and the residual context information of other components. And the weight of this component is computed by comparing the similarity between the expression’s overall meaning and the componential meaning.

The whole network is called Symbolic Vision Modular Network (**SVMN**) which is composed of four components, namely the Residual Attention Parser, the visual module, the symbolic module, and the matching module, illustrated as Fig. 3.1. Experiments on three popular datasets demonstrate that our model achieves state-of-the-art performance and ablation experiments for each module prove their effectiveness.

To summarize our contributions:

- We propose a new model learning to map language and vision in both the cross-modality manner (language to vision) and the same-modality manner (language to symbol).
- We propose a Residual Attention Parser to process variable expressions.
- We achieve state-of-the-art performance on three popular datasets.

3.2 The SVMN Model

Illustrated as Fig. 3.1, the SVMN model consists of the Residual Attention parser, a hybrid visual module including the proposed symbolic module and the conventional visual module, and a multi-head matching module. The language module is designed to parse an expression into different phrasal representations and their corresponding weights in the whole expression through the operation named residual attention. To process the visual information, we employ the symbolic module to fetch the symbolic features of the objects. Meanwhile, we also implement a conventional module in parallel to fetch objects’ visual features in three aspects, namely object, location, and relation. With the componential features of language and vision, we propose a multi-head matching module to seek the connection in different aspects.

Figure 3.1: Overall view of the SVMN model. The whole model mainly consists of three modules, namely the Residual Attention Parser, the hybrid visual module, and the matching module. The language parser decomposes an expression by reasoning operations into the subject, location, and relation phrases with their corresponding weights. The visual module fetches and organizes visual information also in three aspects. Finally, modular linguistic and visual features are matched by the multi-head matching module. Different modular features are represented in different colors.

Visual Grounding is generally formed into a ranking task that selecting the best-matching proposal from a set of candidates $O = \{o_i\}_{i=1}^N$ in an image I , according to the content of a given expression/query $Q = \{w_m\}_{m=1}^M$. The candidates are usually provided as annotations or could be extracted by off-the-shelf extractors like Faster R-CNN [31]. This ranking process can be formulated as $\operatorname{argmax}\{s(o_i, Q)\}_{i=1}^N$, where s indicates the matching score of an object and the query. The focus of this paper mainly lies on how to compute a proper cross-modality matching score s .

3.2.1 Language Module

3.2.1.1 Language Pre-processing

For a given referential expression Q with M words $\{w_m\}_{m=1}^M$, we first use GloVe [63] to encode each word into an embedding vector $E = \{e_m\}_{m=1}^M$. Then we encode word embedding by a one-layer bi-directional LSTM network. We fetch two parts of the LSTM's output, one is the concatenation of bi-directional hidden states of every word $C = \{c_m = [\vec{h}_t, \overleftarrow{h}_t]\}_{m=1}^M$ which represents the contextual meaning of every word in the expression, the other is the concatenation of bi-directional final hidden states $q = [c_1, c_M]$ which represents the overall meaning of the expression. E , C , and q will all be used in the language parser.

3.2.1.2 Residual Attention Parser

We design a Residual Attention language parser to decompose an expression into three phrases, corresponding to subject, location, and relation information respectively. Subject is the description of the targeted object's category and attributes; location information is about the position in the image; relation information refers to the relationships with its neighbors. The intention to decompose into these three aspects is inclined with humans' habits of referring to an object that firstly what the object is and looks like, then where it is, finally what is the relationship with another object if direct instruction is not clear. To parse the components of an expression, the method in [19], [21] exploits three independent fully connected layers in charge of computing the attentive distribution of

Figure 3.2: Illustration of the computation of subject information by the Residual Attention language parser. The computation of the location and relation information is processed in the same way.

each phrase respectively. However, we argue that different phrasal components in one language expression should be mutually exclusive, i.e. a word with the high attention for one phrase should not be highlighted in the two others. In light of this, we introduce the attentive vectors of any other phrases as a clue to help compute the attentive vector of one phrase. The process of computing the subject phrase is illustrated in Fig. 3.2. The computation of the location and relation parts are operated in the same manner.

The inputs for the parser are context C , final hidden state q , and word embedding E . We exploit three Multi-Layer Perceptron (MLP) denoted as MLP_r , $r \in \{sub, loc, rel\}$ to compute the attention vector for each phrase. To obtain the residual attention vector a_{res} , we first sum up the attention vectors of two other phrases, then clamp it into 0 and 1, and finally subtract it by 1, formulated as Eq. 3.1a. Obtained residual attention a_{res} is then multiplied with context C to get the residual context information which will be concatenated with the original context C . The feature concatenation is used to compute the phrasal attention in MLP_r . These processes are formulated as Eq. 3.1a, 3.1b. It needs to be noted that the attention computing operation in three MLP_r would be applied twice to capture the residual information. During the first computation, the initial attentive vectors a_r are padded by zeros.

With the attention distribution a_r for each phrase, word embedding E is then weighted summed to generate the phrase embedding p_r (Eq. 3.1c). Similarly, a sub-context feature C_r will also be computed with the context C for each phrase r (Eq. 3.1d). Sub-context C_r

is then compared with the overall meaning q to compute a weight w_r which indicates how much the phrase r contributes to this whole expression (Eq. 3.1e). The final outputs of the whole module are three phrase representation p_r , $r \in \{sub, loc, rel\}$ and their corresponding normalized weights w_r , $r \in \{sub, loc, rel\}$ (Eq. 3.1f). Main operations above are formulated as following, among which the subscript r represents a phrase from $R = \{sub, loc, rel\}$, and $R - r$ means the phrases in R other than r .

$$\begin{aligned} a_{res} &= 1 - clamp_0^1(sum(att_{R-r})), \\ r &\in R = \{sub, loc, rel\} \end{aligned} \quad (3.1a)$$

$$a_r = MLP_r(context, a_{res} \cdot context) \quad (3.1b)$$

$$p_r = \sum_{m=1}^M a_{r,m} e_m \quad (3.1c)$$

$$C_r = \sum_{m=1}^M a_{r,m} c_m \quad (3.1d)$$

$$w_r = MLP_w(q, C_r) \quad (3.1e)$$

$$w_r = \frac{exp(w_r)}{\sum_{k \in R} exp(w_k)} \quad (3.1f)$$

3.2.2 Symbolic Module

As mentioning before, visual and linguistic information is organized in different styles. Visual information including color, shape, and position is expressed differently and continuously in images. Illustrated in Fig. 3.3, in visual space (the left column), the color information is represented as continuously changing pixel-value distribution (first line “color”); shape information is represented as a set of pixel-clusters (second line “shape”); position information is represented by a set of pixel-cluster-coordinate-offsets (third line “position”). However, the information which has different and continuous representations in visual space is all expressed as discrete labels in language space (the right column). To handle this discrepancy, our idea is to first transform the changeable and continuous

Figure 3.3: Illustration of the representations of visual and linguistic information. The left side is the showcases of the different representations for color, shape, and position information in visual space. And the right side is the showcase of the representation for all linguistic/symbolic information.

visual feature into the symbolic feature by strongly supervised networks like Faster R-CNN [31]. To be specific, proposals in images can be expressed as a group of symbols including object category, attribute, position, and even relation labels. Then the symbolic labels could be encoded with the same embedding network as that of the referring expressions. Since the embedding vectors of semantically similar words are closed to each other in representing space, for instance, the word 'person', 'guy', and 'adult' have close embedding vectors, this can also solve the problem caused by expression uncertainty. We call this process as the language-to-symbol mapping.

Due to the absence of other ground-truth labels, we design a symbolic network with only category information. During experiments, both the ground-truth annotation and predicted symbolic information are tested. The symbolic labels are embedded by the same embedding network for language embedding. The embedded symbolic feature is denoted as v^{sym} . It will be matched with the phrasal embedding of subject p_{sub} which contains the category description to obtain a symbolic score. The matching module containing several trainable aligning layers will be discussed later. To leverage the inaccuracy caused by the detector when using predicted symbolic labels, the symbolic score is weighted by the confidence score of the symbolic label given by the detector.

3.2.3 Visual Modules

To comprehensively exploit visual information, we also implement three conventional visual modules based on the structure of [21]. These modules fetch an object’s visual feature in three aspects, namely subject, location/position, and relation, which is in agreement with the operation of language. Compared with the structure in [21], our unique design mainly lies on the relation part. And the position feature processing also has some refinement.

3.2.3.1 Subject Module

The subject module aims to extract an object’s subject features, including the category and attribute information. Given a candidate object i in an image, we extract the $C_3 \in \mathbb{R}^{1024 \times 7 \times 7}$ and $C_4 \in \mathbb{R}^{2048 \times 7 \times 7}$ feature from Faster R-CNN [31]. Then we follow the operation in [21] to extract the object’s subject information from C3 and C4 features. The operation can be formulated as the following steps:

$$V = MLP_f(C_3, C_4), \quad (3.2a)$$

$$A = softmax(MLP_a([V, C_4], p_{sub})), \quad (3.2b)$$

$$v_i^{sub} = \sum A \cdot [V, C_4]. \quad (3.2c)$$

In the equations, MLP_f is a one-layer MLP with $\text{ReLU}()$ as the activation function and MLP_a is a two-layer MLP with $\text{Tanh}()$ as activation functions. Eq. 3.2b aims to compute an attention map over the grid, and Eq. 3.2c is a phrase-guided attention pooling operation [21]. The final subject feature of object i is $v_i^{sub} \in \mathbb{R}^{512+2048}$.

3.2.3.2 Location Module

An object’s location information is composed of two parts, one is the absolute location information including words like ‘left’, ‘right’, ‘top’, ‘down’ which describe a specific position in the image, the other is the relative location information including descriptions like “the third bottle” which indicates the relative location in all same type objects. For ob-

ject i , its raw location feature is represented as $v_i^{loc} = [v_i^{abs.loc}, v_i^{rel.loc}]$. Following preceding methods [21, 64, 65], absolute location is fetched as $v_i^{abs.loc} = [\frac{x_i^{tl}}{w_I}, \frac{y_i^{tl}}{h_I}, \frac{x_i^{br}}{w_I}, \frac{y_i^{br}}{h_I}, \frac{w_i h_i}{w_I h_I}]$, among which w_I, h_I denotes the image's width and height and tl, br denote top-left, bottom-right respectively. Relative location is fetched as its relative coordinates to its up-to-five nearest same-category neighbors $v_i^{rel.loc} = 5 \times [\frac{(x_i^{tl} - x_j^{tl})}{w_I}, \frac{(y_i^{tl} - y_j^{tl})}{h_I}, \frac{(x_i^{br} - x_j^{br})}{w_I}, \frac{(y_i^{br} - y_j^{br})}{h_I}, \frac{w_j h_j}{w_I h_I}]$, where j denotes the index of its neighbor.

Rather than fusing two locations into a united representation, we make use of them separately for cross-modality matching. The raw location feature is processed in the following steps:

$$v_i^{abs.loc} = MLP_{abs}(v_i^{abs.loc}), \quad (3.3a)$$

$$v_i^{rel.loc} = MLP_{rel}(v_i^{rel.loc}), \quad (3.3b)$$

$$v_i^{rel.loc} = MLP_{fuse}(v_i^{rel.loc}). \quad (3.3c)$$

MLP_{abs} , MLP_{rel} , and MLP_{fuse} are all one-layer MLP with $\text{ReLU}()$ as the activation function. Eq. 3.3c aims to fuse five channels' relative relation into one representation. This final location feature for the object i is $v_i^{loc} = (v_i^{abs.loc}, v_i^{rel.loc}) \in \mathbb{R}^{2 \times 512}$.

3.2.3.3 Relation Module

To parse objects' relationships in an image, constructing a scene graph is a common practice. However, it will require extra knowledge since referring datasets do not contain the necessary information to build a scene graph. On the other hand, language expressions in the VG task are normally quite simple and do not contain complex multiple relations like VQA. In this case, we design a simple relation module to learn objects' relationships based on their visual feature and location information inspired by [66].

The module structure is illustrated in Fig. 3.4. For object i , we fetch up to five of the nearest objects as its neighborhood since people tend to pick nearby objects as references. We use C_3 and C_4 features in Section 3.2.3.1 after averaged pooling as the object feature, for both the target object v_i^{obj} , and its neighboring objects v_j^{obj} . Besides, the relative location feature v_{ij}^{loc} is calculated the same as $v_i^{rel.loc}$ in Section 3.2.3.2. With the objects' feature v_i^{obj} , v_j^{obj} , and their relative location v_{ij}^{loc} , we compute their relation

Figure 3.4: Illustration of relation network. The network aims to encode an object’s relation features with its nearest neighbors.

representation in the following steps:

$$v_i^{obj} = MLP_{obj}(v_i^{obj}), \quad (3.4a)$$

$$v_j^{obj} = MLP_{obj}(v_j^{obj}), \quad (3.4b)$$

$$v_{ij}^{loc} = MLP_{loc}(v_{ij}^{loc}), \quad (3.4c)$$

$$v_{ij}^{rel} = f(v_i^{obj}, v_{ij}^{loc}, v_j^{obj}). \quad (3.4d)$$

MLP_{obj} is a two-layer MLP using $\text{ReLU}()$ as the activation function and MLP_{loc} is a one-layer MLP with $\text{ReLU}()$. f is a composite function learning to infer relations. We use a two-layer MLP during experiments. The final relational feature of object i with its 5 neighbors is $v_i^{rel} = 5 \times (v_{ij}^{rel}) \in \mathbb{R}^{5 \times 512}$.

3.2.4 Matching and Loss

To handle the componential visual and linguistic features mapping, we propose a multi-head matching module, illustrated as Fig.3.5. The module is composed of three stages. In the first stage, the componential visual and linguistic features are matched respectively in

Figure 3.5: Illustration of the Multi-head Matching Module. L2-N denotes L2 normalization applied for each feature before inner-product. This module aims to match visual and linguistic features in three components. Each component is composed of multiple channels’ features. Cross-modality matching is first applied on each channel and then concluded into a final matching score.

multiple channels. Since the symbolic feature contains the category information, which corresponds to the description of the subject, we add it as an additional subjective channel for matching. Besides, location and relation features have 2 and 5 channels respectively. Each channel’s visual feature will be matched with their corresponding language phrases p_{sub} , p_{loc} , and p_{rel} respectively. For example, to match the subjective information, the symbolic feature v^{sym} and the visual subject feature v^{sub} will be matched with the subject phrase p_{sub} respectively, and will get two channels’ matching scores. A single matching function is illustrated in the top-right corner of Fig. 3.5, among which each feature will first be processed by a 2-layer MLP with $\text{ReLU}()$ and batch normalization after each layer. Then the similarity score is computed as the inner-product of two normalized features (L2 normalization).

In the second stage, multi-channel matching score of each component will be concluded by summing up ($\oplus$) or summing up the top-2 scores (*top2-sum*). Finally, three componential matching scores will be weighted by their corresponding phrasal weights w_{sub} , w_{loc} , and w_{rel} respectively and then summed up to get the final matching score which indicates the matching degree of the object and language.

For training, we exploit the triple hinge loss in Eq. 3.5. For each unit pair that has

an object and its corresponding expression, we randomly sample a negative expression that does not describe the object and a negative object which is not corresponding to the positive expression from the same image. The data is assembled into three pairs, namely (positive object, positive expression), (positive object, negative expression), and (negative object, positive expression). Each pair will be computed for a matching score, namely s_{pos} , s_{neg}^{exp} , and s_{neg}^{obj} . The loss is organized as:

$$loss = \max(s_{neg}^{obj} + \delta - s_{pos}, 0) + \max(s_{neg}^{exp} + \delta - s_{pos}, 0), \quad (3.5)$$

where δ is the margin for negative and positive scores and is set to 0.1 during experiments.

3.3 Experiments

In this section, we first introduce the dataset in Section 3.3.1, then analyse experiments results in Section 3.3.3, 3.3.4. Some visualizations are given in Section 3.3.5.

3.3.1 Datasets

We test our model on three mainstream datasets, namely RefCOCO, RefCOCO+ [64], and RefCOCOg [55]. All experiments are conducted under the split of images rather than objects (denoting as val^* in Table. 3.3.2), which is more inclined with common practice. The images of the three datasets all come from MSCOCO’s training images [67]. The difference is that RefCOCO and RefCOCO+ are collected in an interactive game interface that provides wild expressions, while expressions of RefCOCOg are generated by a designed model. So the averaged expression length of RefCOCO and RefCOCO+ (3.5) is shorter than that of RefCOCOg (8.4). The difference between RefCOCO and RefCOCO+ is that the latter does not contain absolute location words like ‘right’, ‘top’, which enhances the difficulty of referring. Finally, testA splits of RefCOCO and RefCOCO+ are people-targeted, while testB splits contain other category objects. This is for different evaluating purposes.

3.3.2 Implementation Details

During experiments, following previous practices, we train our model with datasets' ground-truth information and test it with both ground-truth and detected information. Since this model mainly aims to handle the alignment problem of cross-modality, we only compare and analyze the performance of localization on ground-truth proposals during ablation experiments. We exploit the Faster-RCNN (ResNet101 as the backbone) model provided by [21] to extract proposals and visual features. The Faster-RCNN was pretrained on MSCOCO's training images excluding those in the validation and test split of three datasets. For evaluation, all proposals in an image will be matched with the expression respectively, and the one with the highest matching score is picked as the target. The inference is counted as successful if the intersection-over-union (IoU) between the predicted box and the target one is larger than 0.5. We compute the accuracy of successful inference counts over all inference counts as the evaluating metric (Acc%).

Apart from the off-the-shelf feature extractor and word embedding network, the whole network is trained end-to-end from scratch. The model is trained for 30 epochs with a batch size of 20 images and optimized by Adam with an initial learning rate of 0.004, gradient clipping of 1.0, and the first 8000-iteration warm-up. Dropout with the rate of 0.4 and weight decaying rate of 0.01 is applied to prevent over-fitting. The word embedding size, the hidden state size of the LSTM, and all feature vector sizes are set to 512 within the model.

3.3.3 Quantitative results

The results of our **SVMN** model on ground-truth evaluation and the comparison of performance to recent state-of-the-art models are shown in Table. 3.3.2. And the results on automatically detected proposals are given in Table. 3.3.3. In general, our **SVMN** outperforms previous models by a large margin in all data splits in two evaluating settings. In all comparing models, MAttNet [21] and LGRAN [33] also make use of three modular networks. Performance gain over them proves the effectiveness of our modular designs. Horizontally viewing, compared with the former state-of-the-art, the performance gains

Model	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val*	val	test
MMI [55]	-	71.72	71.09	-	58.42	51.23	62.14	-	-
visdif+MMI [64]	-	73.98	76.59	-	59.17	55.62	64.02	-	-
NegBag [68]	76.90	75.60	78.00	-	-	-	-	-	68.40
CMN [19]	-	75.94	79.57	-	59.29	59.34	69.3	-	-
S.L.R [65]	78.36	77.97	79.86	61.33	63.1	58.19	72.02	71.32	71.72
VC [69]	-	78.98	82.39	-	62.56	62.90	73.98	-	-
Accu-Att [61]	81.27	81.17	80.01	65.56	68.76	60.63	73.18	-	-
PLAN [70]	81.67	80.81	81.32	64.18	66.31	61.46	69.47	-	-
LGRAN [33]	82.0	81.2	84.0	66.6	67.6	65.5	-	75.4	74.7
MAttNet [21]	85.65	85.26	84.57	71.01	75.13	66.17	-	78.10	78.12
SVMN	87.73	87.01	87.34	73.39	75.69	70.26	-	80.07	79.79

Table 3.1: Comparison with recent state-of-the-art approaches on three popular datasets RefCOCO, RefCOCO+, and RefCOCOg with ground-truth information (Acc%). We achieve state-of-the-art performance on all datasets, demonstrating the effectiveness of our model.

Model	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val*	val	test
MMI [55]	-	64.90	54.51	-	54.03	42.81	45.85	-	-
Neg Bag [68]	-	58.60	56.40	-	-	-	-	-	49.50
CMN [19]	-	71.03	65.77	-	54.32	47.76	57.47	-	-
VC [69]	-	73.33	67.44	-	58.40	53.18	62.30	-	-
PLAN [70]	-	75.31	65.52	-	61.34	50.86	58.03	-	-
S.L.R [65]	69.48	73.71	64.96	55.71	60.74	48.80	-	60.21	59.63
LGRAN [33]	-	76.6	66.4	-	64.0	53.4	62.5	-	-
DGA [35]	-	78.42	65.53	-	69.07	51.99	-	-	63.28
SVMN	75.44	80.96	68.54	65.53	70.22	55.92	-	66.93	64.94

Table 3.2: Comparison with state-of-the-art approaches on automatically detected proposals of Faster R-CNN (Acc%). SVMN also achieves state-of-the-art performance, which proves the robustness of our model.

on testB splits of RefCOCO and RefCOCO+ are the largest (about 3 percent point). Since testB splits of RefCOCO and RefCOCO+ contain multiple objects in different categories, it shows that our proposal to learn cross-modality matching in two manners has the advantage to solve complex instances that have multiple objects in different categories. Further analysis of each module is given in the following section.

No.	Model	RefCOCO		
		val	testA	testB
0	SVMN	87.73	87.01	87.34
1	<i>w/o</i> Residual Attention Parser	86.25	86.16	86.52
2	<i>w/o</i> Symbolic Module	85.93	86.05	85.30
3	<i>w/o</i> Relation Network	87.30	86.69	86.99

Table 3.3: Ablation studies of the main modules of SVMN (Acc%). ‘*w/o*’ represents replacing the module by a straightforward replacement or removing it directly. The results of erased models are compared with that of the full model and prove the effectiveness of the Residual Attention Parser, the symbolic module, and the relation network in the visual module.

3.3.4 Ablation experiment

In this part, we applied a set of ablation experiments to analyze the main modules in detail and the results are shown in Table. 3.3. The ablation experiments are conducted in a manner that each time one specifically designed module network is replaced by a straightforward placebo or taken off directly, which is denoted as ‘*w/o*.’ in the table. The results of the ablated models are compared with that of the full model respectively.

3.3.4.1 Language Parser

To analyze the effectiveness of the Residual Attention Parser, we replace it with the self-attention scheme used in [21, 33]. The substitute uses three isolated MLPs to compute phrases’ attention with context information only and another single MLP to compute three phrases’ weights with the final hidden state only. There is no residual information being shared between MLPs. Results (line 1) show that the performance of the Residual Attention Parser is about 1% better than that of the substitute, which proves the effectiveness of designed reasoning operations.

3.3.4.2 Symbolic Module

To test the effect of the symbolic module, we simply take this module off since the rest part of the model is still intact. Illustrating in line 2, the performance is much worse than that of the full model especially on RefCOCO’s and RefCOCO+’s testB splits

which contain multiple objects other than the people category. This is consistent with the previous analysis. It proves that the symbolic module is helpful especially for scenes with multiple objects in different categories. And the symbolic information is much more clear than visual features in the middle process.

3.3.4.3 Relation Module

To verify the impact of the relation module in the visual module, we replace it with a single MLP in the comparative depth and activation function of the relation module. The input to it is the concatenation of all inputs to the relation module. The result (line 3) shows the effectiveness of the relation network, though not as well as the above two, still very attractive. Most importantly, this does not introduce any extra computational complexity compared with the substitute.

3.3.5 Visualization

In Fig. 3.6, we show some visualizing instances of this model. The ground-truth object is labeled by blue solid boxes and the predicted object is located as red dashed boxes. Line a illustrates some successful instances with the phrase attention distributions below. For the distribution, the darker the color is, the larger the attention value is, and vice versa. The results show that different components could be decomposed properly, which is inclined with our designing purpose. Line b and c give more successful instances. From these two lines, we can find subject/category information is almost indispensable in referring expressions, which explains why the proposed symbolic network is so effective in the whole model.

3.4 Chapter Summary

Our SVMN model aims to address two problems in Visual Grounding, diverse expression processing, and visual-linguistic feature mapping. For the former problem, we propose a Residual Attention Parser. For the latter, we propose two mapping styles,

Figure 3.6: Visualizations of some successful referring instances. The solid blue box is the ground-truth object and the red dashed box is the predicted one. Line A illustrates successful instances with phrases' attention distributions below. For the distribution, the darker the color is, the larger the attention value is, and vice versa. Line B and C provide more successful referring instances.

language-to-symbol and language-to-vision. We design a symbolic module to apply language-to-symbol mapping and adopt three visual modules to process language-to-vision mapping. Finally, the model achieves state-of-the-art performance on three popular referring datasets, demonstrating the effectiveness of our model.

Chapter 4

Natural Language Video Localization

4.1 Introduction

Given an untrimmed video and a language expression, Natural Language Video Localization (**NLVL**) aims to localize a target moment that is semantically matched with the given language description. NLVL could be under wide applications in real life like information retrieval in the videos [71], language guided navigation [72]. In this chapter, we address three weaknesses of previous methods, namely feature representation, scene modeling, and problem formulation.

Feature Representation. Compared with the 2D images, videos contain more information due to the temporal dimension. To obtain the information of a video, the majority of previous work of NLVL [37–44] exploit the 3D Convolution Network like C3D [73] and I3D [74] pre-trained for action recognition [74, 75] to extract features of short clips, each of which contains dozens of continuous frames. The advantage of this is that the atomic representation contains temporal information for action prediction. However, to comprehensively understand a scene, modeling the relationships between the semantic level objects in the scene has shown its effectiveness in the image-text retrieval task [76–78]. And this is also in line with the intuition of NLVL. Given an example from ActivityNet

in Fig. 4.1, “A woman walks to the piano and briefly talks to the elderly man”, it needs to identify the interactions of “the woman” with “the piano” and “the woman” with “the elderly man”. However, it is difficult to capture these details for previous work. In light of this, we describe a scene in the video with the semantic objects in it. To avoid data explosion, we only sample sparse scenes with less than 2 fps from the original video sequence.

Figure 4.1: A showcase of annotations in ActivityNet. Figure source: [4].

Scene Modeling. Based on the observation, there are two kinds of common scenarios in the video, namely the static scene and the movement/action. Graphic examples are illustrated in Fig. 4.2. The static scene is about the states and the relationships of objects in the same spatial frame. The movement/action, on the other hand, should be expressed as the state change of the same object in different temporal points. Meanwhile, we notice that the semantically matching between objects and words could help the model paying more attention to the visual information related to the referring sentence while ignoring the irrelevant one. Considering these, we devise the **Hybrid Graph Network** with three types of sub-graphs, namely the spatial relation graph, a set of temporal relation graphs, and the query-guided graph. The spatial relation graph is designed to model

Figure 4.2: Illustration of common scenarios in the video. (a) is a static scene that could be described with the spatial information purely. (b) is a movement description that involves the interactions of objects in different scenes. (c) demonstrates a scenario that highlights the importance of semantically objects-and-words matching.

the static scene information with objects in the same frame, e.g., “the man holding a cup”, “the boy next to the car”. The temporal relation graphs aim to capture the locally temporal dependency of objects in neighboring scenes which corresponds to the actions and movements. For example, information like “open the door”, “running” and “skiing” should be expressed as the dynamic changes of the shape and position of the same object in different temporal points. The query-guided graph seeks semantically cross-modality matching between words and objects in the scene. The purpose of this is to give more attention to the objects which appear in the given query and dim the ones that are not. After the hybrid fusion, the original discrete objects in each scene are expressed as comprehensive representations in the context. To further capture the long-span relationships of the activities and events and also build event-level correlations between language and video, we also implement a temporal encoder based on the attentive model [3, 9, 10]. The encoder is a stack of self-attention and cross-attention layers that aim to conduct intra-modality and cross-modality interactions respectively in hierarchical levels.

Problem Formulation. Traditionally, there are two styles of methods solving this

task, namely anchor-based and anchor-free. Anchor-based methods like [37–39, 43] usually formulate the problem as a ranking task of the candidate clips that are generated from the video in different strategies. For prediction, the common practice is to fuse the linguistic feature with the clip’s feature to get a fused representation which is then used for predicting an alignment score indicating the matching degree and the fine-tuning offset to the actual boundaries. On the other hand, anchor-free methods like [1, 46–50] model the video as a continuous sequence holistically and predict the specific start and end boundaries in the sequence. As for the detailed operations of prediction, [46, 47] first perform a binary classification for each frame in the sequence, and then densely regress the distances from boundaries for all positive frames. However, compared with the classification, it is difficult to regress the accurate boundary values directly with the local representations. Therefore, recently [1, 48–50] directly predict three probability scores for each frame being the foreground annotation, the start, and end boundaries with two kinds of classification loss. We also model the localization as a classification task, but the difference from previous jobs is that we only conduct the binary classification of every atomic representation in the sequence being the foreground or background and then localize the start and end boundaries in the obtained score vector through some post-processing methods. The whole scheme of our model is illustrated as Fig. 4.3, and the whole model is still named as Hybrid Graph Network (**HyGNet**) to reflect our major innovation.

In conclusion, our main contributions are as follows:

- We represent a video with the sparse discrete semantic features of it and devise a new Hybrid Graph Network (HyGNet) to model the scenes. The HyGNet could capture the spatial and locally temporal relationships of discrete objects and also perform semantically cross-modality interaction. We also implement a temporal encoder based on the attentive model to capture the long-span relationships of events in the two modalities. Finally, we formulate the problem as a simple binary classification task.
- Considering the expensive cost of labeling video data, we design some simple but effective data augmentation methods to improve the variance of the dataset. Meanwhile, we exploit the predicted symbolic feature of objects for better representation.

Figure 4.3: The overall structure of the proposed HyGNet. The model includes three main sub-modules, namely the Hybrid Graph Network aiming to capture the spatial and local dependencies of scenes, the Temporal Encoder to model the temporal dependency and interact with the linguistic information, and the Span Predictor which localizes the target area with a single classifying loss.

- Experiments on two challenging datasets show that the proposed model could achieve surpassing or competitive performance over previous state-of-the-art work.

4.2 Model

In this section, we introduce our proposed HyGNet model in detail. As illustrated as Fig. 4.3, the model consists of three parts, namely the hybrid graph network, the temporal encoder, and the span predictor.

4.2.1 Problem Formulation

Given an untrimmed video $V = \{f_t \mid t = 1, \dots, t\}$ with t frames, and a text query $Q = \{w_j \mid j = 1, \dots, m\}$ with m words, the objective of NLVL is to localize a semantically matching moment on the video indicating by a starting point t^s and an ending point t^e . To extract the visual feature for our model, we first sample a set of frames $V' = \{f_i \mid i = 1, \dots, n\}$ at the frame rate of 2 fps on the original video V , where n is the number of frames. During the experiment, the long video sequence would be down-sampled again for mini-batch packing, so the actual feature sequence is quite sparse. For each frame in V' , we extract K objects with the largest objectness scores by a pre-trained Faster R-CNN [31]. Each object has a ROI pooled feature r_{ik} , a predicted class label c_{ik} , and a normalized position vector $p_{ik} = [\frac{x_{ik}^{tl}}{w_i}, \frac{y_{ik}^{tl}}{h_i}, \frac{x_{ik}^{br}}{w_i}, \frac{y_{ik}^{br}}{h_i}, \frac{w_{ik}h_{ik}}{w_i h_i}]$, where tl means top-left, br means bottom-right, x, y denote the location in the grid, and w, h denote the width and height of the image or the box. We then encode the class label of the object by the pre-trained GloVe embedding [63] and obtain the symbolic feature of the object denoted as s_{ik} . Thus, the original video V could be represented as $H^V = \{r_{ik}, s_{ik}, p_{ik} \mid i = 1, \dots, n; k = 1, \dots, K\}$, among which i is the index of a frame and k is the index of an object in the frame.

For the text query Q , we first encode words in it with the same pre-trained GloVe embedding, obtaining $E^Q = \{e_j \mid j = 1, \dots, m\}$. The word embedding E^Q is then encoded by a trainable one-layer bi-directional LSTM, and we take the hidden state of each word as the linguistic representation, denoted as $H^Q = \{q_j \mid j = 1, \dots, m\}$ below. Then the visual object sequence H^V and the linguistic sequence H^Q will be the input of our model.

4.2.2 Hybrid Graph Network

Intuition. Different from 3D-Conv features extracted over short video clips which contain both the scene information and temporal dependency, the semantic features of each scene are discrete at first. To handle this, we propose the hybrid graph network to make the discrete features interact spatially and temporally through a series of graph operations.

To explain the intuition of the hybrid graph network, let's take a closer look at an

example “A man sits with his dog in the grass and holds out discs in his hands”. First, the first part of the query “A man sits with his dog in the grass” describes a still scene where its counterpart in visual space should be represented as the relationships of objects within the frame. A similar need is seen in Visual Grounding (VG) [21] and Visual Question Answering (VQA) [79], which is usually handled by the graph operation or sequence model processing. In our network, this is handled by the spatial relation graph operation.

Second, some scenarios in the video are not static but involve actions and movements. For example, “man holds out discs in his hands”, should be a set of dynamic and coherent states of the “man” and the “discs” in adjacent scenes. To capture this temporal change and continuity of an object in different scenes, we first compare this object with objects in adjacent frames to find its correspondences. Then the movement/action is expressed as the difference of states (feature and position) between the corresponding objects in two frames. This process is realized by the soft-attentive operation over objects in several adjacent frames.

Finally, to make better use of annotated information and filter out the irrelevant visual information, we perform semantically matching between words and objects by the query-guided graph. The intention of this operation is to highlight the objects which appear in the given language and dim the ones that are not. For the given language above, the objects which are predicted as “man”, “dog”, “grass”, and “discs” should have high attention and the other objects should not.

Graph Structure. We adopt the Graph Attention Network (GAN) as the backbone of the hybrid graph. The objects of the scene and the words in the query are organized as the nodes of different graphs. In the GAN, the connections between nodes are computed through the attention mechanism. During operation, we choose the structure of Transformer proposed by [3] as the standard message-passing layer of the graph. Formally, we can formulate one layer of the update operation as

$$e_{ij} = \text{attention}(Qh_i, Kh_j, Vh_j), \quad (4.1a)$$

$$a_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \quad (4.1b)$$

$$h'_i = \text{LN}(h_i + W \sum_{j \in \mathcal{N}_i} a_{ij} h_j), \quad (4.1c)$$

where the *attention* in Eq. 4.1a is the multi-head scaled dot-product attention, $\mathcal{N}_i$ is the neighbors of node i , and LN denotes Layer Normalization. In practice, another $\text{LN}(x + Wx)$ operation is applied for h'_i . For ease of expression, we represent one updating layer above as $F(h_i, h_{\mathcal{N}_i})$, where h_i denotes the hidden state of the node i being processed and $h_{\mathcal{N}_i}$ indicates the states of its neighbors. In the process, one node is able to selectively absorb the information of its neighborhoods, thereby updating its representation in the context. Different from the conventional sequence-aligned model, the updating operations in GNN are computed in parallelization.

Hybrid Graph. According to the work in image and text tasks like VG and VQA [9, 66, 80, 81], to represent an object in a scene, we need to take both the semantic feature and positional information into consideration. Besides, we exploit both the ROI feature and the symbolic feature to better represent the semantic information of an object v_{ik} . The symbolic feature and the ROI feature of the object are first fused as Eq. 4.2a. Then we combine the position vector p_{ik} with the semantic feature to get the overall representation h_{ik}^V of an object (Eq. 4.2b). Two operations are represented as

$$v_{ik} = W_{rs}(W_r r_{ik} + W_s s_{ik}), \quad (4.2a)$$

$$h_{ik}^V = \text{LN}(W_v v_{ik} + W_p p_{ik}), \quad (4.2b)$$

where all W are learnable parameters, and LN denotes layer normalization. The visual sequence is then represented as $H^V = \{h_{ik}^V \mid i = 1, \dots, n; k = 1, \dots, K\}$.

Spatial relation graph is intended to model the static scene with the objects in the same frame denoting as $v_i = \{v_{ik} \mid k = 1, \dots, K\}$. To assemble it, v_i is organized as the node collection of the graph. For a node v_{ik} , the neighborhoods of it are v_i which contain a self-connecting edge. The initial hidden state of the node v_{ik} is h_{ik}^V , and the operation of this graph is represented as $G_s = F(h_{ik}^V, h_i^V)$.

Temporal relation graphs aim to capture the locally temporal dependencies of objects in adjacent frames. For an object v_{ik} in frame i , we sample multiple neighbor collections of it from adjacent frames with different strides/distances and directions. The neighbor collections could be formulated as $\mathcal{N}_{v_{ik}} = \{v_{(i+\delta)} \mid \delta \in \text{strides}\}$, where the *strides* is a hyper-parameter indicating the distances to frame i . With the expansion of *strides*, the longer dependency could be captured, but at the same time the computation would increase. During experiments, we set *strides* in the dilated style to expand the dependency with the restraint of computation. Then the temporal graphs for the node v_{ik} are composed as $G_t = \{F_\delta(h_{ik}^V, h_{i+\delta}^V) \mid \delta \in \text{strides}\}$. During operation, F_δ in the same direction¹ share weights in each layer. The new representation of the node v_{ik} contains the state difference with its counterparts in neighboring frames.

Query-guided graph is applied for semantically cross-modality matching. The word collection $H^Q = \{q_j \mid j = 1, \dots, m\}$ is used as neighbor collection for each object in all frames. The operation for an object v_{ik} could be represented as $G_q = F(h_{ik}^V, H^Q)$.

After an updating layer, the hybrid graph first sums up the hidden states of all sub-graphs and then normalize the result, formulated as

$$G = LN\left(\sum_{m \in \{s, t, q\}} G_m\right), \quad (4.3)$$

where G_m belongs to all the sub-graphs presented above, and LN denotes layer normalization. In each layer, there are two reusable temporal relation graphs for the forward and backward direction respectively. The layer updating process is performed N times during the experiment for capturing longer distance dependency. Whole process is illustrated as Fig. 4.4.

Scene Fusion. After the operation of the hybrid graph network, the feature of each node is a comprehensive representation in the context and is also query-aware. The graph operation does not change the dimension of the input visual feature, which is still represented as $H^V = \{h_{ik}^V \mid i = 1, \dots, n; k = 1, \dots, K\}$. To obtain the holistic representation of the scene and also facilitate the subsequent processing, we fuse all

¹The sign of δ is the same.

Figure 4.4: Illustration of the structure of one layer’s graph operation.

objects in a scene through the self-attention mechanism, formulated as

$$a_{ik} = \text{Softmax}_k(Wh_{ik}^V), \quad (4.4a)$$

$$h_i^V = \sum_{k=1}^K a_{ik} h_{ik}^V. \quad (4.4b)$$

4.2.3 Temporal Encoder

The processed scene sequence $H^V = \{h_i^V \mid i = 1, \dots, n\}$ contains the locally temporal dependency with the interaction of the hybrid graph operation. However, there is usually the long-distance dependency relationship in the video, i.e. the relationship of events. For example, “A man opens the refrigerator, takes out a carton of milk, and pours it into a cup”. This sentence consists of three sequential events which distribute over a long distance in the temporal space. Previous work [46, 48] exploit conventional sequence models to encode linguistic and visual features separately, then combine two encoded features finally. In this thesis, we adopt the ViLBERT model [9] which is based on the Transformer structure as the backbone of our temporal encoder. ViLBERT model is a multi-modality two-stream model to solve image-and-text tasks like VG and VQA at first. Here we adopt it to process the video and text which is more in line with its operating condition. Since the ViLBERT model contains no recurrence, we first add the positional

information² p_i of the sequence by the “positional encoding” (PE) as

$$h_i^V = LN(h_i^V + W_p p_i). \quad (4.5)$$

The encoder is composed of M layers to fuse information in hierarchical scales. One layer in the encoder is a standard self-attention (SA) layer followed by a co-attention (CA) layer. The structure of the SA and CA layers is the same and formulated as Eq. 4.1a, 4.1b, 4.1c. The difference between them lies in the source of the reference information. The SA layer applies intra-modality interaction in global scope for the visual and linguistic sequences respectively, and the CA layer applies cross-attentive operation from video to query and query to video. With the stack of multiple layers’ operation, the visual and linguistic features interact in different levels, which could better capture the events in different scales. For more information, we recommend reading [9]. The output visual embedding $H^V \in \mathbb{R}^{n \times D}$ is used for prediction in the next section.

4.2.4 Span Predictor

Inspired by the practice in TVRetrieval [82], we model the prediction of the target span as the binary classification of every atomic feature being foreground or background rather than regressing specific boundaries with the local representations. We exploit a 1D convolution layer with a single output dimension to process the output sequence of the temporal encoder, formulated as

$$S_{span} = \sigma(Conv_{span}(H^V)), \quad (4.6)$$

where σ is the *Sigmoid()*, and $S_{span} \in \mathbb{R}^{n \times 1}$. Then we apply the Gaussian Smoothing to S_{span} for better continuity. To transform the span score into the starting and ending boundaries, we exploit two 1D boundary detectors to localize the boundaries, represented as

$$S_{start} = Conv_{start}(S_{span}), \quad (4.7a)$$

²The positional information we exploited is the sequence indexes normalized by the sequence length.

$$S_{end} = Conv_{end}(S_{span}). \quad (4.7b)$$

The weights of the boundary detectors are similar to the edge detectors in image processing³. We pad on the two sides of S_{span} by the minimum value in the vector during experiments. With the obtained boundary score vectors, we apply $argmax()$ function to locate the specific boundary points.

The whole model is trained in an end-to-end fashion without any pre-training. To supervise the training, we exploit the binary cross-entropy loss to compute the loss of the predicted span score S_{span} with the ground-truth annotation, formulated as

$$loss = BCE(S_{span}, S_{gt}). \quad (4.8)$$

$BCE()$ indicates Binary Cross-Entropy loss, and S_{gt} is the ground-truth span score where the foreground frames are labeled as 1 and the backgrounds are labeled as 0.

4.2.5 Effective Data Augmentation

Compared with the annotation of images, it is more time-consuming to label the video dataset. Therefore we propose several simple but effective data augmentation methods to improve the variance of training data.

Video Splicing. For a video, we splice a randomly sampled video clip from other videos at the beginning or end of it. This operation is applied to expand the variance of the relatively short videos.

Video Cropping. For a video, we randomly cut off a clip from the background area of the video. The length of the cropping part is not greater than the length of the margins on both sides of the target span. This operation is applied for relatively long videos to increase their diversity.

Negative Pairing. Inspired by the cross-modality alignment of the image and text, we propose a high-level augmentation method, which is to randomly compose unmatched

³ $[-1, -0.5, 0, 0.5, 1]$ as the weight of starting boundary detector, and $[1, 0.5, 0, -0.5, -1]$ as the weight of ending boundary detector, without weight updating during training.

video and text pairs. During training, the ground-truth span labels of the negative pairs are all zeros. Since the boundary detectors of our model are not trainable, this augmentation method will only affect the model before the span score prediction.

Word Masking. To push the model makes full use of the language rather than attending to single words, we randomly mask words in the input query by “unk” which is a trainable embedding vector.

During experiments, four augmentation methods are sampled with the probability of $[0.2, 0.2, 0.4, 0.2]$. And the augmentation is only operated during the training period with the probability of 0.5.

4.3 Experiments

4.3.1 Datasets

We conduct experiments on two difficult datasets of NLVL, namely ActivityNet Captions [4] and TACoS [75].

ActivityNet Captions (denoted as ActivityNet) is a large-scale benchmark that was first developed for Dense Video Captioning [83,84]. It contains 20k videos amounting to 849 video hours with a total of 100k densely annotated descriptions. On average, each video contains 3.65 annotated sentences, and each sentence has an average length of 13.48 words, both of which follow a relatively normal distribution. The average video length is 117.74 seconds and the average target span is 37.14 seconds. Due to the non-availability of the test set, following the practice in previous work, we take the val_1 as the validation set and val_2 as the test set.

TACoS is built on top of the “MPII Cooking Composite Activities” video corpus [85]. It only contains 127 long videos of different activities in the cooking domain. The video is on average of about 5 minutes (ranging from 1 to 23 minutes). The averaged target span is 6.1 seconds and the averaged query length is 8.86 words. We follow the setting in [37], where there are 10, 146, 4, 589, and 4, 083 annotations for training, validation, and testing, respectively. Compared with ActivityNet, the localizing requirement of TACoS

is higher.

4.3.2 Experimental settings

Implementation Details. To extract the semantic feature of the video sequence, we first sample frames in the video with 2 fps and then extract top $K = 8$ proposals for each frame by Faster R-CNN⁴ and the image feature by ResNet⁵. The image feature is regarded as an additional node of the frame, which means we have $K+1$ proposals for each scene. It is worth mentioning that we only fetch objects with the objectness scores larger than 0.5 and pad the batch with zero if the number of objects is not enough. To facilitate training, we down-sample/pad the frame sequence into 180 for ActivityNet and 360 for TACoS. We embed words with 300d GloVe [63] and curtail/pad the linguistic sequence into 30 for ActivityNet and 15 for TACoS. For the graph network, we set 8 heads for attention in Transformer and set all hidden dimensions of the visual and linguistic features as 256. And the *strides* for temporal relation graphs is set as $[-7, -5, -3, -1, 1, 3, 5, 7]$ for ActivityNet and $[-5, -3, -1, 1, 3, 5]$ for TACoS. A feature projection layer (one fully connected layer and the ReLU activation) is applied for proposal feature, image feature, and word embedding respectively before putting them into the network. The kernel size of the span detector $Conv_{span}$ is 3. The model is trained for 100 epochs with the batch size of 50 and the early stopping strategy. Parameter optimization is performed by Adam [87] with the initial learning rate of 0.0005, gradient clipping of 1.0, and a warm-up proportion of 0.05. Dropout with the rate of 0.4 and weight decaying with the rate of 0.01 is applied to prevent over-fitting.

Evaluation Metrics. Following the common practice in previous work, we adopt the “R@n, IoU= μ ” and “mIoU” as the evaluation metrics. Specifically, the “R@n, IoU= μ ” means the percentage of the top-n predicted moments having the IoU (i.e. Intersection over Union) with ground-truth moment larger than μ . “mIoU” is the average IoU over all testing samples. In our experiments, we use $n = 1$ and $\mu \in \{0.3, 0.5, 0.7\}$, denoting as $R@\mu$ below.

⁴ResNet 50 with FPN as the backbone, pre-trained on MSCOCO [67].

⁵ResNet-152, pre-trained on ImageNet [86].

Table 4.1: Performance comparisons of our approach with existing state-of-the-art methods.

model	ActivityNet				TACoS			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Baseline	42.45	22.30	11.25	32.05	5.90	1.55	0.75	8.73
CTRL [37]	47.43	29.01	10.34	20.54	18.32	13.30	-	-
QSPN [41]	45.30	27.70	13.60	-	20.15	15.23	-	-
DEBUG [46]	55.91	39.72	-	39.51	23.45	11.72	-	16.03
CBP [88]	54.30	35.76	17.80	36.85	27.31	24.79	19.10	21.59
TMLGA [49]	51.28	33.04	19.26	-	24.54	21.65	16.46	-
ExCL [48]	<u>63.00</u>	<u>43.60</u>	23.60	-	45.50	28.00	13.80	-
LGVTI [89]	58.52	41.51	23.07	41.13	-	-	-	-
U-VMR [45]	46.15	26.38	11.64	-	-	-	-	-
HVTG [53]	57.60	40.15	18.27	-	-	-	-	-
DORi [52]	57.89	41.49	26.41	<u>42.78</u>	31.80	<u>28.69</u>	24.91	<u>26.42</u>
HyGNet	63.18	44.29	<u>26.10</u>	44.87	<u>43.26</u>	30.12	<u>19.65</u>	30.65

4.3.3 Comparison with the state-of-the-art

We compare the proposed HyGNet with existing state-of-the-art work in Table. 4.1. The best results are in **bold** and the second bests are underlined. The default setting of the reported HyGNet is $M = 4$ layers hybrid graph network and $N = 4$ layers temporal encoder. Proposed augmentation methods are applied.

In the Table. 4.1, we report the performance of all retrieval, i.e. using the entire video as the retrieving result as the baseline for reference. CTRL [37] is a traditional anchor-based (sliding window) method that model the localization as the ranking of candidate clips. QSPN [41] introduces a multi-level model to integrate visual and textual features earlier with an attention mechanism for generating clip proposals. U-VMR [45] exploits the graph network to model the relationships among video clips. DEBUG [46] is an anchor-free model that applies a binary classification for frames and then regresses the boundaries on the positive ones. While ExCL [48] and TMLGA [49] model the prediction of the span boundaries as the pure classification task. HVTG [53] and DORi [52] are the most similar work to ours as they also exploit the semantic feature of the scene. [53] utilizes graph networks to model the relationships of objects in the scene and the interactions between the objects and words. The key idea of DORi is to learn a video feature embedding through a language-conditioned message-passing algorithm

Table 4.2: Ablation results of the layer composition analysis.

No	settings	R@0.3	R@0.5	R@0.7	mIoU
1	N = 0	52.98	30.56	15.73	37.47
2	N = 2	62.90	43.32	23.47	44.41
3	N = 4	63.18	44.29	26.10	44.87
4	N = 6	62.71	42.24	23.55	44.25
5	M = 0	56.47	36.03	20.46	39.80
6	M = 2	61.79	42.21	24.34	43.70
7	M = 4	63.18	44.29	26.10	44.87
8	M = 6	62.75	43.42	25.47	43.66

over a spatial graph between objects and humans. These two methods fail to capture the continuity of objects in the temporal space, and the organization of graphs is different from ours.

According to Table. 4.1, our HyGNet achieves superior or competitive results compared with the state-of-the-art methods. It needs to mention that our model only exploits the sparse semantic feature (≤ 2 fps) extracted by Faster R-CNN pre-trained on MS-COCO. While previous methods exploit 3D-Conv features extracted by I3D or C3D model. DORi [52] exploits semantic feature which is extracted by Faster R-CNN pre-trained on Visual Genome with more fine-grained annotations and also the I3D features for the activity graph. Through the comparison of results, we can find that using our network to model the semantic information in the sparse frames could achieve superior or competitive performance to that of the network using 3D features.

4.3.4 Ablation Experiments

To validate the effectiveness of our proposed model, we conduct some ablative experiments in different aspects. All ablation experiments are conducted on the ActivityNet dataset.

Layer Composition Analysis.

To find an ideal layer composition for Hybrid Graph Network N and Temporal Encoder M , we conduct a series of tests with different layer compositions shown in Table. 4.2. Lines 1 to 4 are with $M = 4$ and lines 5 to 8 are with $N = 4$ for comparison.

Table 4.3: Comparison of results of sub-graphs ablation in Hybrid Graph Network.

No	settings	R@0.3	R@0.5	R@0.7	mIoU
0	default	63.18	44.29	26.10	44.87
1	w/o G_q	62.51	41.16	22.76	43.76
2	w/o G_t	60.16	41.78	23.28	43.30
3	w/o G_s	63.41	42.12	22.85	44.32
4	w/o HGNet	52.98	30.56	15.73	37.47

Table 4.4: Results of ablation experiments of the feature analysis.

No	settings	R@0.3	R@0.5	R@0.7	mIoU
0	default	63.18	44.29	26.10	44.87
1	w/o augment	62.70	43.66	25.08	44.47
2	w/o symbolic	62.42	43.10	24.07	43.91
3	w/o image	62.60	43.85	24.90	44.51

According to the experimental results in Table. 4.2, the model without the Hybrid Graph Network (line 1) or Temporal Encoder (line 5) have a bad performance, which demonstrates the effectiveness of the two proposed modules. With the number of layers increasing from 2 to 4, the effect of the model gradually gets better. However, when the number of layers increases too large (6), the performance does not increase much or even decrease a bit. This limitation could be caused by insufficient data volume because our model does not use any large-scale data for pre-training. Therefore, we choose $N = 4$ and $M = 4$ as the default setting of the network.

Hybrid Graphs Analysis.

To analyze the effectiveness of our proposed Hybrid Graph Network, we conduct ablation experiments by removing one sub-graph at a time and compare the performance of the ablated model with that of the full model under the setting of $N = 4$, $M = 4$. According to the Table. 4.3, the model without any sub-graph has inferior performance compared to that of the default setting. If we only compare the $R@0.7$ column, the query-guided graph has a higher effect on the performance, which shows the importance of conducting semantically cross-modality (objects and words) matching. Besides, the spatial graph and temporal graph are also necessary. The model without the entire Hybrid Graph Network has terrible performance.

Feature Analysis.

We conduct three experiments to demonstrate the effects of the data augmentation, the usage of the symbolic feature and the image feature respectively. According to the comparison of the results in Table. 4.4, three feature processing methods all have some positive effect, especially the usage of the symbolic feature. It is worth mentioning that the model trained without the data augmentation methods also performs well, which demonstrates the effectiveness of the model structure. Compared with the ROI feature, we argue that the symbolic feature is more in line with the referring linguistic feature and could provide a more accurate representation of an object.

4.3.5 Visualization

Fig. 4.5 provides some visualization of the localization results learned by our model. (a) to (c) are three successful instances with iou larger than 0.99, while (d) is an unsatisfactory predicting result with iou at 0.34. In more detail, (a) is a scenario shot underwater, which has poor picture quality. However, the annotated description includes the word “car”, and only the ending part of the video appears as “car”. So we argue that our proposed semantically cross-modality matching in Query-guided Graph is very useful. (b) and (c) are the examples that require the reasoning of relationships between objects in the scene. And the unsatisfactory prediction in (d) may be due to the similarity of scenes in different parts of the video. This is more common in relatively short videos.

4.4 Chapter Summary

In this chapter, we propose the HyGNet to address three weaknesses of previous work in NLVL. The proposed network is able to model the sparse discrete semantic features in the video through two stages, one is to build local relationships between objects and capture connection between objects and words, the other is to construct the long-span dependencies in the two sequential features. Experimental results validate the effectiveness of the proposed model.

Figure 4.5: Illustration of some localization results. (a) to (c) are three successful instances with the iou larger than 0.99. While (d) is an unsatisfactory prediction, where the green bars indicates the scope of prediction.

Chapter 5

Conclusion

In this thesis, we take research on two sub-tasks of Language-guided Visual Retrieval, one is Visual Grounding (VG), and the other is Natural Language Video Localization (NLVL).

For VG, our proposed SVMN model aims to address two problems, i.e. the diverse expression processing, and the visual-linguistic feature mapping. For the former problem, we propose a Residual Attention language Parser to take the computing results in different phrases into consideration. For the latter, we propose two mapping styles, language to symbol and language to vision. In detail, we design a symbolic module to match the symbolic feature of the object with the language and adapt three traditional visual modules to exploit the subject, location, and relation information in the image. Finally, the model achieves state-of-the-art performance on three popular referring datasets, demonstrating the effectiveness proposed two styles of alignment.

To solve the problems in NLVL, we propose the HyGNet which utilizes the fine-grained semantic features in the sparse frames of a video sequence. To capture the spatial and temporal dependencies in discrete objects in the scenes, and conduct object-to-words alignment at the same time, we design three different sub-graphs. To modeling the long-span relation between events in the two sequential features, we additionally perform a temporal encoding based on the attentive structure. The final experiment results prove the effectiveness of the proposed model.

References

- [1] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, “Span-based localizing network for natural language video localization,” *arXiv preprint arXiv:2004.13931*, 2020.
- [2] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [4] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, “Dense-captioning events in videos,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 706–715.
- [5] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, 2015, pp. 2048–2057.
- [6] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.
- [7] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, “Massive exploration of neural machine translation architectures,” *arXiv preprint arXiv:1703.03906*, 2017.
- [8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

-
- [9] J. Lu, D. Batra, D. Parikh, and S. Lee, “Vilbert: Pretraining task-agnostic vi-siolinguistic representations for vision-and-language tasks,” in *Advances in Neural Information Processing Systems*, 2019, pp. 13–23.
 - [10] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, “12-in-1: Multi-task vision and language representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 437–10 446.
 - [11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
 - [12] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, “Benchmarking graph neural networks,” *arXiv preprint arXiv:2003.00982*, 2020.
 - [13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
 - [14] X. Bresson and T. Laurent, “Residual gated graph convnets,” *arXiv preprint arXiv:1711.07553*, 2017.
 - [15] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.
 - [16] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Learning to compose neural networks for question answering,” *arXiv preprint arXiv:1601.01705*, 2016.
 - [17] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 804–813.
 - [18] R. Hu, J. Andreas, T. Darrell, and K. Saenko, “Explainable neural computation via stack neural module networks,” in *European Conference on Computer Vision*, 2018.

- [19] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, “Modeling relationships in referential expressions with compositional modular networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1115–1124.
- [20] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Inferring and executing programs for visual reasoning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2989–2998.
- [21] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, “Mattnet: Modular attention network for referring expression comprehension,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1307–1315.
- [22] R. Hu, M. Rohrbach, and T. Darrell, “Segmentation from natural language expressions,” in *European Conference on Computer Vision*. Springer, 2016, pp. 108–124.
- [23] R. Li, K. Li, Y.-C. Kuo, M. Shu, X. Qi, X. Shen, and J. Jia, “Referring image segmentation via recurrent refinement networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5745–5753.
- [24] H. Shi, H. Li, F. Meng, and Q. Wu, “Key-word-aware network for referring expression image segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 38–54.
- [25] X. Li and S. Jiang, “Bundled object context for referring expressions,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2749–2760, 2018.
- [26] L. Ye, M. Rochan, Z. Liu, and Y. Wang, “Cross-modal self-attention network for referring image segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 502–10 511.
- [27] L. Ye, Z. Liu, and Y. Wang, “Dual convolutional lstm network for referring image segmentation,” *IEEE Transactions on Multimedia*, 2020.

-
- [28] T. Wang, Z. Ji, Q. Sun, Q. Chen, and X.-Y. Jing, “Interactive multilabel image segmentation via robust multilayer graph constraints,” *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2358–2371, 2016.
 - [29] J. Tighe, M. Niethammer, and S. Lazebnik, “Scene parsing with object instances and occlusion ordering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3748–3755.
 - [30] S. Qiu, Y. Zhao, J. Jiao, Y. Wei, and S. Wei, “Referring image segmentation by generative adversarial learning,” *IEEE Transactions on Multimedia*, vol. 22, no. 5, pp. 1333–1344, 2019.
 - [31] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.
 - [32] X. Liu, Z. Wang, J. Shao, X. Wang, and H. Li, “Improving referring expression grounding with cross-modal attention-guided erasing,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1950–1959.
 - [33] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, “Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1960–1968.
 - [34] S. Yang, G. Li, and Y. Yu, “Cross-modal relationship inference for grounding referring expressions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4145–4154.
 - [35] S. Yang, G. Li, and Y. Yu, “Dynamic graph attention for referring expression comprehension,” *arXiv preprint arXiv:1909.08164*, 2019.
 - [36] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, “Localizing moments in video with natural language,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5803–5812.

- [37] J. Gao, C. Sun, Z. Yang, and R. Nevatia, “Tall: Temporal activity localization via language query,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5267–5275.
- [38] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, “Cross-modal moment localization in videos,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 843–851.
- [39] R. Ge, J. Gao, K. Chen, and R. Nevatia, “Mac: Mining activity concepts for language-based temporal localization,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 245–253.
- [40] S. Chen and Y.-G. Jiang, “Semantic proposal for activity localization in videos via sentence query,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8199–8206.
- [41] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, “Multilevel language and vision integration for text-to-clip retrieval,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9062–9069.
- [42] S. Zhang, J. Su, and J. Luo, “Exploiting temporal relationships in video moment localization with natural language,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1230–1238.
- [43] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. 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*, 2019, pp. 1247–1257.
- [44] S. Zhang, H. Peng, J. Fu, and J. Luo, “Learning 2d temporal adjacent networks for moment localization with natural language,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 870–12 877.
- [45] J. Gao and C. Xu, “Learning video moment retrieval without a single annotated video,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

-
- [46] C. Lu, L. Chen, C. Tan, X. Li, and J. 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 (EMNLP-IJCNLP)*, 2019, pp. 5147–5156.
- [47] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, “Dense regression network for video grounding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 287–10 296.
- [48] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann, “Excl: Extractive clip localization using natural language descriptions,” *arXiv preprint arXiv:1904.02755*, 2019.
- [49] C. Rodriguez, E. Marrese-Taylor, F. S. Saleh, H. Li, and S. 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*, 2020, pp. 2464–2473.
- [50] H. Tang, J. Zhu, M. Liu, Z. Gao, Z. Cheng *et al.*, “Frame-wise cross-modal match for video moment retrieval,” *arXiv preprint arXiv:2009.10434*, 2020.
- [51] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “Qanet: Combining local convolution with global self-attention for reading comprehension,” *arXiv preprint arXiv:1804.09541*, 2018.
- [52] C. Rodriguez-Opazo, E. Marrese-Taylor, B. Fernando, H. Li, and S. 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*, 2020, pp. 1079–1088.
- [53] S. Chen and Y.-G. Jiang, “Hierarchical visual-textual graph for temporal activity localization via language,” in *European Conference on Computer Vision*. Springer, 2020, pp. 601–618.

- [54] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, “Referitgame: Referring to objects in photographs of natural scenes,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.
- [55] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.
- [56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [57] Y. Liu, Y.-M. Zhang, X.-Y. Zhang, and C.-L. Liu, “Adaptive spatial pooling for image classification,” *Pattern Recognition*, vol. 55, pp. 58–67, 2016.
- [58] Z. Fang, J. Liu, Y. Li, Y. Qiao, and H. Lu, “Improving visual question answering using dropout and enhanced question encoder,” *Pattern Recognition*, vol. 90, pp. 404–414, 2019.
- [59] Z. Bai, Y. Li, M. Woźniak, M. Zhou, and D. Li, “Decomvqanet: Decomposing visual question answering deep network via tensor decomposition and regression,” *Pattern Recognition*, vol. 110, p. 107538, 2020.
- [60] Q. Cheng, Q. Zhang, P. Fu, C. Tu, and S. Li, “A survey and analysis on automatic image annotation,” *Pattern Recognition*, vol. 79, pp. 242–259, 2018.
- [61] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, “Visual grounding via accumulated attention,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7746–7755.
- [62] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with compositional vector grammars,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.
- [63] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

- [64] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, “Modeling context in referring expressions,” in *European Conference on Computer Vision*. Springer, 2016, pp. 69–85.
- [65] L. Yu, H. Tan, M. Bansal, and T. L. Berg, “A joint speaker-listener-reinforcer model for referring expressions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7282–7290.
- [66] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, “A simple neural network module for relational reasoning,” in *Advances in neural information processing systems*, 2017, pp. 4967–4976.
- [67] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [68] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, “Modeling context between objects for referring expression understanding,” in *European Conference on Computer Vision*. Springer, 2016, pp. 792–807.
- [69] H. Zhang, Y. Niu, and S.-F. Chang, “Grounding referring expressions in images by variational context,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4158–4166.
- [70] B. Zhuang, Q. Wu, C. Shen, I. Reid, and A. van den Hengel, “Parallel attention: A unified framework for visual object discovery through dialogs and queries,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4252–4261.
- [71] G. Castanon, M. Elgharib, V. Saligrama, and P.-M. Jodoin, “Retrieval in long-surveillance videos using user-described motion and object attributes,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 12, pp. 2313–2327, 2015.

- [72] W. Zhang, C. Ma, Q. Wu, and X. Yang, “Language-guided navigation via cross-modal grounding and alternate adversarial learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
- [73] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.
- [74] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.
- [75] M. Regneri, M. Rohrbach, D. Wetzell, S. Thater, B. Schiele, and M. Pinkal, “Grounding action descriptions in videos,” *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 25–36, 2013.
- [76] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, “Language-conditioned graph networks for relational reasoning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 294–10 303.
- [77] Y. Liu, B. Wan, X. Zhu, X. He *et al.*, “Learning cross-modal context graph for visual grounding,” 2020.
- [78] S. Yang, G. Li, and Y. Yu, “Graph-structured referring expression reasoning in the wild,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9952–9961.
- [79] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
- [80] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, “Uniter: Learning universal image-text representations,” *arXiv preprint arXiv:1909.11740*, 2019.

-
- [81] H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” *arXiv preprint arXiv:1908.07490*, 2019.
 - [82] J. Lei, L. Yu, T. L. Berg, and M. Bansal, “Tvr: A large-scale dataset for video-subtitle moment retrieval,” *arXiv preprint arXiv:2001.09099*, 2020.
 - [83] N. Xu, A.-A. Liu, Y. Wong, Y. Zhang, W. Nie, Y. Su, and M. Kankanhalli, “Dual-stream recurrent neural network for video captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2482–2493, 2018.
 - [84] T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, “Event-centric hierarchical representation for dense video captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
 - [85] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, “Script data for attribute-based recognition of composite activities,” in *European conference on computer vision*. Springer, 2012, pp. 144–157.
 - [86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
 - [87] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
 - [88] J. Wang, L. Ma, and W. Jiang, “Temporally grounding language queries in videos by contextual boundary-aware prediction.” in *AAAI*, 2020, pp. 12 168–12 175.
 - [89] J. Mun, M. Cho, and B. Han, “Local-global video-text interactions for temporal grounding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 810–10 819.