REFERRING EXPRESSION SEGMENTATION: FROM CONVENTIONAL TO GENERALIZED

LIU CHANG

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2024

Referring Expression Segmentation: From Conventional to Generalized

Liu Chang

School of Electrical & Electronic Engineering

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

2024

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 Jun 2023
.....

Date

.....

Liu Chang

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.

16 Jun 2023

.....

Date

.....

Prof. Jiang Xudong

Authorship Attribution Statement

This thesis contains material from 5 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Chang Liu, Xudong Jiang and Henghui Ding, "Instance-Specific Feature Propagation for Referring Segmentation," in IEEE Transactions on Multimedia (TMM), doi: 10.1109/TMM.2022.3163578.

The contributions of the co-authors are as follows:

- Prof. Jiang supervised the team on the overall research direction.
- I prepared the manuscript drafts, developed the algorithm, performed all the experiments and collected the results.
- Dr.Ding gave valuable suggestions on the research direction and the design of the algorithm.
- Prof. Jiang and Dr. Ding together revised the manuscript.

Chapter 4 is published as Henghui Ding*, Chang Liu*, Suchen Wang and Xudong Jiang, "Vision-Language Transformer and Query Generation for Referring Segmentation," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi: 10.1109/ICCV.48922.2021.01601.

The contributions of the co-authors are as follows:

- Prof. Jiang supervised the team on the overall research direction.
- I prepared the manuscript drafts, performed all the experiments and collected the results.
- Dr.Ding and I together determined the research direction and the design of the algorithm.
- Prof. Jiang, Dr. Ding and Dr. Wang together revised the manuscript.

Chapter 4 is also published as Henghui Ding*, Chang Liu*, Suchen Wang and Xudong Jiang, "VLT: Vision-Language Transformer and Query Generation for Referring Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2022, doi: 10.1109/TPAMI.2022.3217852.

The contributions of the co-authors are as follows:

- Prof. Jiang supervised the team on the overall research direction.
- I prepared the manuscript drafts, performed all the experiments and collected the results.

- Dr.Ding and I together determined the research direction and the design of the algorithm.
- Prof. Jiang, Dr. Ding and Dr. Wang together revised the manuscript.

Chapter 5 is published as Chang Liu, Henghui Ding, Yunlun Zhang and Xudong Jiang, "Multi-Modal Mutual Attention and Iterative Interaction for Referring Image Segmentation," in IEEE Transactions on Image Processing (TIP), 2023, doi: 10.1109/TIP. 2023.3277791

The contributions of the co-authors are as follows:

- Prof. Jiang supervised the team on the overall research direction.
- I prepared the manuscript drafts, developed the algorithm, performed all the experiments and collected the results.
- Dr.Ding gave valuable suggestions on the research direction.
- Prof. Jiang, Dr. Ding and Dr. Zhang together revised the manuscript.

Chapter 6 is published as Chang Liu, Xudong Jiang and Henghui Ding, "GRES: Generalized Referring Expression Segmentation," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 23592-23601.

The contributions of the co-authors are as follows:

- Prof. Jiang supervised the team on the overall research direction.
- I prepared the manuscript drafts, developed the algorithm, performed all the experiments and collected the results.
- Dr.Ding gave valuable suggestions on the research direction.
- Prof. Jiang and Dr. Ding together revised the manuscript.

*: equal contribution, co-first authorship.

16 Jun 2023

Date

Liu Chang

Acknowledgements

Completing this PhD has been a profoundly enriching and challenging journey, one that would not have been possible without the support and encouragement of many.

First and foremost, I extend my deepest gratitude to my supervisor, Prof. Jiang Xudong, for his unwavering guidance, invaluable advice, and constant encouragement throughout this research journey. Your expertise and insight have been pivotal in shaping both this work and my growth as a scholar. Thank you!

Besides, my sincere thanks also go to the members of my Thesis Advice Committee and the Panel, Prof. Mao Kezhi and Prof. Chen Lihui, for their valuable time and constructive suggestions. The support from my colleagues and friends at NTU has also been invaluable. I am particularly grateful to my research collaborator Dr. Ding Henghui, for his camaraderie and stimulating discussions. I also extend my great thanks to Dr. Wang Suchen, Dr. Wang Xiaohong, Dr. Xiao Xiao, Dr. Huang Ling, Dr. Liu Chang, Mr. Zheng Weihua, Mr. You Jingbo, Mr. Ding Yang, Ms. Tao Xiyan for all the times we shared. Your support and friendship made my PhD journey a memorable experience.

I must express my profound gratitude to my my parents, for their unwavering love, understanding, and support. Your consistent support, especially during challenging times when the world faced unprecedented crises and isolation, has been a source of unending strength.

This thesis stands as a landmark in my personal and academic life, a testament to the collective effort and camaraderie of everyone who has been part of this journey. Thank you all.

Liu Chang, June 2023

Abstract

In recent years, many remarkable achievements have been made in the field of deep machine learning in various data modalities, such as image processing and natural language comprehension. Based on the good performance of deep neural networks in single modalities, multi-modal tasks, which integrate data from different modal domains, are becoming emerging research topics. Among the complex integrated tasks, one particularly challenging and important task is Referring Expression Segmentation (RES), which aims to generate a segmentation mask for a target object in a given image as described by a given natural language query expression, involving both computer vision and natural language processing. This thesis addresses the problem of RES from multiple angles to investigate the topic of this complex multi-modal task.

Firstly, we propose an efficient, instance-specific framework that optimizes the traditional CNN-RNN pipeline. Traditional RES methods usually either use an FCN-like network that directly generates the segmentation mask from the image or first extract all instances using a standalone network and then select the target from targets. We combine the strengths of both kinds of methods and propose a novel framework that can analyze the relationship among instances while maintaining the efficiency of the FCN-like network.

Secondly, we employ an attention-based network to model long-range dependencies in both image and language modalities. In CNN networks, the large receptive field is achieved by stacking multiple small-kernel convolutional layers, which is indirect and lacks efficiency when exchanging long-distance features. From this point, we utilize the Transformer-based network that can model long-range dependencies in a more efficient way. Next, based on this work, we find that the generic attention mechanism used in the classic Transformer is designed for processing single-modal data. We further enhance the mechanism of generic attention with feature-fusing capabilities, achieving denser feature fusion.

Lastly, to accommodate multi-object and no-object expressions, we introduce a novel task called Generalized Referring Expression Segmentation (GRES). To facilitate research in this field, we also construct a large-scale dataset for GRES and design a baseline method, namely ReLA. The proposed method implicitly divides the image into regions and explicitly analyzes the relationship among them, achieving state-of-the-art performance on both RES and GRES datasets.

Our proposed approach advances the state-of-the-art in referring segmentation, and further generalizes the conventional RES to Generalized RES, providing new insights, methods and topics for further research in this field.

Contents

List of Figures

List of Tables

Abbreviations

CV	Computer Vision
NLP	Natural Language Processing
RES	Referring Expression Segmentation
REC	Referring Expression Comprehension
GRES	Generalized Referring Expression Segmentation
RVOS	Referring Video Object Segmentation
GT	Ground-Truth
IoU	Intersection-over-Union
Pr@X	Precision with thresholds
DNN	Deep Neural Network
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
FFN	feed-forward network
LSTM	Long-Short Term Memory
COCO	Common Objects in Context (datasets)
FCN	Fully Convolutional Network
FPN	Feature Pyramid Network
R-CNN	Region Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformer
DETR	End-to-End Object Detection with Transformers
ViT	Vision Transformer
CLIP	Contrastive Language-Image Pre-Training

GPT	Generative Pre-trained Transformer
ISF	Instance-Specific Feature
FPM	Feature propagation Module
RAM	Relationship Analysis Module
IEM	Instance Extraction Module
RM	Refinement Module
VLT	Vision-Language Transformer
QGM	Query Generation Module
QBM	Query Balance Module
MCL	Masked Contrastive Learning
SDF	Spatial Dynamic Fusion
SISO	Same Image, Same Object
SIDO	Same Image, Different Object
DI	Different Image
VAL	Vision-Attended Language features
LAV	Language-Attended Vision feature
M^3Att	Multi-Modal Mutual Attention
M^3Dec	Multi-Modal Mutual Decoder
IMI	Iterative Multimodal Interaction
LFR	Language Feature Reconstruction
ReLA	ReLAtionship modeling
RIA	Region-Image Cross Attention
RLA	Region-Language Cross Attention
N-acc.	No-target accuracy
T-acc.	Target accuracy
gIoU	generalized IoU
cIoU	cumulative IoU

Chapter 1

Introduction

1.1 Overview and Backgrounds

Deep machine learning is a subset of artificial intelligence that focuses on training algorithms to hierarchically learn complex data representations from a number of samples, starting from shallow to deep [6, 7]. By utilizing neural networks with multiple layers, it enables computers to process and identify complex patterns. Its applications span many domains, including image classification [8–11], semantic segmentation [12, 13], natural language processing [14], video processing [15], *etc.*. In recent years, the research community has made remarkable progress in many specific tasks using data from different modalities, such as semantic segmentation and object detection in images, machine translation and sentence classification in natural language processing. We have witnessed the practical application of these tasks in various industries. However, the extensive exploration of deep machine learning in more complex *multi-modal tasks* that involve the integration or combination of two or more data modalities into a single model, *e.g.* combining computer vision (CV) and natural language processing (NLP), is still an ongoing challenge. There are numerous significant challenges in this field, including effectively fusing information data from different modalities, designing a robust and efficient network architecture, and modeling the long-range dependencies of data, and so on.

One representative task of this kind is *Referring Expression Segmentation* (RES), or *referring segmentation* for short. Figure 1.1 gives a brief introduction of this task. In referring segmentation, given an image and a natural language expression that

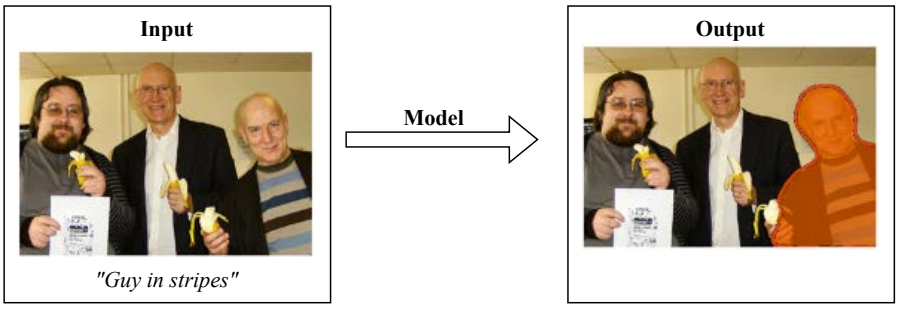


FIGURE 1.1: **A demonstration of Referring Expression Segmentation (RES).** Given an image and an language expression, the model is expected to identify and locate the target object, and output a segmentation mask for it.

describes a specific object in the image, the goal is to identify and locate the target object by generating a segmentation mask for it. This task naturally combines two of the most important data modalities in data science: vision and language. Besides, this complex task also implicitly involves many single modal sub-tasks. For example, the model needs to analyze the image to gather information, such as the semantic class of candidate objects and generating segmentation masks, similar to the task of semantic segmentation; and also needs to understand the language expression and locate the target object, which involves language comprehension and object detection. Therefore, RES serves as an excellent case for investigating the challenges and opportunities in the field of complex multi-modal deep learning tasks.

In this thesis, our main focus will be on the task of referring segmentation. This task requires the development of an integrated approach that combines both computer vision and natural language processing techniques. Our objective is to propose and evaluate novel deep learning architectures and methodologies to create a robust and efficient multi-modal framework, and meanwhile enhance and generalize the task of referring segmentation to maximize its practical value. Our frameworks should effectively address the challenges associated with referring segmentation and also have the potential to expand and generalize to other multi-modal tasks. Through rigorous experimentation and analysis, our goal is to bridge the gap between single-modal and multi-modal deep learning tasks and ultimately pave the way for further advancements in the field of artificial intelligence.

1.2 Challenges and Proposed Methods

Referring Expression Segmentation (RES) is a complex task that requires the integration of image processing and natural language understanding capabilities into the model. This task presents several challenges that need to be addressed, such as the need for a robust method that can understand the language expression and establish connections between the language expression and the image. In this thesis, we concentrate on addressing four representative challenges by developing appropriate models. The following list outlines the challenges we focus on and the corresponding models we design to tackle them:

1. For the problem of referring segmentation, there are typically two types of existing CNN methods: *one-stage methods* and *two-stage methods*. Most current works employ *one-stage methods* [16–18], which directly fuse visual and linguistic features, conducting pixel-level classification on the combined features to produce segmentation masks. These methods are straightforward and efficient; however, their referring capability is limited due to their instance-agnostic nature and difficulty in analyzing relationships among instances in the image. In contrast, *two-stage methods* initially detect and generate masks for all instances as proposals using an off-the-shelf instance segmentation model and subsequently select the best match among them. These methods directly model interactions among instances and can generate high-quality masks. However, in this type of approach, linguistic information is only utilized in the matching stage for selecting existing instance masks, without influencing the segmentation stage. As a result, two-stage methods are heavily dependent on the off-the-shelf instance segmentation network, and their performance is constrained by the candidate instances.

   From this point, we propose a novel framework that simultaneously detects the target-of-interest via feature propagation and generates a fine-grained segmentation mask. In our framework, each instance is represented by an Instance-Specific Feature, and the target-of-referring is identified by exchanging information among all Instance-Specific Features using our proposed Feature Propagation Module. Our instance-aware approach learns the relationship among all objects, which helps to better locate the target-of-interest

compared to one-stage methods. Comparing to two-stage methods, our approach collaboratively and interactively utilizes both vision and language information for synchronous identification and segmentation.

2. In the previous work, we tried to optimize the usage of a CNN architecture. For referring segmentation, the query expression typically indicates the target object by describing its relationship with others. Therefore, to find the target object among all instances in the image, the model must have a holistic understanding of the whole image, especially considering the long-range dependencies inside the image. However, achieving this with conventional CNN-based methods is challenging due to their small receptive fields. Additionally, in practice, there are usually many different ways to understand the emphasis of a language expression. However, most existing methods do not consider this point, which makes it difficult to cope with the randomness and huge diversity of language expressions.

   To address these challenges, we propose a different approach. Instead of using the conventional CNN pipeline that treats referring segmentation as a pixel-level classification problem similar to semantic segmentation, we reformulate the task as an attention problem: finding the region in the image where the query language expression receives the most attention. We introduce transformers and multi-head attention to build a network that "queries" the image using the language expression. Furthermore, we propose a Query Generation Module that produces multiple sets of queries with different attention weights, representing diversified comprehensions of the language expression from different aspects. Moreover, to determine the best way to utilize these diversified comprehensions based on visual clues, we propose a Query Balance Module to adaptively select the output features of these queries for better mask generation.

3. In the previous work, we utilized the generic dot-product attention mechanism to implement the Transformer for referring segmentation. However, the generic attention mechanism only uses the language input for attention weight calculation, which limits the effective fusion of language features in its output. As a result, the output feature is dominated by visual information, hindering the model's comprehensive understanding of the multi-modal

information and introducing uncertainty for the subsequent mask decoder in extracting the output mask.

To address this issue, we propose Multi-Modal Mutual Attention (M^3Att) and Multi-Modal Mutual Decoder (M^3Dec) that better fuse information from the two input modalities. Based on M^3Dec, we further propose Iterative Multi-modal Interaction (IMI) to allow continuous and in-depth interactions between language and vision features. Furthermore, we introduce Language Feature Reconstruction (LFR) to prevent the language information from being lost or distorted in the extracted feature.

4. The existing classic datasets and methodologies for RES are primarily geared toward single-target expressions, in which one expression is used to refer to one target object, while expressions that do not have a valid target and multi-target expressions are not accommodated. This limitation hinders the practical utility of RES in real-world scenarios. Specifically, the lack of no-target expressions that do not have a target implies that the operation of existing RES methods remains undefined in instances where the target object is not present in the input image, necessitating that the input expression correspond to an object in the image to prevent complications. Additionally, the failure to support multi-target expressions requires multiple individual inputs for searching multiple target instances, leading to inefficiencies and impracticalities in real-world applications.

   To this end, we introduce a new benchmark called Generalized Referring Expression Segmentation (GRES), which extends the classic RES to allow expressions to refer to an arbitrary number of target objects. Towards this goal, we construct the first large-scale GRES dataset called gRefCOCO that contains multi-target, no-target, and single-target expressions. GRES and gRefCOCO are designed to be well-compatible with RES, facilitating extensive experiments to study the performance gap of the existing RES methods on the GRES task. In the experimental study, we find that one challenge of GRES is complex relationship modeling. Based on this finding, we propose a region-based GRES baseline called ReLA that adaptively divides the image into regions with sub-instance clues and explicitly models the region-region and region-language dependencies.

1.3 Contributions and the Outline of the Thesis

The primary focus of this thesis is to address the multi-modal complex problem Referring Expression Segmentation (RES). Utilizing deep machine learning techniques, we aim to develop models that can effectively segment the target-of-interest from an image using a referring expression. Additionally, we propose an expansion of the classic RES problem, known as Generalized Referring Expression Segmentation (GRES), which allows expressions to refer to multiple target objects. The contributions of this thesis are as follows:

- **Chapter 2** provides a comprehensive literature review and introduces the preliminary knowledge relevant to this thesis. The chapter also gives a brief overview of the related tasks, methods, and datasets, providing a solid foundation for the subsequent chapters.
- **Chapter 3** introduces a novel framework that simultaneously detects the target-of-interest and generates a fine-grained segmentation mask. The proposed instance-aware approach learns the relationship among all objects, leading to better localization of the target-of-interest compared to one-stage methods. The approach collaboratively and interactively utilizes both vision and language information for synchronous identification and segmentation. The chapter is mainly based on our publication in [19].
- **Chapter 4** proposes the utilization of the Transformer model to address the referring segmentation problem. The chapter reformulates the problem from the perspective of attention and introduces a transformer-based model called VLT, along with a Query Generation Module (QGM) to facilitate the model. The QGM generates vectors that can query the input image using the input language expression in different ways of understanding, providing a holistic understanding of the multi-modal information. The chapter is mainly based on our publications in [20] and [21].
- **Chapter 5** focuses on optimizing the generic attention mechanism in the regular Transformer model to better suit multi-modal tasks. The chapter proposes Multi-Modal Mutual Attention (M^3Att) and Multi-Modal Mutual Decoder (M^3Dec) to fuse information from the two input modalities more effectively. The M^3Attmodels two attention pathways simultaneously, offering

a dense and efficient approach to fuse features from the two modalities. The chapter is mainly based on our publication in [22].

- **Chapter 6** extends the classic referring segmentation task to a more generalized setting called Generalized Referring Expression Segmentation (GRES), which allows expressions to refer to any number of target objects. We also introduce the first large-scale GRES dataset, gRefCOCO, which includes multi-target, no-target, and single-target expressions. Based on this dataset, the chapter proposes a region-based GRES baseline called ReLA, which adaptively divides the image into regions with sub-instance clues and explicitly models the region-region and region-language dependencies. The chapter is mainly based on our publication in [23].

- **Chapter 7** concludes this thesis and discusses the remaining challenges in the field of RES and GRES. The chapter also explores potential future research directions, providing insights into how the proposed methods can be further improved or extended.

Chapter 2

Literature Review

This chapter provides an extensive review of the Referring Expression Segmentation (RES) literature, covering various research topics. To provide the necessary context, this chapter also introduces background and closely related information on Deep Neural Network (DNN), semantic segmentation, object detection, and Referring Expression Comprehension (REC).

2.1 Deep Neural Network (DNN)

Deep Neural Network (DNN) is a type of artificial neural network that uses multiple layers to learn the features of data. DNNs have been successfully applied to various fields, including computer vision, speech recognition, and natural language processing. In recent years, two specific types of DNNs have emerged as leading architectures for many tasks including image processing and multi-modal information processing: Convolutional Neural Network (CNN) and Attention-based Network. CNN have been proposed eariler, and are commonly used for image and video processing tasks, *e.g.*, semantic segmentation, while Transformers have initially shown remarkable success in Natural Language Processing (NLP) tasks and are also showing great potential in computer vision tasks.

2.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is one of the most well-known DNN architectures, and has been widely used in computer vision tasks, such as image classification [8–11], semantic segmentation [12, 13], instance segmentation [1, 24].

Prior to the deep learning era, there have already existed works on the topic of convolutional networks, for example LeNet[25]. However, due to many limitations like hardwares, the performance of these networks are not satisfactory at that moment. AlexNet [26] is usually considered to be the first breakthrough of CNN. It consists of 5 stacked convolutional layers that are distributed in 2 GPUs, achieving the best perofrmance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [27]. The remarkable success of AlexNet can be attributed to two main factors: the use of more layers to achieve a deeper network depth, and the use of GPU to accelerate the training process. As a result of this understanding, numerous CNN architectures have emerged afterwards. Researchers and practitioners have developed a wide range of CNN models, each with its own unique architectural designs. For example, VGG [28] further increased the network depth to 19, achieving better performance on the ImageNet dataset. U-Net [29] is a CNN architecture designed for biomedical image segmentation, which consists of a contracting path and an expansive path. Another major breakthrough of CNN is ResNet [9], which won the ILSVRC 2015. ResNet is a CNN architecture that uses "residual blocks" to solve the problem of vanishing gradient. The principle of residue is that rather than simply stack convolution layers together, it use the identity mapping to let the network learn the residual of the input. With residual blocks, ResNet can achieve a depth of 152 layers, which is much deeper than previous CNN architectures.

Furthermore, specialized convolutional neural networks (CNNs) such as MobileNet [30–32] and ShuffleNet [33, 34] have been developed for mobile devices. These CNNs are particularly noteworthy due to their lightweight nature, offering improved efficiency compared to conventional CNNs. MobileNet achieves this by utilizing depthwise separable convolution to decrease the parameter count, and ShuffleNet employs group convolution and channel shuffle techniques to effectively reduce computational costs. Most recently though, attention-based network achieves better performance with more network parameters, there are still many works that are based on CNNs, and achieve competitive performance with higher speeds, *e.g.*

ConvNext [35]. CNN is still an essential architecture for computer vision research and application.

2.1.2 Attention-based Network

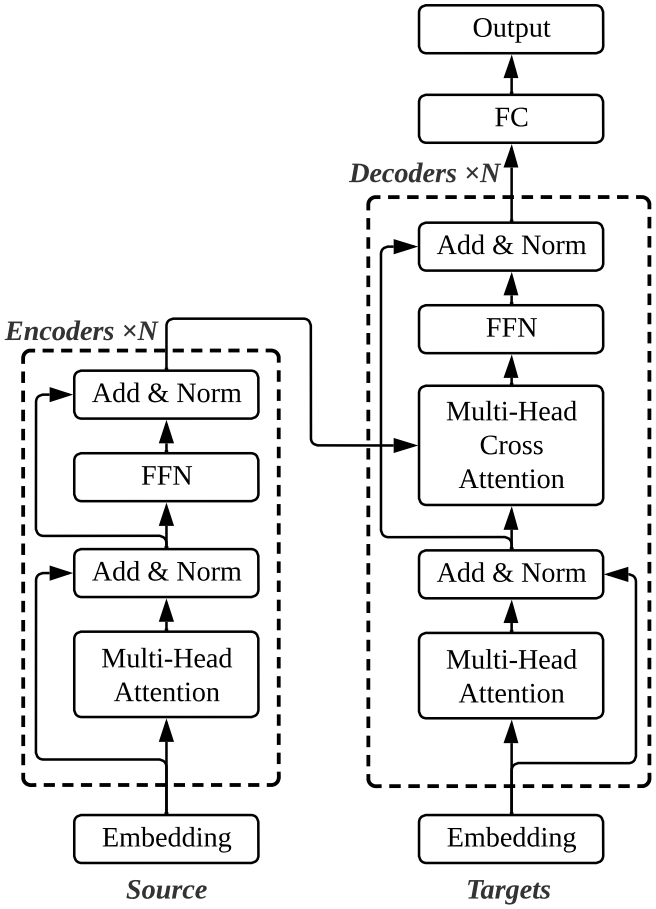


FIGURE 2.1: **A simplified architecture of the Transformer.** The Transformer consists of a stack of "encoder" and "decoder" layers. FC: Fully-connected layer. FFN: Feed-forward Network that consists multiple FC layers.

The first fully attention-based network is defined by Vaswani *et al.* [36]. They proposed a novel architecture called *Transformer*. Unlike CNN which initially proposed for computer vision tasks, Transformer is initially proposed for sequence-to-sequence tasks, such as machine translation. As shown in Figure 2.1, a typical Transformer network consists a stack of "encoder" and "decoder" layers, but the core of both of them is the attention mechanism. Regular Transformer network utilizes dot-product attention mechanism, which is used to capture the relationship

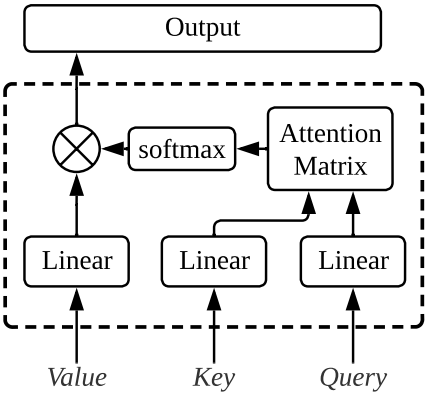


FIGURE 2.2: **A simplified diagram of the dot-product attention.**

between different elements in the input sequence. The computation flow of the dot-product attention is shown in Figure 2.2. It takes three inputs: the query Q, the key K, and the value V. The dot-product attention first computes the similarity between the query and the key, and then uses the similarity as the weight to compute the weighted sum of the value. The dot-product attention can be formulated as [36]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.1}$$

where d_k is the dimension of the key K, and $\sqrt{d_k}$ serves as a scaling factor. The Transformer network further employ a multi-head mechanism on the regular attention to attend to different information in different heads [36]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{2.2}$$

2.2 DNN in Computer Vision

Though the basic version of the aforementioned networks are usually trained for some basic tasks like image classification, the features learned by these networks are usually not task-specific. After modifying and fine-tuning, a network that is pretrained on one task can also be used for many other tasks. Therefore, they can be seen as "base" of many tasks. In this section, we introduce the most commonly

used networks in computer vision (CV), for semantic segmentation and object detection.

2.2.1 Semantic Segmentation

Semantic Segmentation is one of the most important tasks in computer vision. Given an image, the aim of semantic segmentation is to assign a semantic label to each pixel in an image. In other words, it is a pixel-level classification task. It has applications in many areas, such as autonomous driving, medical image analysis, and video surveillance. The Fully Convolutional Network (FCN) [13] marked a critical milestone in the field. Developed on top of the VGG-16 architecture. As its name indicates, FCN is fully-convolution based and is able to perform end-to-end training and inference. Following FCN, there are a lot of works have been proposed. PSPNet [37] use feature pyramids to achieve a feature aggregation among local features and global features, and deeplab introduces dilated convolution into semantic segmentation to enlarge the receptive field of the network. Most recently, attention-based networks also show impressive performance on semantic segmentation. For example, networks based on Swin-Transformer [38] surpass its previous semantic segmentation methods with large margins on ADE20k dataset. In referring segmentation, many works utilize the pipeline of FCN and formulate the problem as a pixel-level classification task, similar with semantic segmentation.

2.2.2 Object Detection and Instance Segmentation

Object Detection is a task of detecting and classifing all objects inside an image by outputting a bounding box for each detected object. In this area, Region Based Convolution Neural Network (R-CNN) series is one of the most classic series of models. The core of R-CNN is to extract "region proposals". The first model of R-CNN is proposed by Girshick *et al.* [39], which is a two-stage model. It first extracts region proposals using selective search, and then extracts features from each region proposal using CNN. These region features are further processed to generate the detection and classification results. Following R-CNN, Fast R-CNN [40] and Faster R-CNN [41] are proposed to improve the speed and performance of the original R-CNN model. Fast R-CNN proposes the region of interest (RoI) pooling layer to

extract a fixed-length features from the region proposals, and Faster R-CNN uses a region proposal network (RPN) to generate Region of Interests (RoIs). YOLO [42–44] is another model series for object detection. The main idea of YOLO is to divide the image into grids, and find objects within these grids. Compared with R-CNN series, YOLO is a one-stage model, which is much faster than R-CNN series. Another closely related task is mask segmentation. Similar with object detection, it also needs to detect all instances that appear in an image, but requires the model to output segmentation masks for each detected instances instead of bounding boxes. One well-known work is Mask R-CNN [24], which is proposed to extend Faster R-CNN to instance segmentation. In referring segmentation, some works uses instance segmentation methods for generating candidate proposals.

2.3 DNN in Natural Language and Multi-Modal Processing

Besides computer vision (CV) which focuses on processing images or videos, Natural Language Processing (NLP) is another important field of artificial intelligence. It focuses on processing natural language, such as English, Chinese, and French. Eariler works usually utilizes the Recurrent Neural Network (RNN), such as Long-Short Term Memory (LSTM), to process the language. RNNs uses a network to process the input sequence one by one, and the output of the previous step is used as the input of the next step, which can handle sequences of arbitrary length and is suitable for processing natural language. Recently, with the development of deep learning, the attention-based methods, *i.e.* Transformers, have shown great potential in NLP tasks. BERT [45] is one of the most influential Transformer-based models, which is pre-trained on a large corpus and can be fine-tuned for various NLP tasks, such as machine translation, semantic role labeling and sentence classification. The BERT model utilizes the regular unmodified Transformer architecture, while its training involves two large text corpora: BookCorpus and English Wikipedia, encompassing over 3 billion words. During training, BERT's objective is to predict randomly masked words within the input sequence.

The Generative Pre-trained Transformer (GPT) family is another series of famous large-scale pretrained language model. GPT-1 [46] and GPT-2 [47] are the earilest

models in this series. Being trained by 40GB text data, GPT-2 proves the potential of unsupervised learning of language models. GPT-3[48] utilizes significantly larger training datasets that contains 570GB text data and larger model sizes of up to 175 billion parameters, and produces remarkable generative results [48]. Further, its improved version, GPT-3.5 and GPT-4 (or ChatGPT), is trained on even more data. Tests shows that GPT-3.5 and GPT-4 can pass many academic and professional exams, such as Graduate Record Examination (GRE) and Medical Knowledge Self-Assessment Program [49].

Most recently, multi-modal processing, which involves more than one data modalities in one single model, has become a hot topic. One of the most popular areas is Vision+Language tasks which combines both CV and NLP. Contrastive Language-Image Pre-Training (CLIP) [50] is a representative work that utilizes self-supervision techniques to train a model on 400 million image-text pairs, and achieve state-of-the-art on over many tasks such as action recognition in videos, geo-localization and vision questioning-answering. The aforementioned GPT-4 also have general multi-modalities information processing ability.

2.4 Referring Expression Segmentation (RES)

2.4.1 Problem Definition

Referring Expression Segmentation (RES), or referring segmentation, is one of the most fundamental while challenging multi-modal tasks, involving both language and vision information. Given an image, a referring expressions is an expression written in natural language that describes one certain object in this image. The goal of the referring expression segmentation is to comprehensively understand the image, the expression, and the relationship between them, and generate a segmentation mask for the target object. The input language expression is called a "referring expression" of the input image. In conventional RES and related tasks, the content of the referring expression is unconstrained, except for that it should be written in natural language, and should be umambiguously point to the target object. The RES is naturally a multi-modal task, involving both language and vision information.

2.4.2 Previous Works

Being defined by Hu *et al.* [16], Referring Expression *Segmentation* (RES) comes from a similar task, Referring Expression *Comprehension* (REC) [51–57] that outputs a bounding box for the target. Hu *et al.* [16] concatenates the linguistic features extracted by Long Short-Term Memory (LSTM) networks and the visual features extracted by Convolutional Neural Networks (CNN). Then, the fused vision-language features are inputted to a fully convolutional network (FCN) [13] to generate the target segmentation mask. In [58], in order to better utilize the information of each word in the language expression, Liu *et al.* propose a multimodal LSTM (mLSTM), which models each word in every recurrent stage to fuse the word feature with vision features. Li *et al.* [59] utilize features from different levels in the backbone progressively, which further improves the performance. To better utilize the language information, Edgar *et al.* [60] propose a method that uses the features of each word in the language expression when extracting language features, not just the final state of the RNN. Chen *et al.* [61] employ a caption generation network to produce a caption sentence that describes the target object, and enforce the caption to be consistent with the input expression. In [17], Luo *et al.* propose a multi-task framework to jointly learn referring expression comprehension and segmentation. They build a network that contains a referring expression comprehension branch and a referring expression segmentation branch, each of which can reinforce the other during training. Jing *et al.* [62] decouple the referring segmentation to localization and segmentation and propose a Locate-Then-Segment (LTS) scheme to locate the target object first and then generate a fine-grained segmentation mask. Feng *et al.* [63] propose to utilize the language feature earlier in the encoder stage. Hui *et al.* [64] introduce a linguistic structure-guided context modeling to analyze the linguistic structure for better language understanding. Yang *et al.* [65] propose a Bottom-Up Shift (BUS) to progressively locate the target object with hierarchical reasoning on the given expression.

With the introduction of attention-based methods [36, 66], researchers have found that the attention mechanism is suitable for the formulation of referring segmentation. For example, Ye *et al.* propose the Cross-Modal Self-Attention (CMSA) model [18] to dynamically find the most important words in the language sentence and the informative image region. Hu *et al.* [67] propose a bi-directional attention module to further utilize the features of words. Most of these works are built

on FCN-like networks and only use the attention as auxiliary modules. Our concurrent work MDETR [68] employs DETR [1] to build an end-to-end modulated detector and reason jointly over language and image. After the proposed VLT [69], transformer-based referring segmentation architectures receive more attention [70–74]. MaIL [70] follows the transformer architecture ViLT [75] and utilizes instance mask predicted by Mask R-CNN [24] as additional input. Yang *et al.* [71] propose Language-Aware Vision Transformer (LAVT) to conduct multi-modal fusion at intermediate levels of the network. CRIS [72] employs CLIP [50] pretrained on 400M image text pairs and transfers CLIP from text-to-image matching to text-to-pixel matching.

2.4.3 Framework Taxonomy

Most of the previous RES works can be roughly categorized into two categories: one-stage methods and two-stage methods.

One-stage methods, or top-down methods [16, 17, 58–65] are more like semantic segmentation methods, which directly outputs the desired target mask using a segmentation network. As shown in Figure 2.3, the architecture of one-stage methods is very intuitive and simple. Firstly, the model extracts the visual features of the input image and the language features of the input expression using two backbones, then fuse them to generate the target mask. Next, a Fully Convolutional Network (FCN) [13] like network is used to generate the target mask. Some recent works [17, 62, 63] propose to inject the language information into the image backbone or FCN to achieve a denser fusion of multi-modal features, but the basic architecture of them are still consistent. The one-stage methods are simple and efficient, but they lack the intepretebility of the intermediate features and the ability to reason over the language and image.

Two-stage methods, or bottom-up methods [19, 76] need to employ an extra standalone out-of-the-box instance segmentation network such as Mask R-CNN [24]. As shown in Figure 2.4, the first stage of the two-stage methods is to generate a mask for each instance that appears in the image, as instance proposals, or candidates. The aforementioned instance segmentation network is used

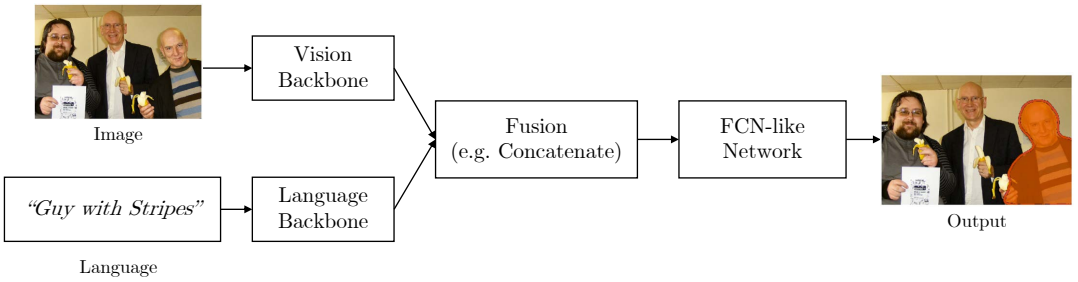


FIGURE 2.3: **One stage methods** utilizes a FCN-like network on the fused features.

in this step. Then, the model extracts the language feature of the input referring expression, and give a "matching" score of each instance candidate indicating how likely it is the target object. Finally, the instance with the highest matching score is selected as the target instance, and its corresponding mask generated by the instance segmentation network is outputted. As two stage networks can explicitly get information of each instance in the image, it is easier and more convenient for them to analyze the relationship among instances. However, the two-stage methods are more complex and time-consuming, and the performance of the instance segmentation network will directly affect the performance of the referring segmentation network.

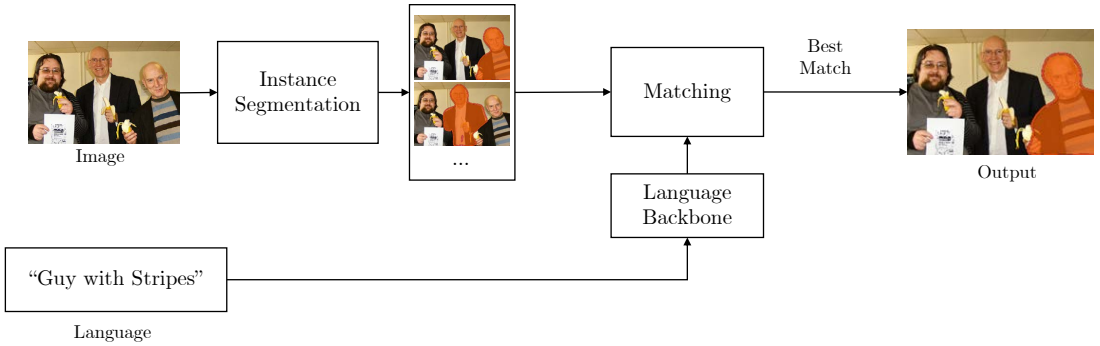


FIGURE 2.4: **Two-stage methods** need to employ a standalone instance segmentation network.

2.4.4 Datasets and Benchmarks

ReferIt [2] is the first large-scale dataset for referring segmentation. It contains 130,525 referring expressions for 96,654 objects in 19,894 images from ImageCLEF

dataset [77]. ReferIt introduces a game-based annotation methods for collecting the referring expressions, called ReferItGame. In this game, annotators are divide into two groups. One group is called the *speaker*, who is given an image and a target object, and is asked to describe the target object with a language expression. The other group is called the *listener*, who is given the same image but without the target object, and is asked to find the target object according to the language expression. If the listener can find the target object, this expression is considered to be a valid referring expression and both players get scores as rewards. The ReferItGame is designed to simulate the real-world referring scenarios, where the speaker and listener are separated and cannot communicate with each other. This method is also adopted by many later referring segmentation datasets such as RefCOCO and G-Ref.

RefCOCO & RefCOCO+ [3] are two of the largest and well-known datasets for referring segmentation. They are also called UNC & UNC+ datasets in some literature. 142,209 referring language expressions describing 50,000 objects in 19,992 images are collected in the RefCOCO dataset, and 141,564 referring language expressions for 49,856 objects in 19,992 images are collected in the RefCOCO+ dataset. The difference between two datasets is that the RefCOCO+ restricts the expression ways for the language sentences. For example, descriptions about absolute locations, *e.g.*, *"leftmost"*, are forbidden in the RefCOCO+ dataset. Therefore, RefCOCO+ is usually considered to be harder. As they are based on the famous MS COCO [78] dataset for image source, they are available for both RES and REC tasks.

G-Ref [4, 79] Also called RefCOCOg, it is another well recognized referring segmentation dataset. Like RefCOCO, it is also based on MS COCO [78] dataset. 104,560 referring language expressions for 54,822 objects in 26,711 images are used in G-Ref. Unlike RefCOCO & RefCOCO+, the language usage in the G-Ref is more casual but complex, and the sentence lengths of G-Ref are also longer in average. Notably, G-Ref has two versions: one is called UMD split [79], the other is called Google split [4]. The UMD split has both validation and testing set publicly available, but the Google split only makes its validation set public.

More related datasets Most recently, as referring expression related tasks are getting more and more popular in the research community, there are many new datasets proposed. For example, PhraseCut [5] is a large-scale referring expressions based on the Visual-Genome [80]. It contains 77,262 images and 345,486 expressions, including multi-target expressions. However, its expressions are generated in templates so the language are not as natural as free-form language datasets like RefCOCO. There are more datasets for more visual modalities other than 2D image. For referring expressions in videos, firstly, A2D [81, 82] and JHMDB [83] are designed for understanding actor's actions in videos. Next, Ref-DAVIS [84] extends them into general referring expressions that can refer to any objects in videos. It contains 1,544 expressions in 90 videos in its 2017 version. Later, Refer-Youtube-VOS (RVOS) [85] provides a larger dataset that contains 27,000+ expressions in 3,900+ videos. For 3D objects, ScanRefer [86] provides 51,583 expressions for 11,046 objects in 800 ScanNet [87] scenes. Moreover, PhraseClick [88] proposes to combine referring expression segmentation with interactive segmentation together, and provides a new dataset for this task. These datasets further enlarges the range of applications of referring expression related tasks.

2.5 Referring Expression Comprehension (REC)

Referring Expression Comprehension (REC), or referring comprehension, is a highly relevant task to referring segmentation. Referring comprehension also takes an image and a language expression as inputs and identifies the target object referred by the language expression. However, while referring segmentation aims to output a segmentation mask for the target object, the referring comprehension outputs a grounding box. In this thesis, we will focus more on RES methods, but we also briefly introduce some recent works on REC. Unlike the FCN-like pipeline of referring segmentation, most earlier referring comprehension works are based on the multi-stage pipeline[53–55, 89–91]. Like RES, in these works, often an out-of-the-box instance segmentation network, *e.g.*, Mask R-CNN [24], is first applied to the image and generates a set of instance proposals, regardless of the language input. Next, the candidate proposals are compared with the language expression, to find the best match. For example, Yu *et al.* [76] propose a two-stage method that first extracts all instances in the image using Mask R-CNN [24], then employs

a modular network to match and select the target object from all the instances detected by Mask R-CNN. In recent years, one-stage methods [57, 92, 93] have also been increasingly adopted in the referring comprehension area, *e.g.*, Sadhu *et al.* propose a "Zero-Shot Grounding" network for referring comprehension [94], and Yang *et al.* design a recursive sub-query construction framework to gradually reason between the image and query language [56].

2.6 Chapter Summary

In this chapter, we have introduced the background on Referring Expression Segmentaion (RES). Starting from the preliminary knowledge of Deep Neural Networks (DNN), we introduced two kinds of DNN architectures: Convolutional Neural Networks (CNN) and attention-based networks. Then, we discuss some of most common DNN applications, including semantic segmentation, object detection, instance segmentation and natural language processing. Next, we introduce the concepts of RES, including the problem definition, existing works and taxonomy. We also lists the related datasets and benchmarks for RES, and introduces a highly related task: and Referring Expression Comprehension (REC). Starting from the next chapter, we will introduce our proposed methods for referring segmentation, and our exploration on expanding and generalizing the conventional referring segmentation task.

Chapter 3

Instance-Specific Feature Propagation

3.1 Introduction

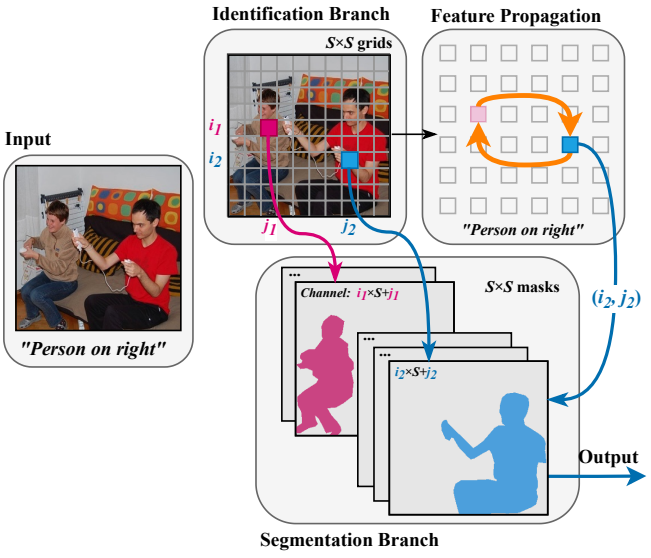


FIGURE 3.1: **Illustration of our method.** We spatially divide the image into grid and assign each instance to a grid box where its center falls in. The model learns an Instance-Specific Feature for each grid as the representation of the corresponding instance, identifies the target and generates the mask simultaneously.

In order to select the target instance from all instances in the image, one of the most challenging problems is to model the interaction among all candidates. For example, given an image of several persons standing and a query expression saying

"the rightmost person", the model must do comparison among all instances of "person" before determining which is the "rightmost". Most of the current works use *one-stage methods* [16–18], which directly fuse the vision and language features together and perform pixel-level classification on the fused features to generate the segmentation mask. This kind of methods are direct and efficient, but their referring capability is limited as they are instance-agnostic. Without explicitly detected instances, it is hard to directly model and represent the "interaction" among them. Thus, this kind of method often does not work as expected in the scenario where the layout of the object is tangled or the input language expression describes a complicated relationship among different instances.

To alleviate this issue, MattNet [76] proposes *two-stage methods*, which first detect and generate masks for all instances as proposals with an off-the-shelf instance segmentation method, *e.g.*, Mask R-CNN [24], then select the best match among them. They directly model the interaction among instances and can generate high-quality masks with the help of the instance segmentation network. However, in this kind of method, the language information is only used in the matching stage for selecting the existing instance masks, and has no influence on the segmentation stage. Therefore, two-stage methods rely heavily on the off-the-shelf instance segmentation network and their performance is limited by the candidate instances.

In this chapter, we aim to achieve two goals: first, we would like the network to be instance-aware so that to inspect and model the interaction among instances directly, and second, to be an integrated one-stage method where the instance selection process and mask generating process is simultaneous and collaborative. Inspired by previous works [42–44, 95], in this work, we use grid boxes to represent instances and identify the target by building interaction among them. We employ both vision and language features to generate segmentation masks and identify the right one simultaneously.

In our work, as shown in Figure 3.1, we have an Identification Branch that evenly divides the image spatially into $S \times S$ grids. We let each gird represent the specific instance whose center is located at it. Accordingly, we generate a feature vector for each grid, called an Instance-Specific Feature (ISF). One ISF contains all information about the specific instance, such as size, texture, and shape. ISFs are spatially organized with regard to the spatial location of their corresponding instances into a feature map. This enables the interaction among instances.

To model the interactions among instances, some two-stage methods like MattNet [76] first select a few instances using pre-defined rules, *e.g.*, 5 instances from the same semantic category, and then model the relationship within selected items. Although such hand-crafted and rule-based selection is reasonable for some cases, it sometimes may not well fit the referring expression. We propose a method that globally models the relationship among all instances. To this end, a bi-directional propagation process is employed to model the "comparison" relationship among different instances. During the propagation, the ISF of each instance is exchanged with all other instances, and finally, the target instance is highlighted.

Simultaneously, we have a Segmentation Branch, which generates a mask for every grid. We inject the language features in the mask generation process to enhance the features of the target object. Finally, the mask corresponding to the target grid location is selected as the output mask. The mask generation part can work alone as a one-stage method with limited performance, and our experiment show that its performance can be greatly enhanced with the awareness of other instances and the interaction modeling among instances. Furthermore, we propose a refinement module that refines the coarse mask to generate a more detailed mask prediction.

In summary, the major contribution or our work can be listed as follows:

- We propose a novel referring segmentation framework that generates segmentation masks and identifies the target-of-interest simultaneously and collaboratively by modeling the relationship among all explicitly-defined instances in the image.
- We propose a propagation based Relationship Analysis Module (RAM) and a Refinement Module to enhance the performance of the framework for both identification and segmentation.
- Experiments show that the proposed approach outperforms other methods and achieves the new start-of-the-art consistently on all three RefCOCO series datasets.

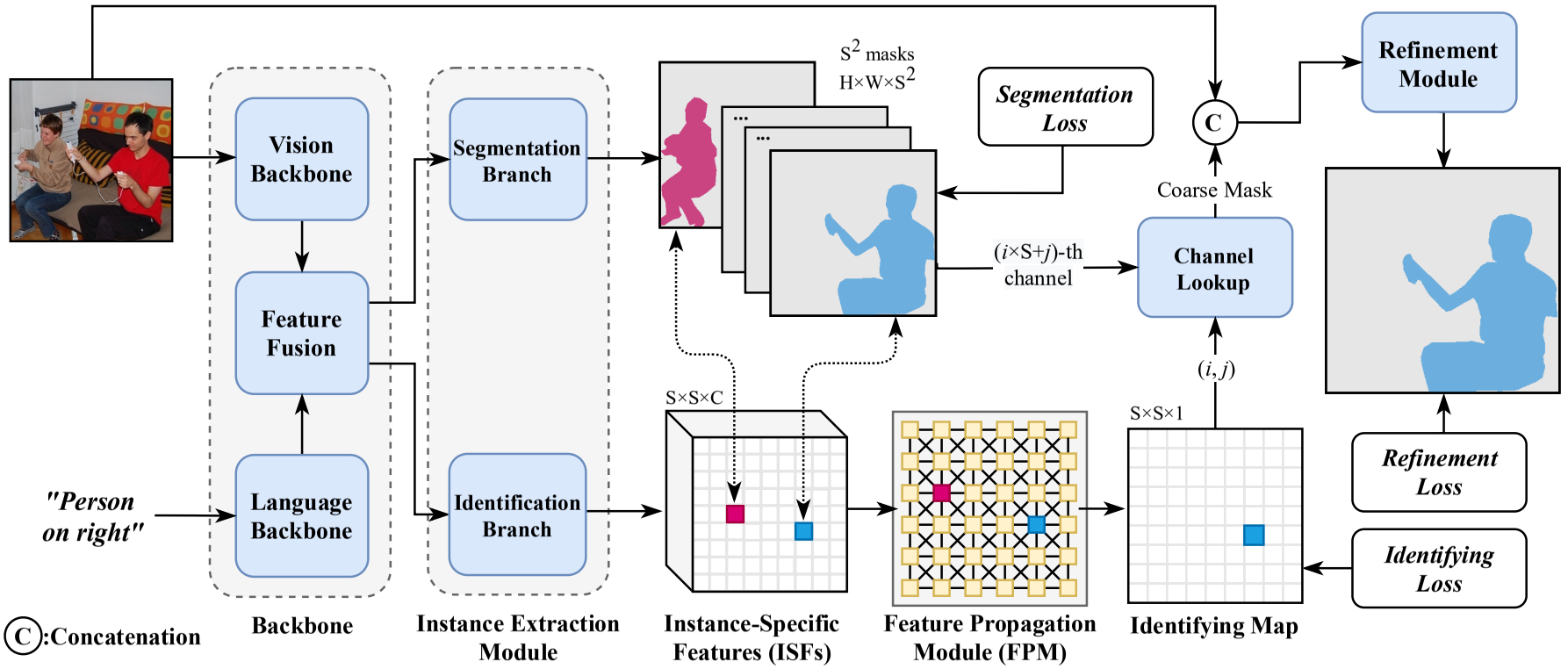


FIGURE 3.2: **Overview of the proposed framework.** The framework contains four main components: Backbone to extract vision and language features and fuse them together; Instance Extraction Module to extract Instance-Specific Features (ISFs) and generate a mask for each grid; Relationship Analyze Module to determine the location of the target in grid; Mask Refinement Module to refine the coarse mask generated by the Segmentation Branch.

3.2 Methodology

An overview of our method is shown in Figure 3.2. A training sample consists of an image $I \in \mathbb{R}^{H \times W \times 3}$, a referring expression $T = \{w_i\}_{i=1...t}$ with the length of t, and a segmentation mask of the target object M.

3.2.1 Backbone

First, the Backbone extracts features from I and T and fuses them to a set of fused vision-language feature F.

Vision feature. In order to enhance the multi-scale performance, the Feature Pyramid Network (FPN) [96] is employed to extract the basic vision features. We utilize features from the last three blocks of the FPN and denote them, from higher resolution to lower resolution, as: F_{vl}, F_{vm}, F_{vs} with sizes of $H_l \times W_l \times C_v, H_m \times W_m \times C_v$ and $H_s \times W_s \times C_v$, respectively. C_v is the number of vision feature channels.

Language feature. We first utilize the GloVE [97] to generate embeddings E for all words in the input referring expression. Then a bi-directional GRU [14] module is applied on E to extract the language features. Suppose the hidden states of all words are $F_w \in \mathbb{R}^{t \times C_l}$, where C_l is the number of language feature channels. A set of attention weights $W_w \in \mathbb{R}^{t \times 1}$ is derived using a self-attention module [11] to measure the importance of each word in the expression. The final linguistic feature F_t is the weighted sum of all hidden state outputs of the GRU, *i.e.*, $F_t = {W_w}^T F_w$.

Feature fusion. Next, we generate the fused vision and language features F. We use a dense multiplication operation for fusion. We first apply a linear layer on language feature F_t and a 1×1 convolutional layer on each of the 3 FPN layers to transform them to the same channel depth, then do the element-wise multiplication. Let the $f_v^{i,j} \in \mathbb{R}^{1 \times C_v}$ denote the feature vector at spatial position (i, j) of F_{vl}, F_{vm} and F_{vs}. The fused feature at the pixel (i, j), $f^{i,j}$ is derived as:

$$f^{i,j} = (f_v^{i,j} W_v) * (F_t W_t) \tag{3.1}$$

where $W_v \in \mathbb{R}^{C_v \times C_f}, W_t \in \mathbb{R}^{C_l \times C_f}$ are learnable parameters, C_f is the number of fused feature channels and $*$ denotes element-wise multiplication. The same fusing

process is done on all the three FPN layers F_{vl}, F_{vm}, and F_{vs}. As the length of the referring expression is usually not very long, the computation cost of the process is acceptable, while the operation can densely fuse the language information into the vision features.

In referring segmentation, it is essential to consider the interaction among different objects, including objects of different sizes. Thus, we further fuse feature from all FPN layers together to enable features exchange across FPN layers. We use two pathway directions based on the lateral connection [96] for processing features: a downsampling pathway generating a distilled semantic information output F_{ide} of size $H_s \times W_s$ for the Identification Branch and an upsampling pathway generating a higher resolution feature map F_{seg} of size $H_l \times W_l$ for Segmentation Branch.

3.2.2 Instance Extraction Module

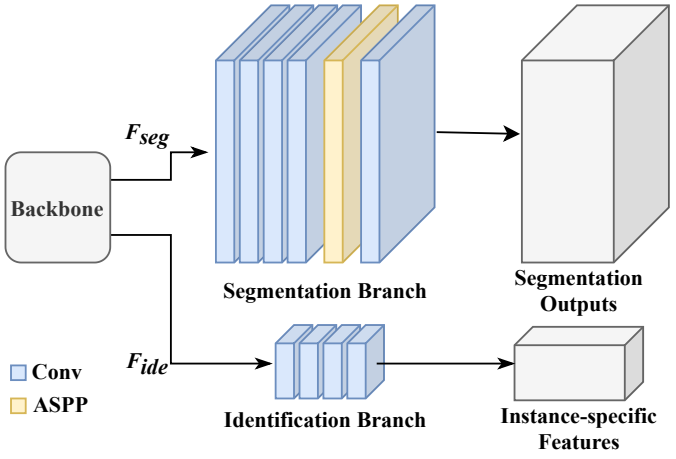


FIGURE 3.3: **Architecture of the Instance Extraction Module.** The module consists of two branches: an Identification Branch outputting the Instance-Specific Feature (ISF) map where each grid represents an instance and a Segmentation Branch generating masks for each instance.

The Instance Extraction Module has two branches: Identification Branch and Segmentation Branch. The structure of this module is concise and direct, as shown in Figure 3.3.

Identification Branch divides the image into grids and link each instance in the image to its nearest gird. Inspired by [42, 95], we divide the image into $S \times S$ grid, and each grid is responsible for detecting a possible object centered at its location. Accordingly, the Identification Branch outputs an Instance-Specific Feature (ISF)

map $F_{ins} \in \mathbb{R}^{S\times S\times C}$, or $F_{ins} = \{f_{ins}^{i,j}\}_{i,j=0,1,...S-1}$, where C is the number of instance feature channels. Since each $f_{ins}^{i,j}$ specifies the characteristics of an object centered at location (i, j), we call each $f_{ins}^{i,j}$ ISF vector.

The branch takes as input F_{ide} from the backbone, and resize its spatial size to $S \times S$ using bilinear interpolation to match the spatial correspondence between feature map and grids. The body is formed by four 3×3 stacked convolutional layers, generating the ISF map F_{ins}, where each position corresponds to one grid in the image.

Segmentation Branch takes F_{seg} as input, and generates one binary mask for each grid, leading to a $H \times W \times S^2$ output. To ensure that each ISF vector is uniquely associated with one specific object, we designate a fixed mapping relationship between the spatial location of the grid and the channel location of the target mask: for the grid at the i-th row and j-th column, the channel number c of its mask will be $i \times S + j$, where $0 \leq i, j \leq S - 1, 0 \leq c \leq S^2 - 1$.

This branch has a concise architecture. Its main body is formed by four 3×3 convolutional layers. Then we use an ASPP [12] module to enhance its multi-scale performance. At last, we use a convolutional layer with S^2 output channels to generate a mask for every grid.

3.2.3 Relationship Analyze Module (RAM)

As mentioned before, given an image, a natural way to point an instance out is by "comparison". To model this relationship, we propose a global modeling method, Relationship Analyze Module (RAM), that lets each instance be aware of all other instances.

The RAM is a recurrent based method, which has four propagation directions: DR↘, DL↙, UR↗, UL↖ (U, D, L, R: Up, Down, Left, Right) like Figure 3.4b. For each pixel in one direction, information about the current pixel and the past pixels are fused together, producing a hidden state feature. Then, this feature is sent to the next pixel, as the information about past pixels.

In detail, firstly, feature of each pixels are initialized with its Instance-Specific Features (ISFs). Since each ISF represents one specific instance in the image, it is

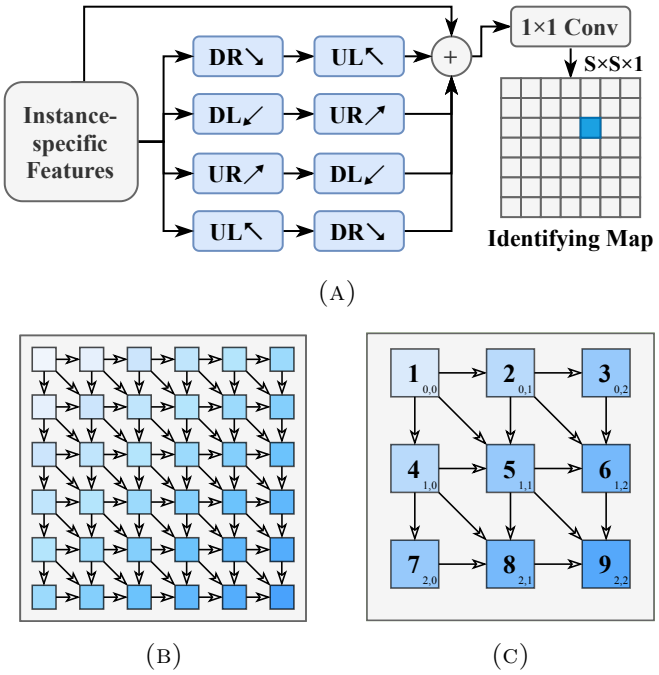


FIGURE 3.4: **The Relationship Analyze Module.** (a): Structure of the four bidirectional paths. The module has four paths, each of which consists of two opposite directions. (b): The detailed propagation process of the DR↘ (Down, Right) direction. (c):The propagation order in the DR↘ direction.

convenient to model the interactions among them. For each direction, the hidden state of one pixel is derived from itself and at most 3 past pixels: row, column, diagonal, like shown in Figure 3.4c. The hidden state of the pixel on i-th row and j-th column $F_{i,j}$ is calculated as:

$$F_{i,j} = W_1 p_{i,j} + \alpha W_2 \sum_{m,n}^{incoming} F_{m,n} \tag{3.2}$$

where $p_{i,j}$ is the ISF of this pixel, $W_1, W_2 \in \mathbb{R}^{C_v \times C_v}$ are learnable weights, α is a constant to control the forget rate. A non-linear activation function is applied on the derived $F_{i,j}$. For example, in the DR↘ direction, when $i = j = 1$:

$$F_{1,1} = W_1 p_{1,1} + \alpha W_2 \left(F_{0,1} + F_{1,0} + F_{0,0}\right) \tag{3.3}$$

In order to model the "comparison" relationship, we combine each direction with its opposite direction as a bidirectional path, as shown in Figure 3.4a. *E.g.*, in the DR↘ - UL↖ direction, assume that there are two instances in the image: A located on the top left corner and B on the bottom right. In the first direction (DR↘),

B gets information about A. Then in the reversed direction (UL↖), A gets a second-order knowledge that contains information of B itself and B's information of A. In such a way, the relationship of "comparison" is modeled.

To maintain the instance correspondence property, we sum up the final state from all paths, as well as the input ISFs together. The result is finally processed by a 1×1 convolutional layer, generating a $S \times S \times 1$ Identifying Map. This map indicates the probability of the instance in different grids being the target instance.

3.2.4 Refinement Module

Once the target probability map is derived, the segmentation mask that corresponds to the grid position of the maximum value is selected as the target mask. However, because the output spatial size of the Segmentation Branch is limited due to the limitation of computational resources, and the vision features used in our network are from higher-level layers of the vision backbone, the output directly taken from the Segmentation Branch is coarse. It is desired to introduce low-level and high-solution features to enhance the spatial details of predicted mask. Thus, we concatenate the original image and the resized predicted mask together, as the input to the refinement module. The refinement module consists of three 3×3 convolution layers with upsampling layers in between and outputs a 1-channel prediction map. This output map is then added with the upsampled coarse segmentation map from the Segmentation Branch to form the final prediction.

3.2.5 Loss Functions

In the proposed network, there are three loss terms to guide the network training: identifying loss l_{ide}, segmentation loss l_{seg}, and refinement loss l_{ref}. The places where these losses are applied are shown in Figure 3.2. The whole loss function is formed by the weighted sum of all loss terms:

$$l = w_{ide} l_{ide} + w_{seg} l_{seg} + w_{ref} l_{ref} \tag{3.4}$$

Identifying loss. The ground-truth of identifying loss is a $S \times S \times 1$ grid map, indicating the grid location of the target instance. We first calculate the gravity

center G of the target instance, then label the grid that G falls into and its 8 neighborhood grids by 1, called positive grids G_{pos}. All other grids are annotated as negative grids with 0. Compared with assigning all grids in the mask as positive, this setting gives all objects the same number of grids, which allows the network to equally treat every objects, despite of their sizes. The Binary Cross Entropy (BCE) is used to measure the difference between the predictions and ground-truths. We use BCE in all three loss terms.

Segmentation Loss. The Segmentation Branch outputs S^2 masks, each of which corresponds to one grid of the Identification Branch. The ground-truth corresponding to the positive grid is the ground-truth instance mask. Notably, in referring segmentation, we only have one mask for each training sample: the mask of the instance that is referred to by the referring expression. Thus, we set the loss for negative grids to be zero and do not use them in training.

Refinement Loss. The refinement loss is the BCE of the ground-truth mask and the predicted mask. In the early stage of training, the output of the Identification Branch will be very inaccurate. Since the gradient of the refinement branch can be propagated back, selecting the wrong channel in the Segmentation Branch will be harmful to the whole network. Thus, we use an adaptive training strategy in this module. During training, after the coarse segmentation map is selected from the Segmentation Branch output, we evaluate its IoU with the ground-truth. If the coarse IoU is greater than a threshold θ, it is considered to be a successful identification. Otherwise, we consider the channel selection is wrong and set the refinement loss as 0.

TABLE 3.1: Ablation study results on the validation set of the RefCOCO. (IEM: Instance Extraction Module, RAM: Relationship Analyze Module, RM: Refinement Module)

Model	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9	IoU
Baseline	59.88	50.64	38.70	20.68	3.36	52.40
Baseline+IEM	74.53	67.80	58.09	35.58	5.73	61.47
Baseline+IEM+RAM	76.23	70.87	61.03	36.99	6.33	62.48
Baseline+IEM+RAM+RM	**77.18**	**73.43**	**66.59**	**53.01**	**18.85**	**66.19**

TABLE 3.2: Performance gap of replacing the output of each branch with the Ground-Truth. The experiments reveal the great potential of our new framework.

(A) Replacing the output of Identification Branch with GT

	GT Location	Predicted Location	Gap
val	76.32	66.19	−10.13
testA	78.23	68.45	−9.78
testB	73.35	62.73	−10.62

(B) Replacing the output of Segmentation Branch with GT

	GT Mask	Predicted Mask	Gap
val	80.41	66.19	−14.22
testA	82.48	68.45	−14.03
testB	77.22	62.73	−14.49

3.3 Experiments

3.3.1 Implementation Details

We train and test the proposed approach on three widely-used referring segmentation datasets: RefCOCO [3], RefCOCO+ [3] and RefCOCOg [4, 79]. We strictly follow the experiment settings in the pervious works [17, 76], *e.g.* all test/val images in three RefCOCO datasets are excluded when training the Darknet backbone. We also follow prior work [17, 98] and adopt the GloVe word embeddings [97] on Common Crawl 840B tokens. We use Adam with the base learning rate of 0.001 for optimization. Images are resized to 416×416 before sending to the network, and the network outputs a 8x downsampled mask to save memory usage in our implementation. All convolutional layers, except for the output layer of each module, are followed with a Leaky ReLU activation function and batch normalization. We set $w_{ide} = 10.0, w_{seg} = 0.03$, $w_{ref} = 0.5$ and $\theta = 0.3$, and grid number $S \times S = 13 \times 13 = 169$ in our settings. The network is trained for 55 epochs with a batch size of 12.

3.3.2 Ablation Study

In this section, we report the ablation study results of our framework. Besides the regular IoU, we also use another widely-used metric: Precision@X, to better demonstrate the function of each module in our framework. The Precision@X represents the percentage of images whose prediction IoU is above threshold X. The

TABLE 3.3: Experimental results of the IoU metric, and comparison of other methods with ours. *: Google split.

Model	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val	test	val*
DMN [60]	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [59]	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [76]	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [18]	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
BRINet [67]	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [98]	60.98	62.99	59.21	48.17	52.32	42.11	-	-	39.98
LSCM [64]	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
MCN [17]	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
CGAN [99]	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
Ours	**66.19**	**68.45**	**62.73**	**52.70**	**56.77**	**46.39**	**52.67**	**53.00**	**50.08**

value of low-threshold precisions reflects the percentage of successfully identified instances, which correspond to the identifying ability of the model. The value of high-threshold precisions reflects the quality of output masks, corresponding to the model's segmentation ability. Following [17, 67, 98], we compare the precision from threshold 0.5 to 0.9, on the validation set of the RefCOCO dataset. The results are shown in Table 3.1.

For our baseline model, we remove the Identification Branch, and change the number output channels of the Segmentation Branch to 1, *i.e.*, let the Segmentation Branch directly output the mask of the target instance, which is the same as one-stage methods. From Table 3.1, we can see that our baseline, with the Segmentation Branch only, works well as a one-stage method and achieves acceptable results. However, its precision at low-threshold is not high, which indicates its defects in identification.

For the "Baseline+IEM" model, we recover the Identification Branch and Segmentation Branch in the Instance Extraction Module (IEM), making network instance-aware, but replace the Relationship Analyze Module (RAM) with a convolutional layer. It can be seen that the performance is significantly increased, especially at lower thresholds, which shows that the IEM greatly enhances the identifying ability of the network.

Next, we test the proposed RAM. The proposed RAM further improves the IoU performance by enhancing the identification ability. The improvements brought by RAM shows that it can further enhance prediction by reasoning relationships

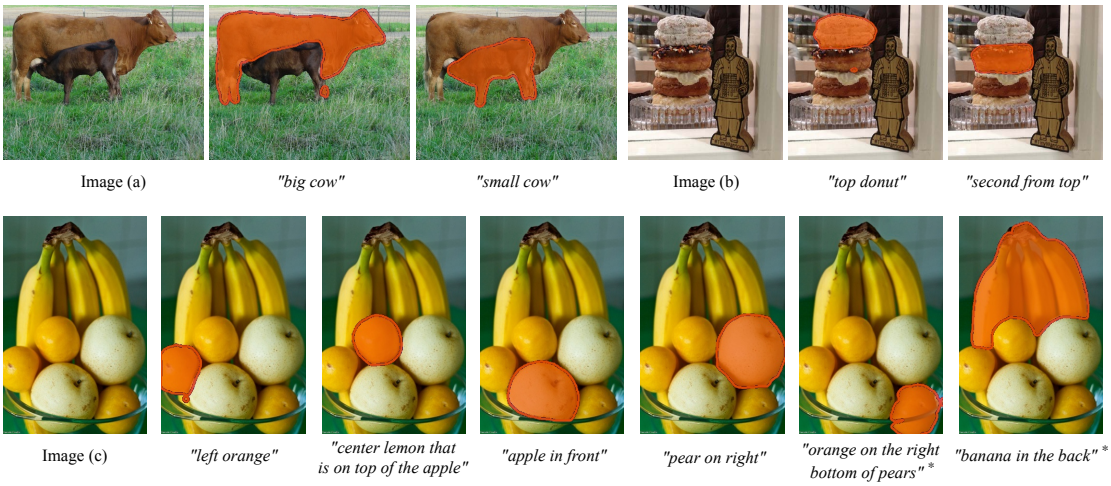


FIGURE 3.5: **Qualitative examples of the proposed approach.** The segmented part is highlighted in orange. (*: The datasets do not provide referring expressions for these two instances. We create phrases for them as user study.)

among instances in the input image. Also, from the last row of Table 3.1, we can see that the Refinement Module (RM) enhances the network's performance more significantly at higher thresholds, which demonstrates the effectiveness of the RM on segmentation quality.

Compared to the baseline, the overall performance of the proposed framework is 13.79% better in terms of IoU and 17.30% better in terms of Pr@0.5. This demonstrates the superiority of our method and shows that our method significantly enhances the referring segmentation from different aspects, *i.e.*, identification, and segmentation.

3.3.3 Branch Performance

In this section, we analyze the detailed performance of each branch in Table 3.2, by using the ground-truth to take over its output. The gap between the resulting performance and the original performance shows the areas for continued development of this branch. Notably, RAM is included in the Identification Branch in experiments in this section.

Identification Branch: Prediction *v.s.* GT. We disable the Identification Branch and use the ground-truth identify map to substitute for its output. This makes the model always find the right grid location of the target instance, which

represents the upper limit of the identifying ability of our model. Results are reported in Table 3.2a.

Segmentation Branch: Prediction *v.s.* GT. In the experiment, we replace all the output masks of Segmentation Branch with the corresponding ground-truth masks. In other words, the network always outputs the ground-truth mask of the object selected by the Segmentation Branch, whether this selection is correct or not. We get higher results with the ground-truth mask (see Table 3.2b), which demonstrate the effectiveness of the Identification Branch.

Comparing the two experiments, it is shown that replacing the outputs of Segmentation Branch with the ground-truth gives more performance gain. This is because the scale of the Segmentation Branch in our approach is small and the architecture is straightforward, suggesting the potential of our network.

3.3.4 Results on Benchmarks

Here we compare the proposed approach with previous state-of-the-art methods on three public referring segmentation benchmarks, RefCOCO, RefCOCO+, and RefCOCOg. We report the IoU results of the proposed approach against other methods in Table 3.3. It can be seen that our approach outperforms previous state-of-the-art methods on three datasets by $\sim 1\%$ in term of IoU. As mentioned previously, the Output Stride (OS) of our network is set to 8 to save memory resources. The performance of our network can be further enhanced by reducing the OS.

3.3.5 Visualizations

Example Results of our method are shown in Figure 3.5. To demonstrate the ability of our method of analyzing the relationships among instances, we report the segmentation results of different referring expressions on each image. The first row in Figure 3.5 is a comparison between two objects. In image (a) our method is aware of the spatial sizes of instances and in image (b) we show more complicated cases where exist numbered relationships, *e.g.*, "second from top". From the figure, we can see that our method successfully targets the instance with an expression

FIGURE 3.6: **Visualization of the Identification Branch.** The "Identifying Map" shows the output of the FPM.

that describes only the relative locations without the help of other information such as attributes. This shows that our method has a strong ability on modeling the relative relationship among objects. The image (c) is the most difficult case where multiple instances of different semantic categories are tangled together in a complex layout. Notably, the dataset only provides referring expressions for at most four instances in the image, so we add two more expressions of the rest of the instances in the last of the figure for a complete demonstration. Our method makes accurate identification of all instances in the image. It can be seen that our method can handle the relationship among objects of different semantic categories, *e.g. center lemon that is on top of the apple*, and can identify objects of different sizes, *e.g.* the big banana in the back, as well as small oranges in the front.

Identification Branch. Figure 3.6 displays the Identifying Map, *i.e.*, the output of the RAM module. It is shown that the RAM does not output segmentation

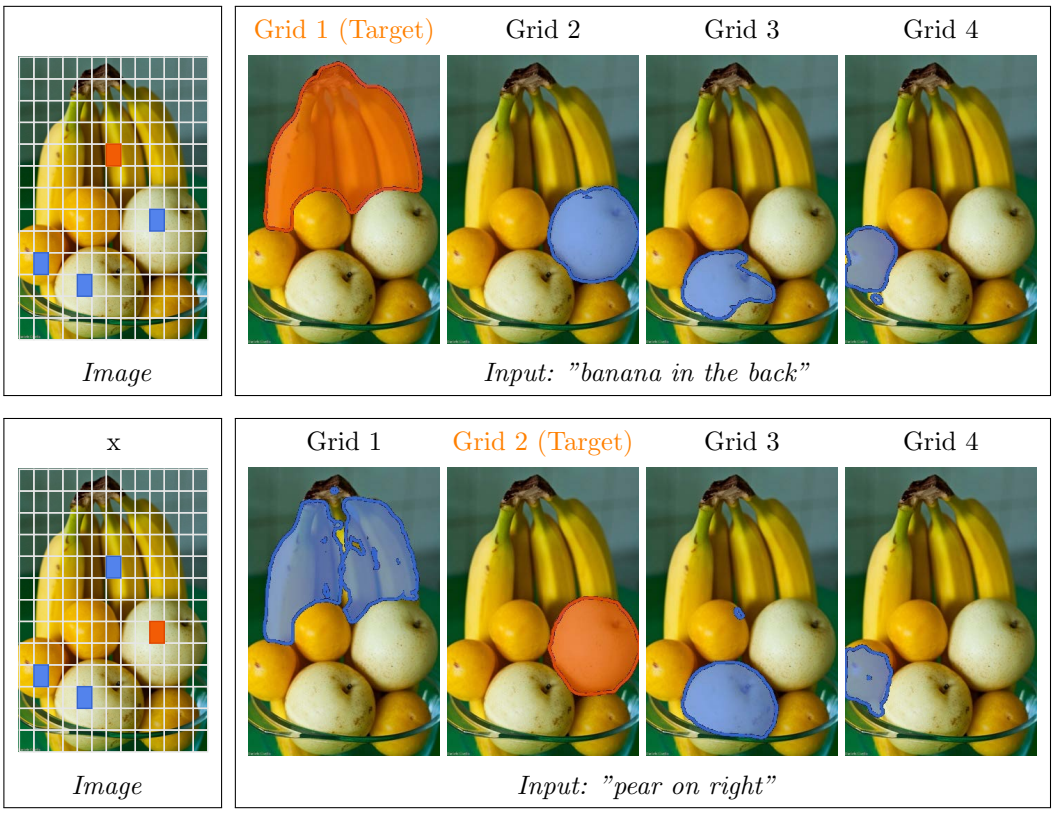


FIGURE 3.7: **Visualization of the output of the Segmentation Branch.** The linguistic feature helps the Segmentation Branch to focus on generating fine-grained segmentation masks for instances that are more possible to be the target, but still aware of other instances.

masks, but only highlights the grids that corresponding to the target instance.

Segmentation Branch. As we mentioned previously, we inject the language features to the Segmentation Branch to enhance the features of the target object. Thus, to illustrate the effectiveness of the Segmentation Branch and demonstrate the influence of language features on it, we visualize the output of the Segmentation Branch in Figure 3.7. From the figure, it can be seen that the output of the Segmentation Branch is influenced by language features. However, the model is still aware of other non-target objects, only the segmentation quality is influenced in different input expressions. Since the Segmentation Branch has a small network scale that only consists of a few convolutional layers but has a hard task to generate S^2 masks, this setting can help the network focus on outputting fine-grained masks for instances that are more possible to be the target. This also shows that though we only use one instance mask for each sample in training, our network still aware of other non-target instances.

3.4 Chapter Summary

In this chapter, we address the challenging task of referring segmentation. To inherit the merits and overcome the limitations of the previous one-stage and two-stage methods, we propose a new framework that simultaneously and collaboratively segments instances in the image and builds the interactions among them for identification. We propose a feature propagation module to model the comparison relationships among all instances. In addition, we propose a refinement module that introduces low-level and high-resolution features to enhance the spatial details of the predicted segmentation mask. Experiments show that our approach enhances both the targeting performance and the mask quality. Without bells and whistles, we achieve new state-of-the-art results on three referring segmentation datasets, which demonstrates the effectiveness of our proposed approach.

Chapter 4

VLT: Vision-Language Transformer and Query Generation

4.1 Introduction

In referring segmentation, while the query expression implies the target object by describing its attributes and its relationships with other objects, objects in images relate to each other in a complex manner. Therefore, a holistic understanding of the image and language expression is desired. Another challenge is that the diverse objects/images and the unconstrained language expressions bring a high level of randomness, which requires the model high generalization ability in understanding different kinds of images and language expressions. In the previous section, our method proposes an identification branch, which is a downsampled feature with smaller spatial sizes. While this helps to address the holistic understanding issue, it still has many limitations due to the local nature of the CNN. Besides, the language understanding problem is not well-discussed in the last method. In this chapter, we will dive deeper into the two unsolved issues.

Firstly, to address the challenge of complicated correlations in the input image and query expression, we propose to enhance the holistic understanding of multi-modal information by designing a framework with global operations, in which direct interactions are built among all elements, *e.g.*, word-word, pixel-pixel, and pixel-word.

The Fully Convolutional Network (FCN)-like framework [13] is commonly used in existing referring segmentation methods [16, 60]. They usually perform convolution operations on the fused, *e.g.*, concatenated or multiplied, vision-language features to predict the segmentation mask for the target object. However, the long-range dependency modeling is intractable by regular convolution operation as its large receptive field is achieved by stacking many small-kernel convolutions. This oblique process makes the information interaction between long-distance pixels/words inefficient [66], thus is undesirable for the referring segmentation model to understand the global context expressed by the input image and language [18]. In recent years, attention mechanism has gained considerable popularity in the computer vision community thanks to its advantage in building direct interaction among all elements, which greatly helps the model in capturing global semantic information. There have been some previous referring segmentation works that use attention to alleviate the long-range dependency issues, *e.g.*, [18, 67, 100]. However, most of them rely on FCN-like pipelines and only use the attention mechanism as auxiliary modules, which limits their ability to model the global context. In this work, we reformulate the referring segmentation problem as a direct attention problem and re-construct the current FCN-like framework using Transformer [36]. We generate a set of query vectors from language features using vision-guided attention, and use these vectors to "query" the given image and predict the segmentation mask from the query responses, as shown in Figure 4.1. This attention-based framework enables us to implement global operation among multi-modal features in each computation stage and enhances the network's ability to capture the global context of both vision and language information.

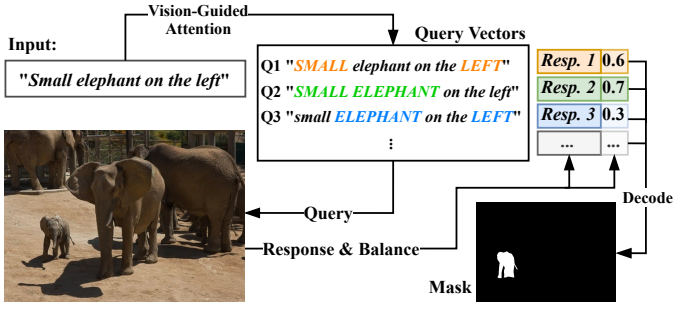


FIGURE 4.1: **The proposed method** dynamically produces multiple sets of input-specific query vectors to represent the diverse comprehensions of language expression. Queries are derived from language feautres, and different queries may emphasize different words by weighting, as indicated by the capitalized words in "Query Vectors".

Secondly, in order to handle the randomness caused by the various objects/images and the unrestricted language expressions, we propose to understand the input language expression in different ways incorporating vision features. In many existing referring segmentation methods, such as [17, 76], the language self-attention is often used to extract the most informative part and emphasized word(s) in the language expression. However, for these methods, their language understanding is derived solely from the language expression itself without interacting with the vision information. As a result, when the text prompt is complex and contains multiple aspects of description (*e.g.* multiple adjectives for one subject), they cannot distinguish which one will be more suitable and effective that can fit the image better. For example, solely given an expression *"The large balloon on the left"*, it is hard to determine whether the word *"large"* or the word *"right"* will be more helpful to find the target object. Hence, their detected emphases might be inaccurate or inefficient, as we will further discuss in Figure 4.5. On the other hand, in most existing vision-transformer works [1], the queries of the transformer decoder are a set of fixed and learned vectors, each of which predicts an object. However, experiments show that each query vector has its own operating modes, and is specifically targeted at certain kinds of objects [1], *e.g.*, specifically targeted at objects of a certain type or located in a certain area. The fixed queries in these works implicitly assume that the objects in the input image are distributed under some certain statistical rules, which does not consider well the randomness and huge diversity of the referring segmentation, especially the randomness brought by unconstrained language expressions. Also, the learnable queries are designed for detecting all the objects in the whole image instead of focusing on the target object indicated by the language expression, thus cannot efficiently extract informative representation that contains the clues to the target object.

To address these issues, we propose to generate input-specific queries that could focus on the clues related to the referred target object. We herein propose a Query Generation Module (QGM), which dynamically produces multiple query vectors based on the input language expression and the vision features. Each query vector represents a specific comprehension of the language expression and queries the vision features with different emphases. As shown in Figure 4.1, three queries focus on different information, respectively. These generated query vectors produce a set of corresponding masks in the transformer decoder though we only need one mask selected from them. Besides, we also hope to choose a more reasonable and

better comprehension way from these query vectors. Therefore, we further propose a Query Balance Module (QBM), which assigns each query vector a confidence measure to control its impact on mask decoding, and then adaptively selects the output features of these queries to better generate the final mask. The proposed QGM dynamically produces input-specific queries that focus on different informative clues related to the target object, while the proposed QBM selectively fuses the corresponding responses by these queries. These two modules work together to improve the different ways to understand the image and query language and enhance the network's robustness towards highly random inputs.

Thirdly, we introduce masked contrastive representation learning to further enhance the model's generalization ability and robustness to unconstrained language expressions. With the proposed Query Generation Module and Query Balance Module, we provide different understandings of a given expression, which can be viewed as a kind of intral-sample learning. Here we further consider inter-sample learning to explicitly endow the model with knowledge of different language expressions to one object. For the same target object, there are multiple ways to describe it. However, the final representations that predict the target mask should be the same. In other words, the output features of the Query Balance Module by different expressions for the same object should be the same. To this end, we utilize contrastive learning to narrow down the features of different expressions for a same target object, while distinguishing the features of different objects. What's more, we observe that the model tends to overly rely on specific words that provide the most discriminative clues or frequently occur in training samples, while ignoring other complementary information. The excessive reliance on specific words will damage the model's generalization ability, for instance, the model may not well understand testing expressions that do not contain common discriminative clues in the training samples. To address this issue, we introduce masked language expressions in contrastive representation learning, which randomly erases some specific words from the original language expression. The masked language expression and the original expression refer to the same target object, they are considered as a positive pair in the contrastive representation learning to be close to each other and reach the same representation. The masked contrastive representation learning significantly enhances the model's ability in dealing with diverse language expressions in the wild.

The proposed approach builds deep interactions between language and vision information at different levels, which greatly enhances the utilization and fusion of multi-modal features. In addition, the proposed network is lightweight and its parameter scale is roughly equivalent to just seven convolution layers. In summary, our main contributions are listed as follows:

- We design a Vision-Language Transformer (VLT) framework to facilitate deep interactions among multi-modal information and enhance the holistic understanding of vision-language features.
- We propose a Query Generation Module (QGM) that dynamically produces multiple input-specific queries representing different comprehensions of the language, and a Query Balance Module (QBM) to selectively fuse the corresponding responses by these queries.
- We introduce a masked contrastive representation learning to enhance the model's generalization ability and robustness to deal with the unconstrained language expressions by learning inter-sample relationships.
- The proposed approach is lightweight and achieves new state-of-the-art performance consistently on three referring image segmentation datasets, RefCOCO, RefCOCO+, G-Ref, and two referring video object segmentation datasets, YouTube-RVOS and Ref-DAVIS17.

4.2 Methodology

The overall architecture of the proposed Vision-Language Transformer (VLT) is shown in Figure 4.2. The network takes a language expression and an image as inputs. First, the input image and language expression are projected into the linguistic and visual feature spaces, respectively. Then, vision and language features are inputted to the proposed Query Generation Module (QGM) to generate a set of input-specific query vectors, which represent different aspects of the language expression under the guidance of visual clues. At the same time, vision and language features are fused to multi-modal feature by the proposed Spatial Dynamic Fusion (SDF), and the multi-modal feature is sent to the transformer encoder to produce a group of memory features. The query vectors in Q generated by our

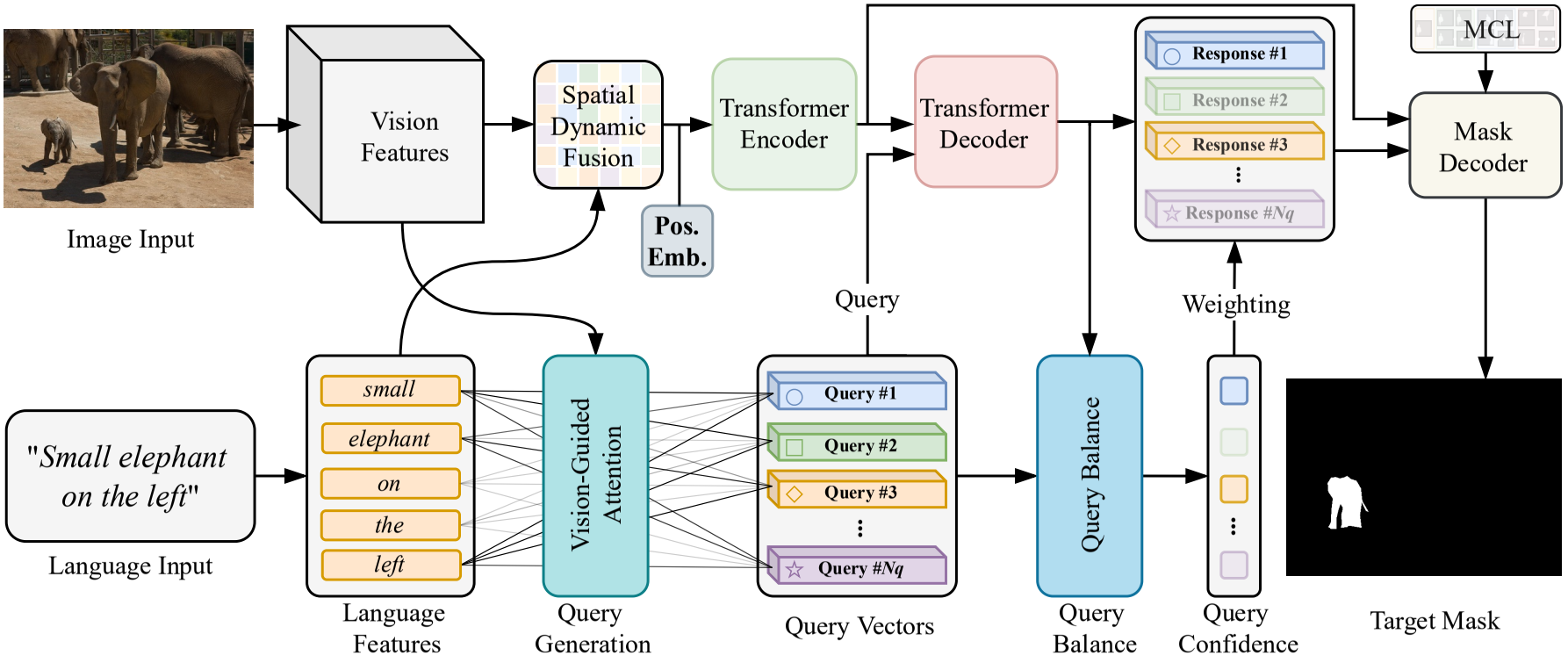


FIGURE 4.2: **The overview architecture of the proposed Vision-Language Transformer (VLT).** Firstly, the given image and language expression are projected into visual and linguistic feature spaces, respectively. A Spatial Dynamic Fusion module is then employed to fuse vision and language features, generating multi-modal feature inputted to the transformer encoder. The proposed Query Generation Module generates a set of input-specific queries according to the vision and language features. These input-specific queries are sent to the decoder, producing corresponding query responses. These resulting responses are selected by the Query Balance Module and then decoded to output the target mask by a Mask Decoder. "Pos. Emb.": Positional Embeddings. "MCL": Masked Contrastive Learning.

proposed QGM are employed to "query" K and V derived from memory features in transformer decoder, *i.e.*, Attention$(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$, where d_k is the dimensionality of K. The resulting responses from transformer decoder are then selected by a Query Balance Module (QBM) with different confidence weights. Finally, the mask decoder takes the weighted responses from QBM and the output feature from transformer encoder as inputs and outputs a mask for the target object. Masked Contrastive Learning (MCL) is used to supervise the features in Mask Decoder to narrow down the features of different expressions for the same target object while distinguishing the features of different objects. Positional embeddings are used to supplement the pixel position information in the permutation-invariant transformer architecture.

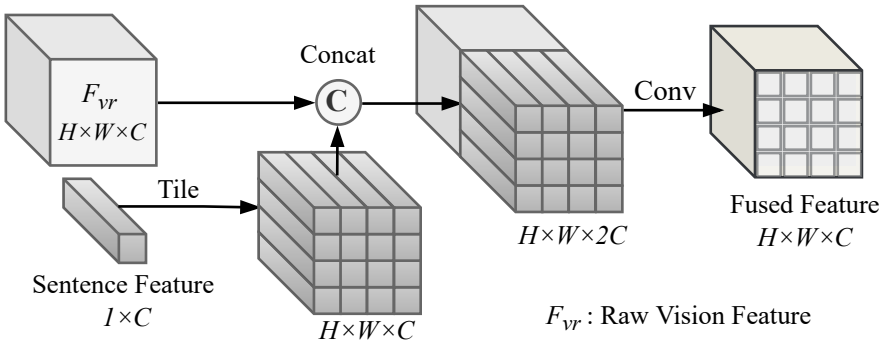


FIGURE 4.3: **Tile-and-concatenate fusion.** The language feature is identically copied to every position across the $H \times W$ map.

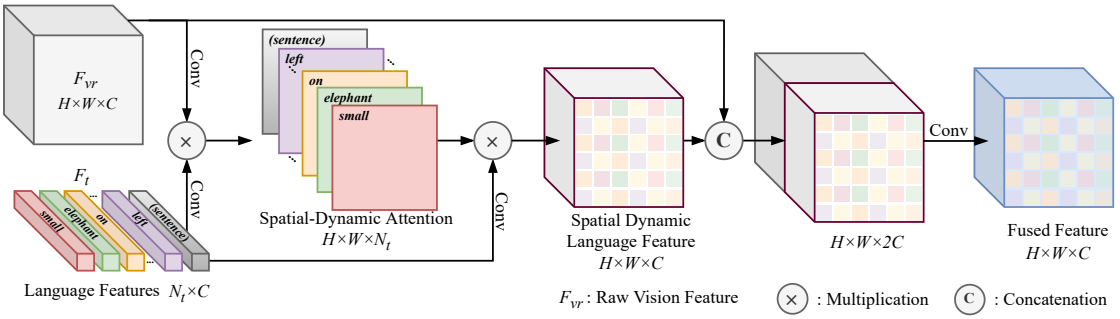


FIGURE 4.4: **An illustration of the proposed Spatial-Dynamic Fusion (SDF).** Different from the conventional "tile-and-concatenate" fusion, the proposed SDF finds a word attention set and derives a tailored language feature vector for each pixel in the image feature.

4.2.1 Spatial-Dynamic Multi-Modal Fusion

After backbone features for language and image are extracted, the first step is to preliminarily fuse them together and generate a multi-modal feature. For referring segmentation, effective multi-modal information fusion is critical and challenging because the unconstrained expression of natural language and the diversity of objects in scene images bring huge uncertainty to the understanding and fusion of multi-modal features. However, as far as our knowledge, most of the previous approaches conduct multi-modal feature fusion simply by either concatenation [16, 18, 58, 67] or point-wise multiplication [17, 62, 69] of vision feature and language feature. The language feature, which is a 1D vector, is usually tiled to every position of the vision feature [16, 18], as shown in Figure 4.3. Under the "tile-and-concatenate" operation, the language feature is identically copied to every position across the $H \times W$ map.

Although such kinds of fusion techniques are simple and have achieved reasonable performance, there are a few drawbacks. Firstly, the features of individual single words are not fully utilized in this step. Secondly, the tiled language feature will be identical for all pixels across the image feature, which weakens the location information carried by the correlation between the language information and the visual information. Due to the diversity of objects in the input image, an image usually contains diverse information that can be very complex, where different regions may contain different semantic information. Meanwhile, the language expression can be interpreted with different emphases from different perspectives. We here emphasize the differences among pixels/objects, *i.e.*, the vision information across the image varies from place to place. Therefore, the informative words in a given sentence are different from pixel to pixel. The way of tiling ignores such differences and simply assigns the same language feature vector to every pixel, resulting in some confusion. It is better to make tailored feature fusion specifically for each individual pixel. In this work, we propose a Spatial-Dynamic Fusion (SDF) module, which produces different language feature vectors for different positions of the image feature according to the interaction between language information and corresponding pixel information. Each position selects its interested words and pays more attention to these words during multi-modal fusion.

An illustration of the proposed Spatial-Dynamic Fusion (SDF) module is shown in Figure 4.4. The proposed SDF module takes language features F_t, including

features of each word and the whole sentence, and image features F_{vr} as inputs. We first use language features and vision features to generate the Spatial-Dynamic Attention matrix by:

$$A_{\text{sd}} = \text{softmax}(\frac{1}{\sqrt{C}}\text{Conv}(F_{vr})\text{Conv}(F_t)^T), \tag{4.1}$$

where C is feature channel number and $\frac{1}{\sqrt{C}}$ is the scaling factor. A_{sd} is with the shape of $H \times W \times N_t$, where H and W are height and width respectively, and N_t is the number of language feature vectors in F_t. In this section, unless specified otherwise, Conv denotes 3×3 convolution operation with ReLU activation and batch normalization. Softmax normalization is applied along the N_t axis of the attention matrix A_{sd}. Each position of the spatial-dynamic attention A_{sd} is a weighting vector that indicates different importance of the N_t language features at this position. Therefore, a spatial dynamic language feature F_{sdl} is generated by:

$$F_{\text{sdl}} = A_{\text{sd}}\text{Conv}(F_t), \tag{4.2}$$

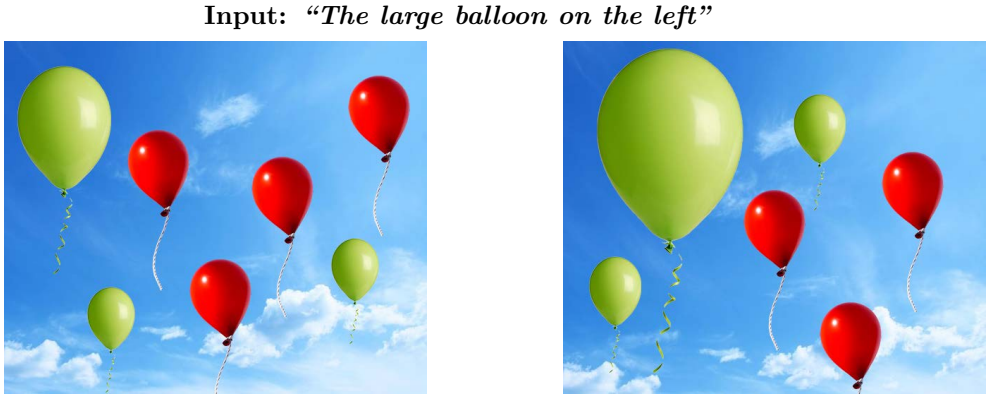
where F_{sdl} is in the shape of $H \times W \times C$, each vector of F_{sdl} across $H \times W$ is the language feature vector weighted by its correlation to the image context at a pixel position. The fused multi-modal feature F_{fused} is generated by:

$$F_{\text{fused}} = \text{Conv}(F_{\text{sdl}} \copyright F_{vr}), \tag{4.3}$$

where $\copyright$ denotes concatenation. Following previous transformer works [1, 101], we employ fixed sine spatial positional embeddings to supplement the pixel position information in the permutation-invariant transformer. The fused multi-modal feature and the positional embeddings are inputted to the transformer encoder (see Figure 4.2).

4.2.2 Query Generation Module

In most existing Vision Transformer works, *e.g.*, [1, 102–104], queries for the transformer decoder are usually a set of fixed learned vectors, each of which is used to predict one object and has its own operating mode, *e.g.*, specifying objects of a certain kind or located in a certain region. These works with fixed queries have an implicit assumption that objects in the input image are distributed under some



(A) (B)

FIGURE 4.5: **An example of one sentence with different emphasis.** For different images, the informative degree of words "large" and "left" are different.

statistic rules. However, such an assumption does not consider the huge diversity of the referring segmentation. The learnable queries are designed for detecting all objects in the whole image instead of focusing on the target object indicated by the language expression, thus cannot effectively extract informative representation that contains clues of the target object.

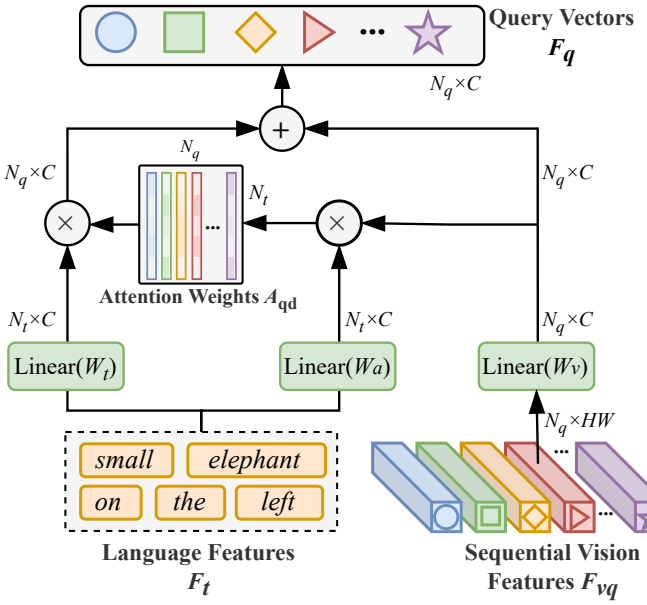


FIGURE 4.6: **Query Generation Module (QGM).** The QGM takes sequential vision feature F_{vq} and language features F_t as inputs and generates a group of input-specific query vectors F_q, which are then sent to the transformer decoder of our VLT.

For referring segmentation, the target object described by the given expression can be any part of the image. One feature of referring segmentation is that both the input image and language expression are unconstrained, and for different objects in different samples, users may give very different expressions that describes the object in different ways. For example, sometimes users may refer an object using its size and shape, and sometimes using its color and texture. In other words, the stochasticity of the target object's properties is significantly high. Therefore, fixed query vectors, like in most existing ViT works, cannot well represent the properties of the target object. Instead, the properties of the target object are hidden in the input language expression, *e.g.*, keywords like "blue/yellow", "small/large", "right/left", *etc.* To capture the informative clues and address the high stochasticity in referring segmentation, we propose a Query Generation Module (QGM) to adaptively generate the input-specified query vectors online according to the given image and language expression. Also, it is well known that for a language expression, the importance of different words is different. Some existing works address this issue by measuring the importance of each word. For example, [17] gives each word a weight and [76, 98] defines a set of groups, *e.g.*, location, attribute, entity, and finds the degree of each word belonging to different groups. Most works derive the weights by the language self-attention, which does not utilize the information in the image and only outputs one set of weights. But in practice, the same sentence may have different understanding perspectives and emphasis, and the most suitable and effective emphasis can only be known with the help of the image. We give an intuitive example in Figure 4.5. For the same input sentence *"The large balloon on the left"*, the word *"left"* is more informative for the first image while the word *"large"* is more useful for the second image. In this case, language self-attention cannot differentiate the importance between *"large"* and *"left"*, making the attention process less effective. In order to let the network learn different aspects of information and enhance the robustness of the queries, we generate multiple queries with the help of visual information, though there is only one target instance. Each query represents a specific comprehension of the given language expression with different emphasized word(s).

The architecture of the Query Generation Module is shown in Figure 4.6. It takes language feature $F_t \in \mathbb{R}^{N_t \times C}$ and raw vision feature $F_{vr} \in \mathbb{R}^{H \times W \times C}$ as inputs. In F_t, the i-th vector is the feature vector of the word w_i, which is the i-th word in the input language expression. N_t denotes the sentence length and is fixed

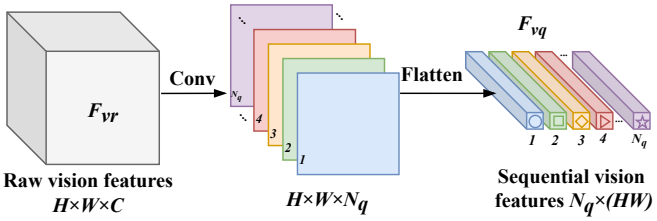


FIGURE 4.7: **The preparation process** of the sequential vision features for our Query Generation Module.

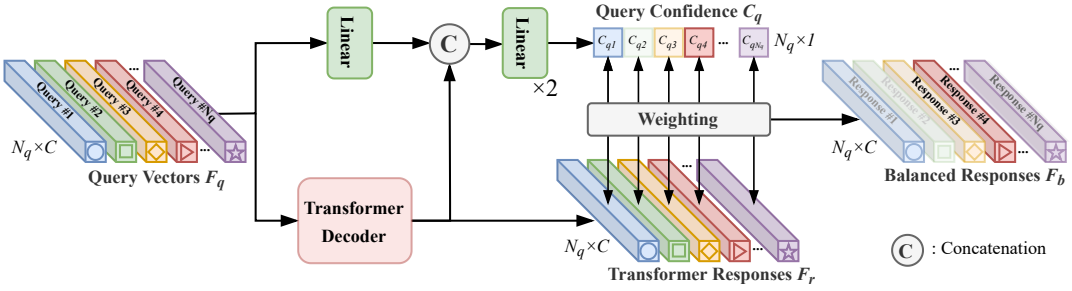


FIGURE 4.8: **Query Balance Module (QBM).** For each query vector, a confidence measure parameter C_q is computed to reflect how much it fits the prediction and the context of the image. The transformer responses F_r is weighted by the corresponding confidences C_q to control the influence of each query vector, generating balanced responses F_b.

over all inputs by zero-padding. This module aims to output N_q query vectors, each of which is a language feature with different attention weights guided by the vision information. Specifically, the vision features are firstly prepared as shown in Figure 4.7. We reduce the feature channel dimension size of the raw vision feature F_{vr} to query number N_q by three convolution layers, resulting in N_q feature maps. Each of them will participate in the generation of one query vector. The feature maps are then flattened in the spatial domain, forming a feature matrix F_{vq} of size $N_q \times (HW)$, *i.e.*,

$$F_{vq} = \text{Flatten}(\text{Conv}(F_{vr}))^T. \tag{4.4}$$

Next, we comprehend the language expression from multiple aspects incorporating the image, forming N_q queries from language. We derive the attention weights for language features F_t by incorporating the vision features F_{vq},

$$A_{\text{qd}} = \text{softmax}(\frac{1}{\sqrt{C}}\sigma(F_{vq}W_v)\sigma(F_tW_a)^T), \tag{4.5}$$

where $A_{\text{qd}} \in \mathbb{R}^{N_q \times N_t}$ is query-dynamic attention matrix, containing N_q different attention vectors to F_t. $W_v \in \mathbb{R}^{(HW)\times C}$ and $W_a \in \mathbb{R}^{C\times C}$ are learnable parameters, σ is activation function Rectified Linear Unit (ReLU). The softmax function is applied across all words for each query as normalization. Like the Transformer architecture, the attention matrix A_{qd} is token-wise, each of the N_q vectors consists of a set of attention weights for different words. Different queries attend to different parts of the language expression. Thus, N_q query vectors focus on different emphasis or different comprehension ways of the language expression. Notably, after this step, for longer sentences, we randomly mask one of the most important words to enhance the generalization ability of the network. The details are shown in Section 4.2.5.

Next, the derived attention weights are applied to the language features:

$$F_q = A_{\text{qd}}\sigma(F_t W_t) + \sigma(F_{vq} W_v), \tag{4.6}$$

where $W_t \in \mathbb{R}^{C\times C}$ and W_v are learnable parameters, $F_q \in \mathbb{R}^{N_q \times C}$ contains N_q query vectors $\{F_{q1}, ..., F_{qn}, ...F_{qN_q}\}$. We add a residual connection from vision feature F_{vq} to enrich the information in query vectors. Each F_{qn} is an attended language feature vector guided by vision information and serves as one query vector to the transformer decoder.

4.2.3 Query Balance Module

We get N_q different query vectors from the proposed Query Generation Module. Each query represents a specific comprehension of the input language expression under the interactive guidance of the input image information. As we discussed before, both the input image and language expression are of high arbitrariness. Thus, it is desired to adaptively select the better comprehension and let the network focus on the more reasonable and suitable comprehension. On the other hand, as the independence of each query vector is kept in the transformer decoder [1] but we only need one mask output, it is desired to balance the influence of different queries on the final output. Therefore, we propose a Query Balance Module (QBM) to dynamically assign each query vector a confidence measure that reflects how much it fits the prediction and the context of the image.

The architecture of the proposed QBM is shown in Figure 4.8. Specifically, the inputs of Query Balance Module are the query vectors F_q from the Query Generation Module and its corresponding responses from the transformer decoder, F_r, which is of the same size as F_q. In the Query Balance Module, the query vectors after going through a linear layer and their corresponding responses are first concatenated together. The linear layers are employed to derive confidence levels according to the query vectors F_q and their corresponding responses F_r. Then, a set of query confidence levels C_q, in the shape of $N_q \times 1$, are generated by two consecutive linear layers. Sigmoid, $S(x) = \frac{1}{1+e^{-x}}$, is employed after the the last linear layer as an activation function to control the output range. Let F_{rn} and C_{qn} denote the corresponding response and query confidence to the n-th query F_{qn}, respectively. Each scalar C_{qn} shows how much the query F_{qn} fits the context of its prediction, and controls the influence of its response F_{rn} to the mask decoding. Each response F_{rn} is multiplied with the corresponding query confidence C_{qn}, *i.e.*, $F_{bn} = F_{rn}C_{qn}$. The balanced responses $F_b = \{F_{b1}, ..., F_{bn}, ..., F_{bN_q}\}$ are sent for mask decoding. The proposed QGM dynamically produces input-specific queries that focus on different informative clues related to the target object, while the proposed QBM selectively fuses the corresponding responses to these queries. These two modules work together to prominently boost the diversity to understand the image and query language, and enhance the model's robustness towards highly stochastic inputs.

4.2.4 Mask Decoder

The output of the Query Balance Module F_b with the size of $N_q \times C$ is sent to the mask decoder, as shown in Figure 4.9. In the mask decoder module, F_b is utilized as a set of mask generation kernel to process the vision-dominated feature F_{ve} from the transformer encoder, to produce mask feature F_m, *i.e.*,

$$F_m = F_{ve}F_b^T, \tag{4.7}$$

where F_{ve} is with size of $HW \times C$ so that F_m has size of $HW \times N_q$. Then we reshape F_m to $H \times W \times N_q$ for the final mask generation. We use three stacked 3×3 convolution layers for decoding followed by one 1×1 convolution layer for outputting the final predicted segmentation mask. To control the output size and

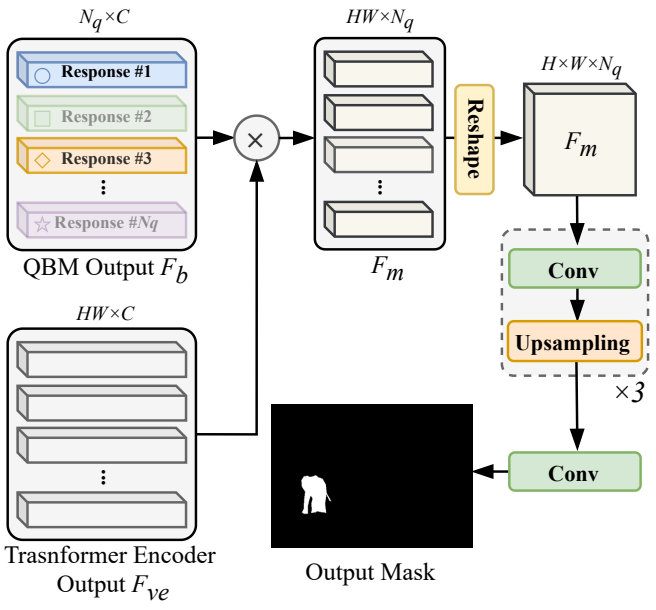


FIGURE 4.9: **The Mask Decoder** takes the outputs of Query Balance Module (QBM) F_b and Transformer Encoder F_{ve} to generate the output mask.

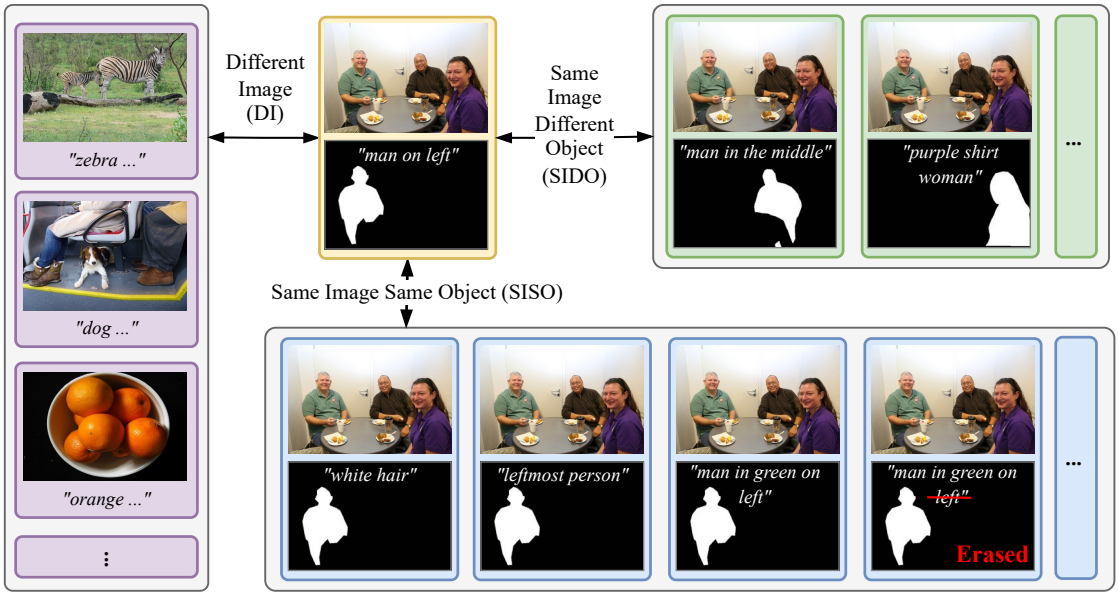


FIGURE 4.10: **Different kinds of inter-sample relationships.** SISO: Same Image, Same Object (but different expressions). SIDO: Same Image, Different Object. DI: Different Image. We erase some common word(s) in the long sentences and add such samples into SISO.

generate a higher-resolution mask, upsampling layers are placed after each of the three 3×3 convolution layers. To better demonstrate the effectiveness of the proposed transformer module, the Mask Decoding Module in our implementation does not utilize any backbone CNN features. We employ the Binary Cross-Entropy loss on the predicted masks to supervise the network training.

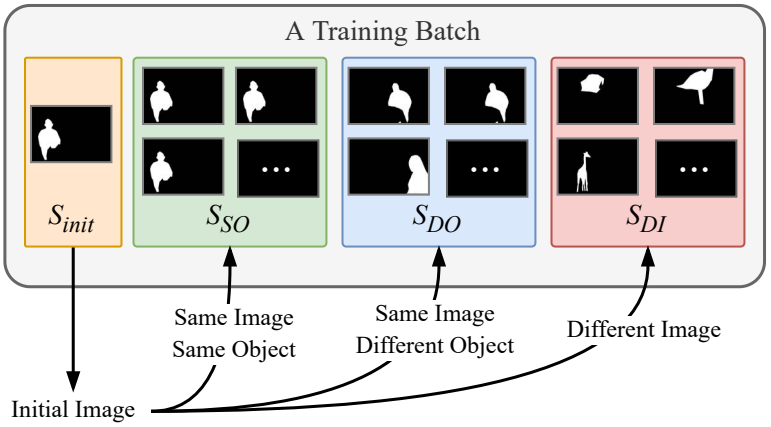


FIGURE 4.11: **One training batch** in inter-sample learning.

4.2.5 Masked Contrastive Learning

Here we further consider inter-sample learning to explicitly endow the model with the knowledge of different language expressions to one object. The given expression in natural language is unconstrained. There are multiple ways to describe the same target object, which brings challenges in understanding these expressions. Given an image I that contains N_O objects $\{O_1, ..., O_i, ..., O_{N_O}\}$, every object O_i in I can be referred by N_E different expressions $\{E_i^1, ..., E_i^j, ..., E_i^{N_E}\}$. A sample $S(I, O_i, E_i^j)$ of referring segmentation defines a mapping from an expression to the target object: $\{E_i^j \rightarrow O_i | I\}$. Based on the original definition of referring segmentation [16], one referring expression can only have one unambiguous target, so the mappings between E_i and O_i are in general many-to-one. An object in the image can be described by many different language expressions, but one language expression should unambiguously point to one and only one instance. Thus, no matter what kind of expressions are given for the target object, the final mask is the same, *i.e.*, the feature F_m in Eq. (4.7) for generating the final mask is the same. Motivated by this, here we introduce contrastive learning that forces the network to narrow the distance of features of different expressions for the same target object while

enlarging the distance of features for different objects. Furthermore, to provide more positive pairs and enhance the model's generalization ability to the input language, we randomly mask some specific words in the language expression and add these masked expressions to the positive samples of the original expression.

To sample the training pairs for contrastive learning, we summarize the inter-sample relationships into three categories: 1) Different Image (DI), 2) Same Image Different Object (SIDO), 3) Same Image Same Object (SISO), as shown in Figure 4.10. Unlike existing methods that construct the training batches in a fully random manner, we intend to let one batch have all kinds of inter-sample relationships. Firstly we randomly choose an initial sample S_{init}, as shown in Figure 4.11. We denote its SISO images as S_{SO}, whose image I and object O_i are the same as S_{init} but expressions E_i^j are different from S_{init}, and denote its SIDO images as S_{DO}, which has the same image I as the initial sample but different target object O_i. When constructing a mini-batch, we first put the initial sample into it. Next we intentionally put at most N_{SO} samples from S_{SO}, and at most N_{DO} samples from S_{DO}. The rest of the batch is filled with the randomly chosen DI samples. Under this mechanism, every training batch will contain all kinds of inter-sample relationships, as shown in Figure 4.11.

As we mentioned earlier, the features of SISO samples for generating the final mask should be the same. In contrast, for SIDO items, though they share the same input image so the output feature of the transformer may tend to be similar, the features for generating the mask prediction should be different because their target outputs are different. From this point, we introduce contrastive learning as feature-level supervision. In our approach, the Mask Decoder module plays the role in generating the output mask, hence we add the contrastive learning on the feature F_m, see Eq. (4.7), of the Mask Decoder module. Inspired by the InfoNCE loss [105], our loss is defined as follows:

$$\mathcal{L}_{CL} = -\frac{1}{N_{SO}} \sum_{S_+ \in S_{SO}} \log \frac{\exp\left(\frac{1}{\tau} \left\langle f_{S_+}, f_{S_{init}} \right\rangle\right)}{\sum\limits_{S \in S_{DO}, S_+} \exp(\frac{1}{\tau} \left\langle f_S, f_{S_{init}} \right\rangle)}, \tag{4.8}$$

where S_+ denotes a SISO sample of the initial sample, τ is a temperature constant, f_S is the feature F_m in the Mask Decoder module, and $\langle , \rangle$ denotes the cosine similarity function. This loss function forces that the mask feature of the initial

sample to be closer to its SISO samples that are supposed to have the identical output feature and mask, and force it to be away from its SIDO samples, which are supposed to have a non-overlap mask with it.

What's more, a sentence usually has more than one informative clue. However, the network tends to capture the most discriminative, or easiest clues to reach the training objective, resulting in underrating, even ignoring, other information. For example, given the image in Figure 4.10 and an expression *"man in green on left"*, we have experimentally observed that the network is over-influenced by the word *"left"* since it is more common in the dataset. We argue that overly relying on discriminative/common words harms the model's generalization ability. To address this issue, we propose to randomly mask some prominent words in the language expression and add these masked samples, as SISO samples, to our contrastive learning. Specifically, we measure the importance of each word when evaluating the language attention in the QGM, as mentioned in Eq. (4.5). For language expression that are longer than N_m words, we sum up the word attention vectors of all the N_q queries to generate a global attention weight for every word: $a_i = \sum_{n=1}^{N_q} a_{ni}$, where $a_{ni} \in A_{\text{qd}}$ represents the attention weight of the i-th word in attention vector for the n-th query, a_i denotes the global attention weight for i-th word. The global attention weights $\{a_1, a_2, ..., a_{N_t}\}$ reveal the importance of each word. To enhance the diversity of the training samples, words are chosen to be masked with a probability p_m, where the probability p_m is determined by the global attention weights by $p_{mi} = e^{a_i} / \sum_{j=1}^{N_t} e^{a_j}$. This probability-guided random setting lets more important words have higher masking chances while keeping the diversity of the training sentences, avoiding the network to be over-fitted by the masked samples. If a word is masked, its corresponding feature in F_t is changed. Thus, we apply a softmax function in Eq. (4.5) again to re-normalize it. As a consequence, a new set of query vectors and query responses are generated. The feature for Mask Decoder by this erased sentence is trained to be close to the original one by adding it as a positive sample S_+ in Eq. (4.8). In such a way, the network is encouraged to extract the information from words that are harder or not so discriminative rather than always relying on some high-frequency keywords, which could enhance the network's versatility in practical usage.

4.2.6 Network Architecture

Feature Extractor. We train two versions of VLT: one CNN backbone version and one Transformer backbone version. For the CNN version, since the transformer architecture only accepts sequential inputs, the original image, and language input must be transformed into feature space before sending to the transformer. For vision features, following [1], we use a CNN backbone for image encoding. We take the features of the last three layers in the backbone as the input for our encoder. By resizing the three sets of feature maps to the same size and summing them together, we get the raw vision feature $F_{vr} \in \mathbb{R}^{H\times W\times C}$, where H, W is the spatial size of features, and C is the feature channel number. For language features, we first use a lookup table to convert each word into word embeddings [106], and then utilize an RNN module to achieve contextual understanding of the input sentence and convert the word embedding to the same number of channels as the vision feature, resulting in a set of language features $F_t \in \mathbb{R}^{N_t\times C}$. F_{vr} and F_t are then sent to the Spatial Dynamic Fusion module and the Query Generation module as vision and language features. For the Transformer backbone version, we employ Swin-Transformer [38] and BERT [45] as vision and language backbone feature extractor.

TABLE 4.1: Comparison with Convolutional Networks, containing seven 3×3 Conv layers, in terms of parameter size and performance.

Type	#params	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
7 Conv Layers	$\sim$ 16.6M	60.42	66.44	61.86	53.22	44.72	17.27
Transformer	$\sim$ 17.5M	65.24	73.39	68.01	60.83	47.99	20.07

Transformer Module. We use a complete but shallow transformer to apply the attention operations on input features. The network has a transformer encoder and a decoder, each of which has two layers. We use the standard Transformer architecture as defined by Vaswani *et al.* [36], in which each encoder layer consists of one multi-head attention module and one feed-forward network (FFN), and each decoder layer consists of two multi-head attention modules and one FFN. We flatten the spatial domain of the fused multi-modal feature F_{fused} into a sequence, forming the multi-modal feature $F'_{fused} \in \mathbb{R}^{N_v\times C}, N_v = HW$. The transformer encoder takes F'_{fused} as input, deriving the memory features about vision information $F_{mem} \in \mathbb{R}^{N_v\times C}$. Before sending it to the encoder, we add a fixed sine spatial positional embedding on F'_{fused}. F_{mem} is then sent to the transformer decoder as

keys and values, together with N_q query vectors produced by the Query Generation Module. The decoder queries the vision memory feature with language query vectors and outputs N_q responses for mask decoding.

4.3 Experiments

We conducted extensive experiments to demonstrate the effectiveness of our proposed Vision-Language Transformer (VLT) for referring segmentation. In this section, we introduce implementation details of our approach, benchmarks we used in the experiments, and report both the quantitative and qualitative results of our proposed approach compared with other state-of-the-art methods.

4.3.1 Implementation Details

Experiment Settings. Following previous works [17, 76], we use the same experiment settings. Our framework utilizes Darknet-53 [42] pretrained on partial MSCOCO as the visual CNN backbone. Images form the validation and test set of the RefCOCO series are excluded in the pretraining. We use bi-GRU [14] as the RNN implementation and the Glove Common Crawl 840B [97] for word embedding. The training image size is set to 416×416 pixels. Each Transformer block has eight heads, and the hidden layer size in all heads is set to 256. For RefCOCO and RefCOCO+, we set the maximum word number to 15, and for G-Ref, we set it to 20 as there are more long sentences. The Adam optimizer is used to train the network for 50 epochs, and the learning rate is set to $\lambda = 0.001$. The batch size is 32 on one 32G V100 GPU.

Metrics. We use two metrics in our experiments: mask Intersection-over-Union (IoU) and Precision with thresholds (Pr@X). The mask IoU demonstrates the mask quality, which emphasizes the model's overall performance and reveals both targeting and segmenting abilities. The Pr@X metric computes the ratio of successfully predicted samples using different IoU thresholds. Low threshold precision like Pr@0.5 reflects the identification performance of the method, and high threshold precision like Pr@0.9 reveals the ability of generating high-quality masks.

TABLE 4.2: Comparison of the proposed Query Generation Module (QGM) with other kinds of query generation ways. "F_t": directly use the language features F_t as query vectors. "Learnt": learnable parameter-queries that are fixed in testing, similar with [1].

No.	Query Type	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
1	F_t	60.26	69.88	64.61	56.70	43.62	18.06
2	Learnt	58.60	67.84	59.98	53.23	44.60	16.33
3	QGM (ours)	65.24	73.39	68.01	60.83	47.99	20.07

TABLE 4.3: Ablation study of Query Numbers N_q. ‡: without Query Balance Module (QBM).

N_q	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
1	57.34	67.04	60.11	52.03	40.82	10.28
2	60.78	70.18	63.50	55.41	44.20	16.03
4	61.58	70.92	64.33	56.02	44.23	15.22
8	64.35	72.61	66.98	58.83	46.98	19.63
16	**65.24**	**73.39**	**68.01**	**60.83**	47.99	**20.07**
32	65.12	73.21	67.59	60.13	**48.03**	18.64
16‡	63.80	71.96	67.46	59.73	47.22	19.71

4.3.2 Ablation Study

In this section, we conduct ablation studies on the test B of RefCOCO to demonstrate the effectiveness of the proposed modules in our Vision-Language Transformer framework.

Transformer *v.s.* ConvNet. To demonstrate the scale of our proposed network and verify the effectiveness of the transformer module, we compare our method with a regular ConvNet in terms of the performance and parameter size in Table 4.1. In the experiment, we replace the whole transformer-based modules, including the transformer encoder-decoder, the Query Generation Module, and the Query Balance Module with seven stacked 3×3 Conv layers that have similar parameters size to our transformer-based modules. It shows that the parameter size of our transformer-based module achieves a much superior performance while is only nearly equal to 7 convolutional layers. The transformer module outperforms the 7 Conv module with ∼5% margin in terms of IoU, and ∼7% margin in terms of Precision@0.5. This proves the effectiveness of the proposed transformer module.

Query Generation. In Table 4.2, we compare different kinds of query generation methods, including our proposed Query Generation Module (QGM), language

features F_t as queries, and learned parameters as queries. The Query Generation Module outperforms the other two methods with a large margin at about 5% - 7% in terms of IoU and 4% - 6% in terms of Pr@0.5. Firstly, we directly utilize the language features F_t as query vectors and send them into the transformer decoder. In detail, the given language expression is processed by an RNN network, then the output for every word, and the output for the whole sentence, are used as query vectors. It can be seen in Table 4.2 that the performance of F_t as queries is ∼5% worse than QGM, which is because the information between words is not sufficiently exchanged and the understanding of language is derived from language itself, as we discussed in Section 4.2.2. The proposed Query Generation Module has a much superior performance to the "F_t" as queries. This demonstrates that the proposed QGM effectively understands the language expressions and produces valid attended language features under the guidance of visual information.

We set 16 query vectors that are initialized with uniform distribution at the beginning of the training in our experiment, and train these query-parameters together with the network. As the "learnt" in Table 4.2, the performance of these learned fixed query vectors is not satisfying, only 58.50%, which shows that such learned query-parameters cannot represent the target object as effectively as online generated input-specific queries by the proposed QGM.

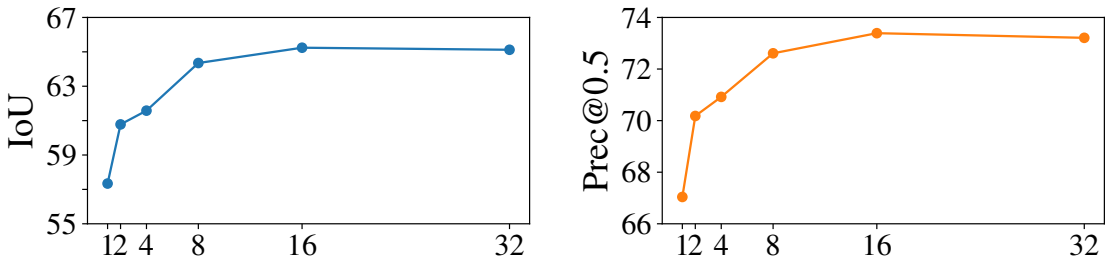


FIGURE 4.12: Performance gain by increasing the query number N_q.

Query Number N_q. To demonstrate the influence of the query number N_q on the results, we evaluate the network's results with different numbers of query vectors. As we can see in Table 4.3 and Figure 4.12, though only one segmentation mask is required in the final prediction, multiple queries are desired for providing diverse clues and can achieve better results than a single query. As shown in Table 4.3 and Figure 4.12, by increasing the query number N_q, the performance gradually gets higher, and a significant performance gain of about 8% is achieved from 1 query

TABLE 4.4: Ablation study of Multi-Modal Fusion.

Type	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
Tile	64.40	72.16	66.82	58.33	47.20	20.03
Tile + Conv×4	64.45	72.19	66.63	58.23	47.40	20.01
SDF	65.24	73.39	68.01	60.83	47.99	20.07

to 16 queries. The performance gain slows down after the query number N_q is larger than 8, therefore we select $N_q = 16$ as the default setting. The performance gain achieved by larger N_q verifies that multiple input-specific queries produced by the proposed QGM dynamically represent the diverse comprehensions of language expression. When the Query Balance Module (QBM) is discarded, marked with ‡ in Table 4.3, a performance drop of 1.44% IoU is observed, which proves the advantage of the proposed QBM.

Tile *v.s.* Spatial-Dynamic Fusion. In Table 4.4, we compare the "tile-and-concatenate" fusion and our proposed Spatial-Dynamic Fusion (SDF). As we discussed in Section 4.2.1, the "tile" operation does not consider the difference of each pixel but uses an identical sentence feature for all pixels across the image. In contrast, the proposed spatial-dynamic fusion customizes a unique language feature for every pixel according to the interaction between language information and corresponding pixel information. As shown in Table 4.4, compared with "Tile", the SDF module brings a performance gain of 0.84% IoU and 1.23% Pr@0.5. The proposed SDF emphasizes the differences among pixels/objects and allows each position to select the more informative words, enhancing the multi-modal fusion and producing better multi-modal features. "Tile + Conv×4" in Table 4.4, which has the same number of parameters as the proposed SDF, does not bring better performance than "Tile" because our network already has a sequential convolution layers after the feature fusion.

Inter-Sample Learning. Here we demonstrate the effectiveness of our proposed inter-sample learning approach, Masked Contrastive Learning (MCL). The results are shown in Table 4.5. Firstly, we add Contrastive Learning (CL) in the training of our network. The CL does not contain masked sentences as SISO samples. From Table 4.5a, on the original testing set, the CL brings a performance gain of 1.27% in terms of IoU and 1.04% in terms of Pr@0.5, which demonstrates that the inter-sample learning does enhance the model's performance. Further, we introduce the samples with masked sentences as SISO samples, *i.e.*, positive pairs in contrastive

TABLE 4.5: Ablation study of Inter-Sample Learning.

(A) Experiments on original dataset and masked dataset

Type	IoU			Pr@0.5		
	Original	Masked	Gap	Original	Masked	Gap
w/o CL	63.43	59.53	-3.90	71.84	67.02	-4.82
w/ CL	64.70	61.02	-3.68	72.88	68.45	-4.43
w/ MCL	65.24	64.20	-1.04	73.39	72.19	-1.20

(B) Cross dataset validation

Type	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
w/o CL†	49.16	56.06	50.13	41.87	34.11	12.26
w/ CL†	49.92	57.01	51.25	42.13	35.54	12.81
w/ MCL†	52.35	60.41	55.12	49.80	39.35	14.76
Native	56.30	66.03	61.53	56.20	41.22	13.09

TABLE 4.6: Ablation study of word selection mechanism in MCL.

Select. Type	IoU	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
None	49.92	57.01	51.25	42.13	35.54	12.81
Random	50.08	57.33	51.30	43.02	35.72	12.60
Threshold θ	51.57	59.08	52.04	46.13	37.57	13.99
p_m	**52.35**	**60.41**	**55.12**	**49.80**	**39.35**	**14.76**

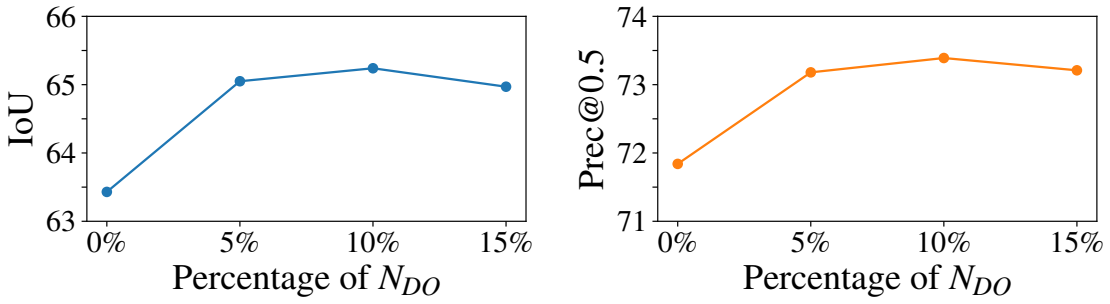


FIGURE 4.13: Ablation study of the percentage of N_{DO} in MCL.

learning. Compared with w/o CL, MCL brings a large performance gain of 1.81% IoU on the original dataset. Compared with CL, MCL further brings a performance gain of 0.54% in terms of IoU and 0.51% in terms of Pr@0.5, which shows the benefits brought by introducing samples with masked expressions in training. To better demonstrate the model's ability in dealing with unconstrained and diverse language expressions in the wild, we do another two testings: 1) erase some informative words of the given language expressions in these testing samples, see "Masked" in Table 4.5a; 2) cross datasets validation between two datasets that have

TABLE 4.7: Results on Referring Image Segmentation in terms of IoU and Prec@0.5. U: UMD split. G: Google split. Methods pretrained on large-scale vision-language training datasets are marked with †.

Methods	Visual Backbone	Textual Encoder	RefCOCO			RefCOCO+			G-Ref		
			val	test A	test B	val	test A	test B	$\text{val}_{(U)}$	$\text{test}_{(U)}$	$\text{val}_{(G)}$
DMN [60]	DPN92	SRU	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [59]	DL101	LSTM	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [76]	mrcn	LSTM	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [18]	DL101	None	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
CAC [61]	R101	LSTM	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
STEP [107]	DL101	LSTM	60.04	63.46	57.97	48.19	52.33	40.41	-	-	46.40
BRINet [67]	DL101	LSTM	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [98]	DL101	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-	39.98
LSCM [64]	DL101	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
CMPC+ [108]	DL101	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
MCN [17]	D53	bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
EFN [63]	R101	bi-GRU	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [65]	DL101	Self-Att	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [99]	DL101	bi-GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [62]	D53	bi-GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT (ours)	D53	bi-GRU	**67.52**	**70.47**	**65.24**	**56.30**	**60.98**	**50.08**	**54.96**	**57.73**	**52.02**
$\text{VLT}_{\text{Darknet53}}$ (ours) Prec@0.5			77.03	81.01	73.39	66.03	71.87	56.91	62.05	60.96	57.88
ReSTR [74]	ViT-B	T.f	67.22	69.30	64.45	55.78	60.44	48.27	-	-	54.48
MaIL [70]†	ViLT	BERT	70.13	71.71	66.92	62.23	65.92	56.06	62.45	62.87	61.81
CRIS [72]†	CLIP	CLIP	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
LAVT [71]	Swin-B	BERT	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
VLT (ours)	Swin-B	BERT	**72.96**	**75.96**	**69.60**	**63.53**	**68.43**	**56.92**	**63.49**	**66.22**	**62.80**
$\text{VLT}_{\text{Swin-B}}$ (ours) Prec@0.5			85.35	87.76	80.48	74.95	80.98	67.44	77.23	78.03	73.84

different common clues, *i.e.*, training on RefCOCO while testing on the validation set of RefCOCO+, marked with w/o CL†, *etc.* in Table 4.5b. Firstly, as shown in Table 4.5a, compared with the original dataset, w/o CL drops 3.9% in terms of IoU and 4.82% in terms of Pr@0.5 on the Masked testing samples. The result shows that the w/o CL model overly relies on common keywords and is heavily affected by the missing of these common clues. While for w/ MCL, the performance drop on the Masked validation is 1.04% and is much less than the performance drop of w/o CL, which verifies the model's robustness and generalization ability brought by introducing masked contrastive learning. Next, we do the cross-dataset validation on RefCOCO and RefCOCO+ in Table 4.5b. In RefCOCO, a large number of samples use absolute location (*e.g.*, *"the left"*, *"on the right"*, *etc.*) for describing the target object, but such kinds of expressions are not allowed in the RefCOCO+. Therefore, the cross datasets validation provides a good simulation of a practical scenario, in which the training information and testing are inconsistent, and only partial clues are available for testing. As shown in Table 4.5b, w/ MCL† outperforms w/o CL† 3.19% in terms of IoU and 4.35% in terms of Pr@0.5, which verifies the model's robustness and generalization ability in dealing with diverse language expressions that are different from training samples. "Native" in Table 4.5b denotes training & testing on RefCOCO+. As w/ MCL† *v.s.* "Native", we can see that the model trained on RefCOCO with MCL achieves competitive results on the validation set of RefCOCO+ compared to the model trained on RefCOCO+, proving that the proposed masked contrastive learning enhances the model's generalizability under open-world practical scenarios.

Next we do an ablation study about the word selection mechanism in our masked constrastive learning. Apart from the baseline model that disables MCL, we test three mask-word selection methods: 1) randomly choose a word to mask, 2) randomly choose a word with the weight a_i greater than a threshold θ to mask, 3) the proposed method that words are masked based on the probability p_m. Table 4.6 shows that our method outperforms other mask word selection mechanisms.

For the setting of N_{DO} in MCL, the ablation study in Figure 4.13 shows that the performance of the network reaches the peak when N_{DO} is set to 10% of the batch size. For N_{SO}, as the average number of expressions for an object is around 3, we can include all available Same Object (SO) samples in most cases.

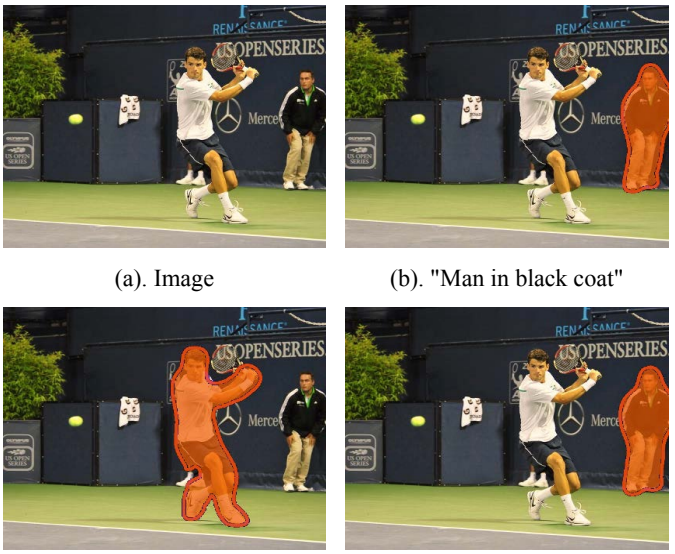


FIGURE 4.14: **Example results** of the Masked Contrastive Learning.

In Figure 4.14, we provide qualitative examples to show the effectiveness of the Masked Contrastive Learning. The original input language expression contains information in two aspects: color (*"black"*) and attribute (*"coat"*). The model without MCL overly relies on the more obvious color information (*"black"*), so it fails to predict when the word is erased. In contrast, the model with MCL successfully finds the target with partial information, showing that the MCL enhances the model's generalization ability to various language expressions.

We further test the training efficiency of our mask contrastive learning approach. We train the network with and without the MCL and report the GPU memory usage during training and the training speed of two runs with batch size set to 16. With MCL enabled, the GPU memory usage and average training speed are 18496MB and 0.479s/iter, respectively. Without MCL, they are 17842MB and 0.471s/iter, respectively. The increased training memory and time by MCL are less than 4% and 2%, respectively.

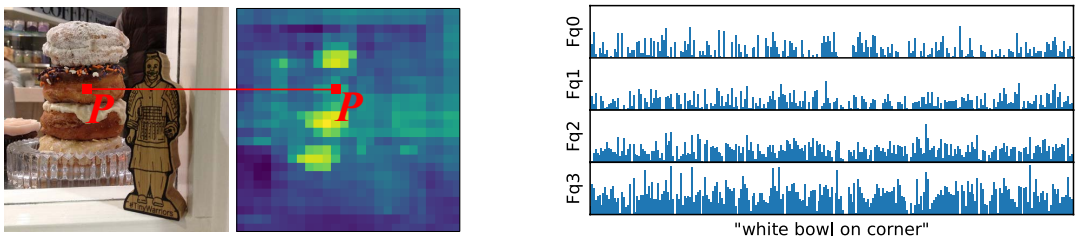
4.3.3 Comparison with State-of-the-art Methods

Here we compare the proposed Vision-Language Transformer (VLT) framework with previous state-of-the-art methods on three commonly-used benchmarks, RefCOCO, RefCOCO+, and G-Ref. The results are reported in Table 4.7. It can be seen that the proposed VLT outperforms previous state-of-the-art methods on all three benchmarks. On RefCOCO, the IoU performance of the proposed VLT is better than other methods, *e.g.*, LTS [62], with ∼2% gain on three different testing splits. Then on RefCOCO+, the proposed VLT achieves new state-of-the-art result and is around 2% better than previous state-of-the-art method. On the hard benchmark G-Ref that has longer language expressions, the proposed VLT consistently achieves new state-of-the-art referring segmentation performance with an IoU improvement of about 0.5%-3%, which demonstrates that the proposed VLT has good abilities to deal with hard cases and long expressions. We assume the reason is that, on the one hand, long and complex expressions usually contain more clues and more emphasis, and our proposed Query Generation Module and Query Balance Module can produce multiple comprehensions with different emphases and find the more suitable ones. On the other hand, harder cases also contain complex scenarios that need a holistic view and understanding of the given language expression and image, and the multi-head attention is more appropriate for such complex scenarios as a global operator. We also compared with other methods with stronger backbones, *e.g.*, DeepLab-R101 [12], MaskRCNN-R101 [24], ResNet101 [9], our backbone Darknet53 and our proposed modules are lightweight.

To compare with methods using stronger backbones, we further provide results with stronger visual and textual encoders in Table 4.7. We use the popular vision transformer backbone Swin-B [38] as visual encoder and BERT [45] as textual encoder to replace the Darknet53 [42] and bi-GRU [14], respectively. For this version, we do not specifically fine-tune Swin or BERT before training, but trained the whole network end-to-end. Methods pretrained on large-scale vision-language datasets are marked with †, *e.g.*, MaIL [70] adopts ViLT [75] pre-trained on four large-scale vision-language pretraining datasets and CRIS [72] employs CLIP [50] pretrained on 400M image-text pairs. As shown in Table 4.7, the proposed approach outperforms MaIL and CRIS by around 2%∼4% IoU without using large-scale vision-language datasets in pretraining, which demonstrates the effectiveness of our proposed modules with stronger visual and textual encoders. Especially,

the proposed approach VLT achieves higher performance gain on more difficult dataset G-Ref that has a longer average sentence length and more complex and diverse word usages, *e.g.*, VLT is ∼4% IoU better than MaIL [70] and LAVT [71] on $\text{test}_{(\text{U})}$ of G-Ref. It demonstrates the proposed model's good ability in dealing with long and complex expressions with large diversities, which is mainly attributed to input-conditional query generation and selection that well cope with the diverse words/expressions, and masked contrastive learning that enhances the model's generalization ability.

4.3.4 Qualitative Results and Visualization



(A) (B)

FIGURE 4.15: Visualizations of: (a) the attention map of point P in the transformer encoder; (b) different query vectors F_q.

In Figure 4.15a, we extract and visualize an attention map for a position "P" from the 2nd layer of our transformer encoder. It shows that in a single layer of the transformer, the attention of one output pixel globally extends to other input pixels far away. We also see that pixel on one instance attends to other instances, showing our network is able to capture long-range interactions between instances. In Figure 4.15b, we visualize four query vectors F_q (see Figure 4.6 and Eq. (4.6)). The four query vectors differ from each other and have different distributions of response peaks, which demonstrates the diversity of these input-specific query vectors.

Then, we visualize some qualitative examples of the proposed VLT in Figure 4.16. To demonstrate the identifying ability of our VLT, we show the mask predictions of two different input language expressions for every example. Image (a) and (c) are two typical examples that the language expression directly provides the location or color clues of the target object. In the second expression of Image (c), *"lighter color cat"*, it can be seen that the proposed VLT is able to handle the expressions

that indicate the target object by providing a comparison of it with other objects, *e.g.*, *"lighter"*. The examples of image (b) and (d) demonstrate the model's ability on understanding the attribute words, *e.g.*, *"stripes"*, and relatively rarer words, *e.g.*, *"floral"*. In the second expression of image (e), our VLT successfully identifies the target object referred by expression describing the relationships between objects, *i.e.*, "Elephant with rider". Image (f) contains a group of people, where all instances distribute densely in a complicated layout. The proposed method manages to identify the target instance with difficult language expressions that contain multiple aspects of clues, such as direction (*"9 o'clock"*), attributes (*"white coat"* & *"gray suit"*), and posture (*"kneeling"*).

FIGURE 4.16: **Qualitative examples of the proposed VLT.** For each example, the first image is the input image, and captions under the second and third images are the given language expressions.

4.3.5 Results on Referring Video Object Segmentation

Our proposed approach can also be applied to referring video object segmentation (RVOS) task with minor adaptation. We apply our model on each individual frame of the input video clip. We use the average vision features of all frames of a video clip as the vision features in the QGM (F_{vq} in Figure 4.6). This enables the query input to be kept identical for all frames in a video clip, achieving a temporal consistency across frames. When performing the contrastive learning on video data, we sample different objects S_{DO} in the ± 2 adjacent frames of the initial

object. As adjacent frames shares similar image structure with the original frame, we can enlarge the number of negative samples while keeping a similar behavior with our image model. According to the experiments, when only sampling S_{DO} in the same video frame, the $\mathcal{J}\&\mathcal{F}$ performance is 63.5 while it increases to 63.8 when adding the ± 2 adjacent frames.

TABLE 4.8: Results on Referring Video Object Segmentation.

Methods	Backbone	YouTube-RVOS			Ref-DAVIS17		
		$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
CMSA [18]	ResNet50	36.4	34.8	38.1	40.2	36.9	43.5
URVOS [85]	ResNet50	47.2	45.3	49.2	51.5	47.3	56.0
PMINet [109]	ResNeSt101	53.0	51.5	54.5	-	-	-
CITD [110]	Ensemble	61.4	60.0	62.7	-	-	-
ReferFormer [111]	V-Swin-B	62.9	61.3	64.6	61.1	58.1	64.1
VLT (ours)	V-Swin-B	**63.8**	**61.9**	**65.6**	**61.6**	**58.9**	**64.3**

In Table 4.8, we report the quantitative results of the proposed VLT on the validation set of the YouTube-RVOS [85] dataset and Ref-DAVIS17 [84] dataset. YouTube-RVOS is a large-scale referring video object segmentation benchmark, containing 3,978 video clips with around 15K language expressions. Ref-DAVIS17, building based on DAVIS17 [112], contains 90 video clips. The results are reported with three standard evaluation metrics: region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, as well as the mean value of the two metrics $\mathcal{J}\&\mathcal{F} = (\mathcal{J} + \mathcal{F})/2$.

To ensure a fair comparison, we use the Base model of Video Swin Transformer (V-Swin-B) [38, 113] as the backbone, the same as ReferFormer [111] (V-Swin-B version). "Ensemble" denotes visual encoder ensemble of three backbones, including ResNet101 [9], HRNet [114], and ResNeSt101 [115]. As shown in Table 4.8, although we do not design specific modules and training losses for RVOS like in ReferFormer [111], the proposed VLT achieves new state-of-the-art RVOS results consistently on both the YouTube-RVOS and Ref-DAVIS17, which demonstrate the effectiveness of the proposed VLT on referring video object segmentation.

4.4 Chapter Summary

In this chapter, we address the challenging multi-modal task of referring segmentation by introducing transformer to facilitate the long-range information exchange

that is difficult to achieve in conventional convolutional networks. We reformulate referring segmentation as a direct attention problem and propose a Vision-Language Transformer (VLT) framework that exploits the transformer to perform attention operations. To emphasize the differences among pixels/objects, we introduce a spatial-dynamic multi-modal fusion to produce a specific language feature vector for each position of the image feature according to the interaction between language information and corresponding pixel information. To solve the problem of ambiguous referring expressions because of the unknown emphasis, we propose a Query Generation Module and a Query Balance Module to comprehend the referring sentence better with the help of the referred image information. These two modules work together to prominently improve the diversity of ways to understand the image and query language. We further consider inter-sample learning to explicitly endow the model with knowledge of understanding different language expressions of one object. Masked contrastive representation learning is proposed to narrow down the features of different expressions for the same target object while distinguishing the features of different objects, which significantly enhances the model's ability in dealing with diverse language expressions in the wild. The proposed model is lightweight and achieves new state-of-the-art performance on three public referring image segmentation datasets and two referring video object segmentation datasets.

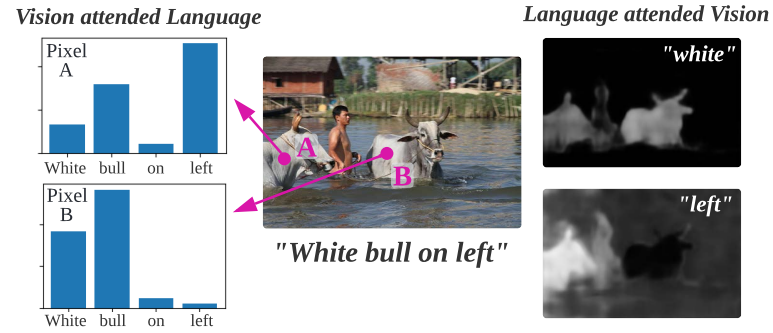
Chapter 5

Multi-Modal Mutual Attention and Iterative Interaction

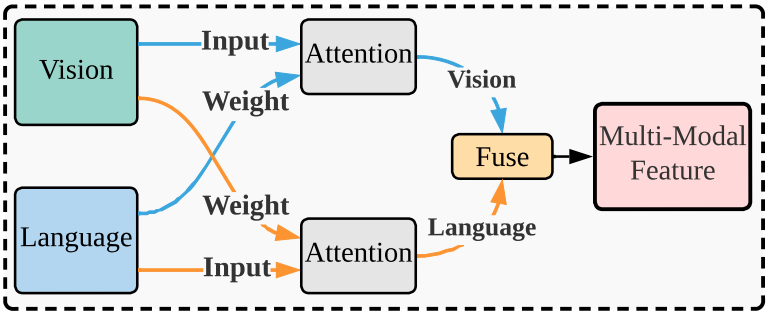
Another one of the biggest challenges in referring segmentation is that this task requires the reasoning of multiple types of information like vision and language, but the unconstrained expression of natural language and the diversity of objects in scene images bring huge uncertainty to the understanding and fusion of multi-modal features.

In the previous chapter, we propose VLT, which introduces the Transformer for referring segmentation, which has shown to be helpful to model the long-range dependencies in the image. However, the VLT, as well as most previous works [20, 62, 71] utilize the generic attention mechanism to model the relationship between language and vision information. The generic attention mechanism highlights the most relevant image region for each word in the language input, as shown in the right part of Figure 5.1 (a). By aggregating the input vision features according to the generated attention weights, as shown in the blue path in Figure 5.1 (b), the derived feature can describe each word using the combination of vision features. As the language feature is only used for calculating the attention weights, we call it language-attended *vision* feature (LAV).

The aforementioned attention mechanism is useful for processing vision information. However, since the referring segmentation is a multi-modal task, the language information is also essential. Thus, for processing the language information, a natural way is to introduce another type of attention that outputs language features.



(a). Two types of attention



(b). Multi-Modal Mutual Attention (M³Att)

FIGURE 5.1: An illustration of two attention types in referring segmentation and our proposed Multi-Modal Mutual Attention (M³Att).

For each pixel in the image, we can find the words that are most relevant to it, as shown in the left part of Figure 5.1 (a). By aggregating the features of these words together according to the attention weights, a set of vision-attended *language* features (VAL) for each image pixel can be derived. In contrast to LAV which is a set of vision features, VAL describes each pixel using language features. However, both VAL and LAV have limitations: they are both essentially single-modal features and only represent a part of the multi-modal information. For example, VAL is a set of language features for describing pixels, but the inherent vision feature of each pixel itself is not preserved. We argue that a holistic and better understanding of multi-modal information can be get by fusing features of two modalities together. However, this is not achievable in the generic single-modal attention mechanism.

Motivated by this, we empower the generic attention mechanism with feature fusing functionality, and design a Multi-Modal Mutual Attention (M³Att) mechanism. It integrates two types of attention into one module, as shown in Figure 5.1 (b). Our M³Att has two attention pathways. One pathway (orange path) processes and

outputs the vision-attended *language* feature, while the other one (blue path) processes and outputs language-attended *vision* feature. Two sets of features are then densely fused together, generating a real multi-modal feature with in-depth interaction of vision and language information. Using this M^{3}Att mechanism, we further design a Multi-Modal Mutual Decoder (M^{3}Dec) as an optimized feature fuser and extractor for multi-modal information, which greatly enhances the performance of the model for referring segmentation.

Next, we address the modal imbalance issue in the attention-based network. Due to the characteristic of the Transformer's decoder architecture, in M^{3}Dec as well as most attention-based works [20, 62], the language feature is only once inputted into the decoder at the first layer. In contrast, vision information is inputted to every decoder layer. This implies a modal imbalance issue: the network will tend to focus more on the vision information, and the language information may be faded away during the information propagation along the network. This issue will limit the strong feature fusing ability because of the lack of direct language input. From this point, we propose Iterative Multi-modal Interaction (IMI), which continuously transforms the language feature and enhances the significance of language information in the multi-modal feature at each layer of the M^{3}Dec, to fully leverage its fusing ability.

Furthermore, since the ground-truth segmentation mask is the only supervision, it cannot give direct and effective feedback to encourage the model to keep the language information from being lost. Also, as the IMI has a function of transforming the language feature, it is helpful to protect the integrity of the language information in the multi-modal information, and prevent them from being lost and distorted. We hence propose the Language Feature Reconstruction (LFR), which protects the validity of language information in the multi-modal features in M^{3}Dec. A language reconstruction loss is then introduced to supervise the multi-modal features directly.

Overall, the contributions of this work can be summarized as follows:

- We propose Multi-Modal Mutual Attention (M^{3}Att) and Multi-Modal Mutual Decoder (M^{3}Dec) for better processing and fusing multi-modal information, and build a referring segmentation framework based on it.

- We propose two modules: Iterative Multi-Modal Interaction (IMI) and Language Feature Reconstruction (LFR), to further promote an in-depth multi-modal interaction in M^{3}Dec.
- The proposed approach achieves new state-of-the-art referring image segmentation performance on RefCOCO series datasets consistently.

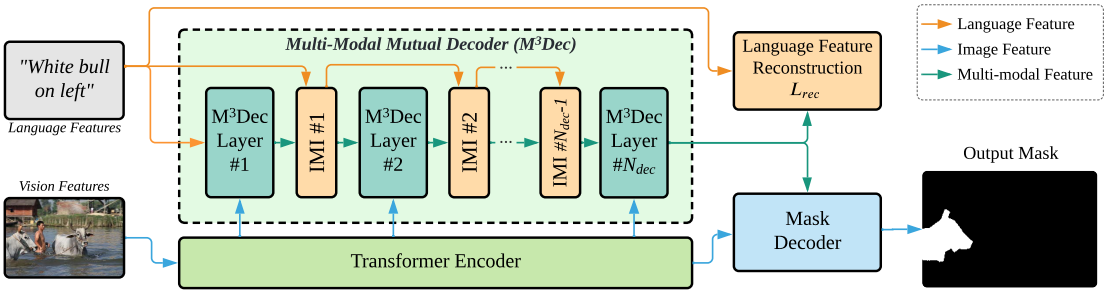


FIGURE 5.2: **The overall architecture of the proposed approach.** We propose Multi-Modal Mutual Decoder (M^{3}Dec) to fuse and process the multi-modal information from two inputs.

5.1 Methodology

The overview architecture of our proposed approach is shown in Figure 5.2. The network's inputs include an image I, and a language expression T containing N_t words. Following previous works [16, 17, 20], we first extract two sets of input backbone features: image feature F_{vis} from I using a CNN backbone, and language feature F_t and F_t' from T using a bi-directional LSTM. The image feature, F_{vis} has the shape of $H \times W \times C$, where H and W denote height and width respectively, C is the number of channels. For the language feature, the hidden states of the LSTM $F_t \in R^{N_t \times C}$ represent the feature for each word, while the final state output F_t' is used as the representation of the whole sentence. The channel number of language features is also C for the ease of fusion.

Then we send vision feature to a transformer encoder with N_{enc} layers to obtain deep vision information F_{enc}. Next, we input F_{enc} into our proposed Multi-Modal Mutual Decoder (M^{3}Dec) and Iterative Multi-modal Interaction (IMI), which give an in-depth interaction for the multi-modal information. Finally, the Mask Decoder takes the output from both transformer encoder and M^{3}Dec, and generates the

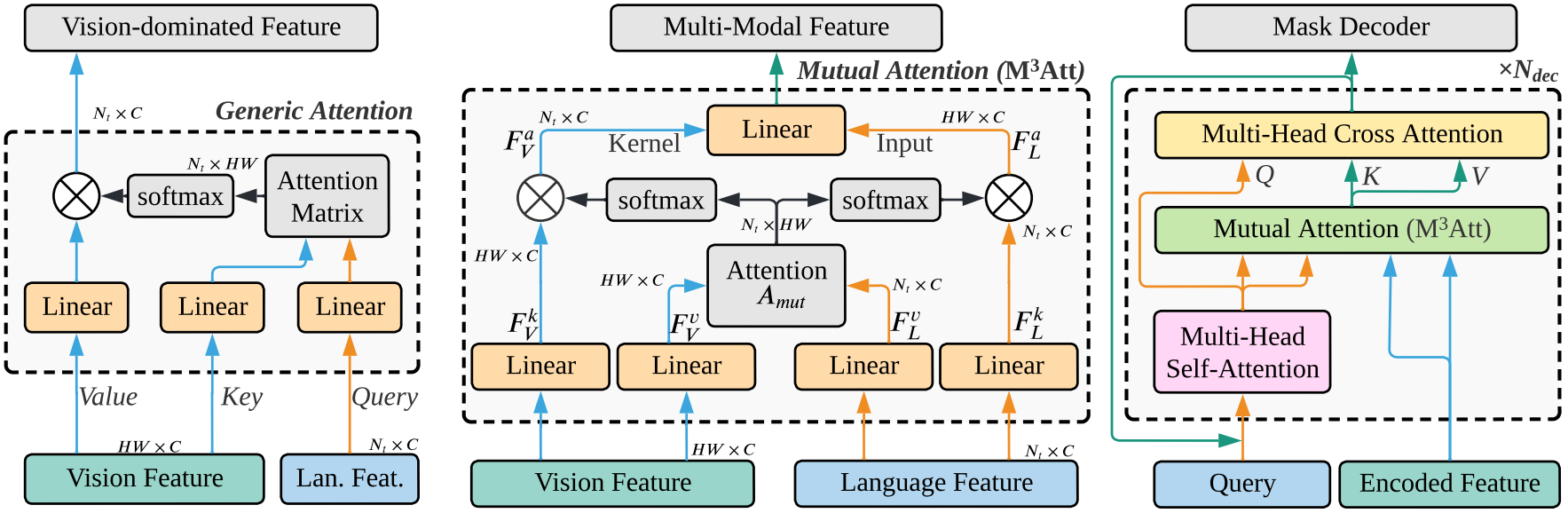
output mask. Moreover, we propose a Language Feature Reconstruction (LFR) module to encourage language usage in the M^{3}Dec, and prevent that the language information from being lost at the rear layers of the network. The details of each part will be introduced in the following sections.

5.1.1 Multi-Modal Mutual Attention

As mentioned above, most previous works use the generic attention mechanism for processing multi-modal information. Figure 5.3 (a) gives an example of such kind of mechanism, similar to [20]. Features from two modalities (query and key) are used to derive an attention matrix, that is then used to aggregate the vision feature for each word. In this process, the language feature is only used to generate the attention weights that indicate the significances of regions in the vision feature. Hence, language information is not directly involved in the output so that the output can be viewed as a reorganized single-modal vision feature. Even worse, this single-modal vision output is used alone as a query in the successive transformer decoder, dominating information in decoder. As a result, language information will be dramatically lost in the decoder. Thus we argue that the generic attention mechanism is good for *processing* features from the value input, but it lacks the ability of *fusing* features from two modalities. So, if it is used to process multi-modal information, the query input is not fully utilized, and features of two modalities are not densely fused and interacted.

To address this issue, we propose a Multi-Modal Mutual Attention (M^{3}Att), as shown in Figure 5.3 (b). M^{3}Att takes inputs from two modalities, and transforms each of them into two roles: key and value. This enables us to equally treat features from both modalities and fuse them together. Herein we use language feature $F_t \in R^{N_t \times C}$ and vision feature from the output of the transformer encoder $F_{enc} \in R^{HW \times C}$ as inputs to illustrate the architecture of M^{3}Att. Firstly, we use linear layers to project the language features into keys F_L^k and values F_L^v, and similarly project the vision features into F_V^k and F_V^v. Next, we use the two keys from two modalities to generate the mutual attention matrix:

$$A_{mut} = \frac{1}{\sqrt{C}} F_L^k (F_V^k)^T, \tag{5.1}$$



(a). Generic attention (b). Multi-Modal Mutual Attention (M³Att) (c). One M³Dec layer

FIGURE 5.3: The architecture of: (a). the generic attention mechanism; (b). the proposed Multi-Modal Mutual Attention (M³Att); (c). one layer of the proposed Multi-Modal Mutual Decoder (M³Dec). For three sub-figures, orange arrow: language-dominated feature flow. Blue arrow: vision-dominated feature flow. Green arrow: multi-modal feature flow.

which is a multi-modal attention matrix with shape of $N_t \times HW$, describing the relationship strength from all elements of one modality to all elements of the other modality. $\frac{1}{\sqrt{C}}$ is the scaling factor [36]. Then unlike the generic attention that only applies the attention matrix on one modality, we normalize the mutual attention matrix in both axes and apply it on features from the both modalities:

$$F_V^a = \text{softmax}(A_{mut})F_V^v, \tag{5.2a}$$

$$F_L^a = \text{softmax}({A_{mut}}^T)F_L^v. \tag{5.2b}$$

Language-attended *vision* feature (LAV), F_V^a : Softmax normalization is applied along each $HW \times 1$ axis of the mutual attention matrix A_{mut}, as in Eq. (5.2a), which is then applied on the vision feature F_V^v to get the language-attended *vision* feature $F_V^a \in R^{N_t \times C}$. There are N_t feature vectors in F_V^a, where each vector represents one attended vision feature corresponding to one element (word) in the language input. In other words, each vector is the vision feature weighted by a word based on its interpretation to the image. It is similar to the output of the generic attention mechanism. As the language features only participate in the attention matrix, the output is essentially still a single-modal feature.

Vision-attended *language* feature (VAL), F_L^a . Another softmax normalization is applied on the transposed mutual attention matrix, ${A_{mut}}^T$, along the $N_t \times 1$ axis, as in Eq. (5.2b). By applying the attention matrix on the language feature F_L^v, we get the vision-attended *language* feature $F_L^a \in R^{HW \times C}$. F_L^a contains HW feature vectors, where each vector represents one attended language feature corresponding to one pixel in the vision feature. In other words, F_L^a is a spatial-dynamic language feature, each vector of F_L^a is the language feature weighted based on a pixel's interpretation of the sentence.

Fusing of multi-modal attended features. Next, we use both attended vision and language features to generate the output. We treat each of the attended vision features in F_V^a as a dynamic kernel of a linear layer applied on F_L^a: $F_{mul} = F_V^a(F_L^a)^T$. Thus, the result is a true multi-modal feature $F_{mul} \in R^{N_t \times HW}$, where N_t is the sequence length of query and HW is the channel number. Finally, a linear layer is used to project the channel number back to C, and generates the output of

this M^{3}Att module. Notably, the mutual attention matrix A_{mul} for two softmax functions in Eq. (5.2a) and Eq. (5.2b) is default to be shared, but it can also be independently computed. More details will be discussed in the experiments. It is also worth noting that our M^{3}Att is not limited to deal with language and vision features but can accept and fuse any two modalities.

Based on M^{3}Att, we build a Multi-Modal Mutual Decoder (M^{3}Dec). M^{3}Dec has N_{dec} stacked layers , as shown in Figure 5.3. (c) for one layer. Each layer has the same architecture and takes two inputs: encoded feature and query. Here we use the first M^{3}Dec layer in our network to illustrate the layer architecture, in which the language feature F_t is taken as the query input, and transformer encoder output F_{enc} is taken as the encoded feature. Inside the layer, firstly a multi-head self-attention layer is applied on the query input, outputting a set of query features F_q. Next, a M^{3}Att module is used to fuse two sets of features: one is the query feature that is derived from the language feature and other is the transformer encoder output that has rich vision clues. The resulting multi-modal feature is further queried again by the query feature using Multi-Head Cross Attention, generating the output of this decoder layer. In this step, we use the multi-modal feature as value input, so that the output can keep its property as a multi-modal feature. The output of each M^{3}Att layer is used as the query input to its successive layer, replacing the language feature of the first layer. The output of the final layer is sent to the Mask Decoder to generate the output mask.

5.1.2 Iterative Multi-Modal Interaction

Due to the characteristic of the attention-based network, as discussed above, in M^{3}Dec, the output of the previous layer is used as the query input to the next layer. Thus, from the second layer onwards, the layer input will be the encoded feature F_{enc} and the output of its previous layer. In other words, vision information F_{enc} is directly inputted into every layer since the beginning, but in contrast, the language feature is only inputted once at the first decoder layer, as shown in Figure 5.2. This leads to a modal imbalance issue, and may cause the language information to fade away in the rear stage of the network. This issue also exists in many previous transformer-based works [20, 62]. Although M^{3}Att addresses this issue by fusing the language and vision information in the first layer using its strong multi-modal

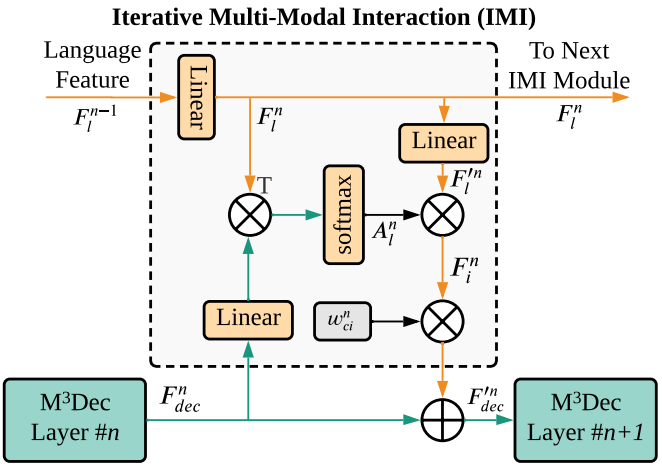


FIGURE 5.4: The architecture of one block of the Iterative Multi-Modal Interaction (IMI) module.

fusing ability, without language information inputted in the later stages, its feature fusing potential is not fully leveraged. From this point, we propose to inject the language information into M^{3}Dec at every layer.

Besides, as features are propagated to higher layers, the model's understanding of language information becomes deeper. This also causes that different layers will focus on different types of information. For example, features from lower layers do not have a contextual understanding of the relationship between language and image so that they desire more specific clues, while features from higher layers need more holistic information as they already have a better understanding of image and language. Therefore, it is desired to transform the language features along with the processing of the multi-modal feature.

Combining the above two points, we propose an Iterative Multi-Modal Interaction (IMI) module, which provides an opportunity for multi-modal features at different layers to query about their desired language information, and continuously inject them into the decoder. The IMI blocks are inserted between each two successive M^{3}Dec layers. For the n^{th} IMI block, as shown in Figure 5.4, it takes two inputs: the output of the n^{th} M^{3}Dec layer $F_{dec}^n \in R^{N_t \times C}$, and the output of the previous IMI layer $F_l^{n-1} \in R^{N_t \times C}$. F_l^{n-1} is firstly transformed with a linear layer, generating the language feature of the current layer, F_l^n. The language input for the first IMI block is the word feature F_t. With each IMI block connected to the previous one, we create a dedicated pathway for processing the language information, parallel with the process of multi-modal information.

Next, we project the multi-modal feature using a linear layer, and compute an attention matrix for reorganizing the language features:

$$A_l^n = \text{softmax}(\text{ReLU}[F_{dec}^n W_a^n](F_l^n)^T), \tag{5.3}$$

The attention matrix A_l^n is then used to reform a new language feature: $F_i^n = A_l^n F_l'^n$, where $F_l'^n = \text{ReLU}[F_l^n W_l'^n]$, as shown in Figure 5.4. The resulting feature F_i^n is then injected back into the M^{3}Dec layer output F_{dec}^n under the control of a learnable scalar w_{ci}^n, *i.e.*, $F_{dec}'^n = \text{BN}(F_{dec}^n + w_{ci}^n F_i^n)$, where BN denotes batch normalization. Using the learnable weight allows the network to determine how much information is needed by itself, and also makes the language feature more adaptable to the multi-modal feature. The output $F_{dec}'^n$ is sent to the next M^{3}Dec layer as the query input.

5.1.3 Language Feature Reconstruction

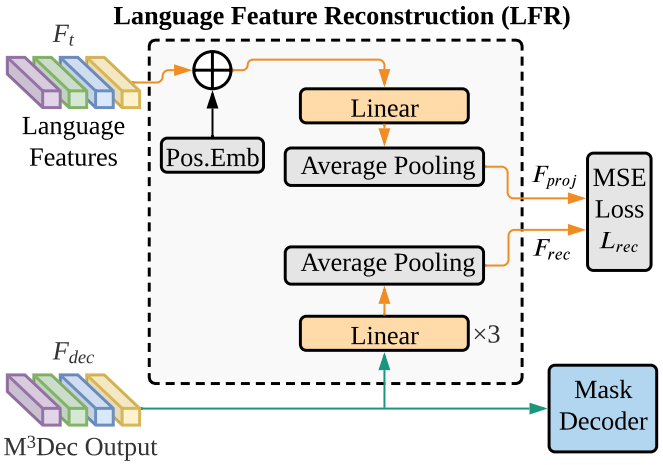


FIGURE 5.5: **The Language Feature Reconstruction (LFR) module.** Pos.Emb: Positional Embedding.

In most referring segmentation methods, the network is only supervised by the output mask loss. This implies a hypothesis: as long as the output mask matches the target object, we consider that the model has successfully understood the language information. However, this is not always true in real-world scenarios. For example, it is assumed in most datasets that there is always one and only one object in the ground-truth segmentation mask for each training sample. The network can easily learn such kind of data bias and always output one object. Therefore, for

some training samples, if the network happens to "guess" the correct target even if the language information has been lost during the propagation, these training samples may not properly contribute to the training of the network, or even be harmful to the network to generalize.

To encourage the network to be better generalized in learning from samples and improve its resistance to the language information lost, we propose a Language Feature Reconstruction (LFR) module, located at the end of the last M^{3}Dec layer in Figure 5.2. The proposed LFR module tries to reconstruct the language feature from the last M^{3}Dec output. So it ensures that the language information is well preserved through the whole multi-modal feature processing procedure. The architecture is shown in Figure 5.5. It takes language features $F_t \in R^{N_t \times C}$, $F_t' \in R^{1 \times C}$, and the output of the last M^{3}Dec layer $F_{dec} \in R^{N_t \times C}$ as inputs. The language features F_t, F_t' and the multi-modal feature F_{dec} are projected into the same feature space for comparison.

$$F_{proj} = \frac{1}{N_t + 1} \sum \mathrm{ReLU}\Big([(F_t + e) \copyright F_t']W_{proj}\Big), \tag{5.4}$$

where © is concatenation. The both © and $\sum$ are conducted along the sequence length dimension (*i.e.*, $[(F_t + e) \copyright F_t'] \in R^{(N_t+1) \times C}$). e denotes the cosine positional embedding, which adds information about the order of words in the sentence. $W_{proj} \in R^{C \times C}$ is learnable parameters for projection, and N_t is the length of the sentence for normalization.

Next, language information is reconstructed from the final multi-modal feature. As shown in Figure 5.5, we first apply three stacked linear layers on the M^{3}Dec output F_{dec}, then use an average pooling layer to shrink the sequence length dimension, producing the reconstructed language feature F_{rec}. The Language Feature Reconstruction loss is derived by minimizing the distance between the reconstructed language feature F_{rec} that is comparable with F_{proj} and the project language feature F_{proj} using the Mean Squared Error Loss.

5.1.4 Mask Decoder and Loss Function

The last step of our framework is to extract the output mask from the multi-modal features. In our framework, since we have a dense multi-modal interaction

in the M^{3}Dec, we would like the decoder to focus more on understanding the semantic clues in the inputs, and use a more vision-dominated feature to focus on the fine-grained vision details. Therefore, we choose to use both encoder output $F_{enc} \in R^{H \times W \times C}$ and M^{3}Dec output $F_{dec} \in R^{N_t \times C}$ to generate the segmentation mask as shown in Figure 5.6. The Mask Decoder firstly processes F_{dec} with a self attention module. Next, the processed decoder feature serves as kernels of a 1×1 convolutional layer. With F_{enc} as input, N_t feature maps are generated by this convolutional layer. Finally, we use four stacked convolutional layers to output the prediction mask. Upsampling layers are inserted between convolution layers to recover the spatial size of the mask.

The output mask is supervised by the Binary Cross Entropy Loss. The final loss function is defined as:

$$\mathcal{L} = w_{mask}\mathcal{L}_{mask} + w_{rec}\mathcal{L}_{rec}, \tag{5.5}$$

where w_{mask} is the weight for the mask loss $\mathcal{L}_{mask}$ and w_{rec} is the weight for the Language Feature Reconstruction loss $\mathcal{L}_{rec}$. The proposed LFR only guides the model during training, and does not participate in the mask prediction and is computationally free during inference. It can work as a plug-in module to any existing referring segmentation methods.

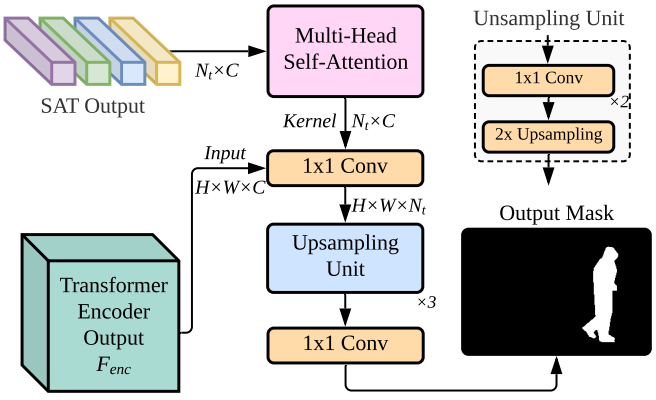


FIGURE 5.6: **The Mask Decoder** takes the output of the Mutual Attention Decoder (M^{3}Dec) and the output of the Transformer encoder to form the output mask.

5.2 Experiments

In this section, we report the experimental results of our method in comparison with previous state-of-the-art methods, and the ablation studies that verify the effectiveness of our proposed modules. We evaluate the performance by two commonly used metrics: the IoU score measures the rate of Intersection over the Union between the model's output mask and the ground-truth mask, and the Precision@X score built on IoU. Given a threshold X, the Precision@X score computes the percentage of successful predictions that have IoU scores higher than X.

5.2.1 Implementation Details

We train and evaluate the proposed approach on three commonly-used referring image segmentation datasets: RefCOCO [3], RefCOCO+ [3], and RefCOCOg [4]. Following previous works [17, 20, 62], the image features are extracted by a Darknet-53 backbone [42] pretrained on MSCOCO [78] dataset and language embeddings are generated by GloVE [97]. Language expressions are padded to 15 words for RefCOCO/RefCOCO+ and 20 words for RefCOCOg. Images from the validation set and test set of the referring segmentation datasets are excluded when training the backbone. Images are resized to 416×416 for CNN backbone following [17, 20, 76] and 480×480 for Transformer backbone following [21, 71]. Channel number C is fixed to 256 for the transformers and 512 for the mask decoder. The network has 2 encoder layers. The head number is 8 for all transformer layers. The weight for mask loss w_{mask} is set to 1 and the Language Feature Reconstruction loss w_{rec} is set to 0.1. All linear layers and convolutions layers are followed by a Batch Normalization and ReLU function unless otherwise noticed. The network is trained for 50 epochs with the batch size set to 48, using the Adam [116] optimizer. The learning rate is set to 0.005 with a step decay schedule. We use 4 NVIDIA V100 GPUs for training and testing.

5.2.2 Ablation Study

We do several ablation experiments to show the effectiveness of each proposed module in our framework. The results are reported in Table 5.1 and Table 5.2.

TABLE 5.1: Ablation results of number of layers of M^{3}Dec in different settings on the validation set of RefCOCO.

Layers	Shared		Independent		Generic (LAV only)	
	IoU	Pr@0.5	IoU	Pr@0.5	IoU	Pr@0.5
1	60.57	72.47	62.36	74.00	55.42	70.02
2	66.30	78.01	66.32	78.55	64.12	75.33
3	**67.88**	**79.01**	67.80	78.94	64.40	75.81
4	67.82	78.95	67.76	78.99	64.42	75.79

TABLE 5.2: Ablation study of components on the validation set of RefCOCO.

No.	Model	IoU	Pr@0.5
#0	Baseline (Generic LAV)	62.04	73.52
#1	Baseline (M^{3}Att)	65.56	76.66
#2	Baseline + IMI	66.70	77.62
#3	Baseline + IMI*	65.92	76.80
#4	Ours	**67.88**	**79.01**

TABLE 5.3: Ablation study of settings of the Mask Decoder on the validation set of RefCOCO.

Model	IoU	Pr@0.5	Pr@0.9
VLT[20]	66.05	78.64	13.81
Concate	66.58	78.90	15.55
Ours	**67.88**	**79.01**	**17.70**

M^{3}Att, VAL and LAV. As mentioned in Section 5.1.1, the attention matrix A_{mut} for two attended features in the M^{3}Att can be computed in two ways: shared or independent. We report the results of the two M^{3}Att settings over the generic attention mechanism in Table 5.1. For the shared setup, two A_{mul} in Eq. (5.2b) and Eq. (5.2a) are identical. For the independent setup, the attention module has two extra linear project layers applied on two inputs, generating two A_{mul}, one for LAV and the other for VAL. As shown in Table 5.1, when the layer numbers are lower, the independent setup performs better than the shared setup. However, with the increase of the layer numbers, the performance of two settings gradually gets similar. When there are 3 decoder layers, the shared setup even slightly outperforms the independent setup. We presume that this is because the independent setup has extra parameters, thus there is a performance gap when the layer numbers are smaller and the parameter numbers are not enough. We use the Shared M^{3}Att with 3 decoders as the default setup of our network.

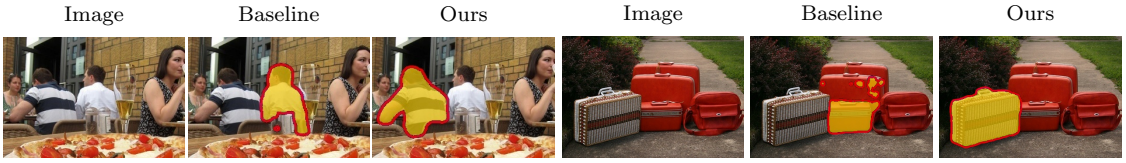
To prove the importance of the feature fusing ability of our transformer, we compare

the performance of M^{3}Att with the generic transformer, which can only generate single-modal features. Firstly, we test the generic-attention base transformer that only use the LAV feature, *i.e.*, word features serve as the query input and vision features serve as key and value input, similar as the transformer architecture in VLT [20]. The results are reported in the "Generic (LAV)" column. It can be seen that our module greatly enhances the performance, showing that multi-modal features are essential for understanding the vision and language inputs. Finally, because VAL feature are single modal language feature that are not feasible for generating masks alone, the transformer with only VAL features, *i.e.*, using language features as key/value input and using vision feature as query, fails to converge. Above two experiments show that VAL feature is a great assistance to the LAV feature.

IMI and LFR. The ablation results of IMI and LFR are reported in Table 5.2. In the baseline model, both IMI and LFR are removed. In Model #1, we validate the effectiveness of the IMI. It brings a performance gain of 1.14% in terms of IoU and 0.96% in terms of Pr@0.5. In Model #2, we verify our motivation that different layers in the M^{3}Dec need different language information. We simplify the transforming function of the IMI by replacing the F_i^n in Figure 5.4 with the language feature F_t. This makes all M^{3}Dec layers receive the same language feature. This method only gives a very slight performance improvement of 0.36% IoU over baseline, showing that by constructing the transformation pathway for language information, the IMI successfully extracts appropriate information for different feature processing stages. Finally, we add the Language Feature Reconstruction (LFR) module. Compared with Model #1, it brings an improvement on the IoU by 1.18%. Totally, the IMI and LFR bring over 2% improvement in terms of the both IoU and Pr@0.5.

Mask Decoder. In Table 5.3, we report the performance of our Mask Decoder against other variants. In the first model, we use the Mask Decoder from VLT [20], which utilizes only the output of the decoder. In the second model, rather than using the M^{3}Dec output as the convolution kernel, we sum and concatenate them with the transformer encoder output. By comparing the precision metrics, our Mask Decoder increases the Pr@0.9 metric by 3.89% from the baseline model and 2.15% from the concatenating method, showing that both the encoder and decoder

information are essential to the performance, and our mask decoder can better preserve the fine-grained image details while not losing the targeting ability.



(A) man with his back away from us in a blue and white striped shirt eating

(B) the suitcase that isn't red

FIGURE 5.7: **Qualitative comparison with the baseline model.** The proposed approach is able to solve the hard cases that cannot be handled by the baseline model.

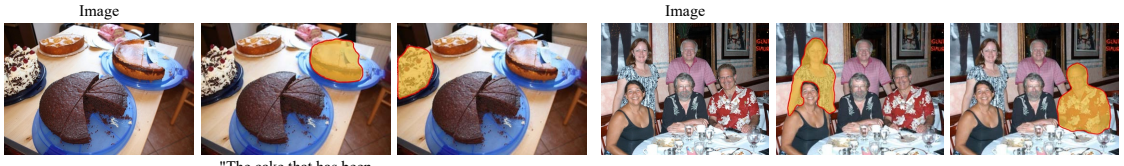
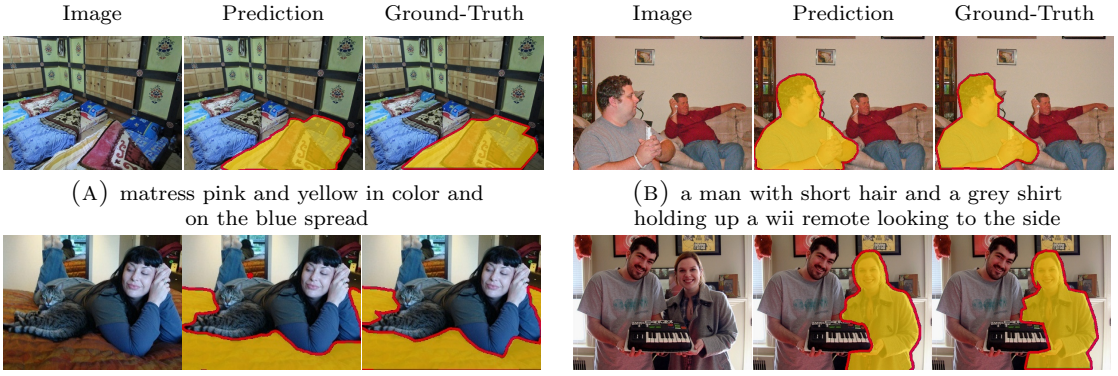


FIGURE 5.8: **More examples** showing our method finding different targets in a same image.



(A) matress pink and yellow in color and on the blue spread

(B) a man with short hair and a grey shirt holding up a wii remote looking to the side

(C) the bed that a woman and cat are laying on

(D) a woman with a grey shirt holding a cake with a man

FIGURE 5.9: **Qualitative referring segmentation examples.** The caption for each set of images is the input language expression.

Figure 5.7 displays some example results produced by the baseline model compared with our full model. In the baseline model, we replace the M^{3}Dec with generic transformer and remove the IMI and the LFR. The language expression of the first image is long, and the baseline model fails in comprehending this complex sentence. The second example in Figure 5.7 (b) shows a more tricky case, which uses a

negative sentence to target the object. The baseline model is distracted by the word "red" and gives wrong results, while our model successfully understands the sentence and finds the right object. From the examples, the language understanding ability of our approach is greatly enhanced compared with the baseline. This shows that the three proposed modules enable our approach to solve the hard and complex cases that the baseline model cannot handle.

5.2.3 Visualizations

In this section, we visualize some sample outputs of our model in Figure 5.9. To show the superior language understanding performance of our method, we use images and language expressions from the RefCOCOg dataset, of which language expressions are more natural and complex than other datasets. All examples in Figure 5.9 have long sentences with more than 10 words, and with more than two instances appearing in the text. Example (a) has a difficult sentence and a complicated layout where three mattresses are crowded in a small room. Our model has a good context understanding of the key words "matress", "pink and yellow", "blue", and their relations, and is not distracted by other mattresses and blue objects. Example (b) has a very long sentence, but most of the information is not discriminative for identifying the target, *e.g.*, both people in the image have short hair and are looking to the side. Our model detects the informative part of the sentence and targets the right object. Example (c) shows that our model can not only identify foreground objects but is also able to detect in the backgrounds. In the language expression of example (d), three objects are mentioned: "a woman", "a cake", and "a man". Our model still managed to find the subject from the difficult sentence and target the instance in the image. In Figure 5.8, we show extra examples of using multiple language expressions to refer to different objects in one image. In example (a), it can be seen that our method successfully handles complex relationships and attributes such as "has been almost half eaten" and "with cherry on it". In example (b) our method can retrieve the correct object from a complex scene. The first expression tells "standing lady" while there are two ladies in the image. Our method found the correct one. The second expression says "man in red sitting". There are three information in this expression: "man", "in red", "sitting". From the image we can see that all of the three points are necessary to find the target, *i.e.*, the target cannot be determined without any

one of the information points. The network have to understand and combine all the information in the expression. This example indicates that our network shows impressive performance on establishing the pixel-language correspondence.

5.2.4 Comparison with State-of-the-Art Methods

We report the experimental results of our method on three datasets, RefCOCO [3], RefCOCO+ [3], and RefCOCOg [4], to compare with previous state-of-the-art methods in Table 5.4. There are two data splitting types for the RefCOCOg dataset. One is referred to as the UMD split and the other is the Google split. The UMD split has both validation set and test set available, while the Google split only has validation set publicly available. We do experiments and report the results on both kinds of splitting. From Table 5.4, it can be seen that our method achieves superior performance on all datasets and outperforms previous state-of-the-art methods. On RefCOCO dataset, our method is $1.5\% - 2\%$ better than the previous SOTA, including VLT [20] and LTS [62]. On the other two datasets, our methods also have a consistent improvement of about 1.5% compared with the previous state-of-the-art methods. Also, for a fair comparison, we also implement our model with the stronger backbone Swin-Transformer[38]. It can be seen that our model with Swin-Transformer backbone also achieves a significant improvement of around 1% across most of the datasets. Especially for RefCOCO+, our model with Swin-Transformer backbone achieves about 2% improvement over the previous SOTA method VLT+[21]. This shows that our model is robust to different backbones and can achieve better performance with stronger backbones.

We also compare the Precision@X scores of the RefCOCO validation set against other methods that have data available, and the results are shown in Table 5.5. From the Pr@0.5 row, it can be seen that our model achieves the highest score. Compared with the VLT [20] that also utilizes the transformer model as prediction head, our method has an over 2% higher result in terms of Pr@0.5. The previous state-of-the-art method on the Pr@0.5 metric, MCN [17], utilizes data from both referring segmentation datasets (segmentation masks) and referring comprehension datasets (bounding boxes) in training for better locating the target, while our model only uses the segmentation mask as ground-truth. But our method achieves better targeting scores on Pr@0.5 with a large margin of 2.41%. We attribute this to

TABLE 5.4: Experimental results of the IoU metric. *: Google split.

Methods	Vis. Encoder	Lang. Encoder	ReferIt	RefCOCO			RefCOCO+			G-Ref		
			test	val	test A	test B	val	test A	test B	val	test	val*
DMN [60]	DPN92	SRU	52.81	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [59]	DL-101	LSTM	63.63	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [76]	M-RCN	LSTM	-	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [18]	DL-101	-	63.80	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
CAC [61]	R-101	LSTM	-	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
STEP [107]	DL-101	LSTM	64.13	60.04	63.46	57.97	48.19	52.33	40.41	-	-	46.40
BRINet [67]	DL-101	LSTM	63.46	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [98]	DL-101	LSTM	65.53	61.36	64.53	59.64	49.56	53.44	43.23	-	-	39.98
LSCM [64]	DL-101	LSTM	66.57	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
MCN [17]	DN-53	GRU	-	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
CMPC+ [108]	DL-101	LSTM	65.58	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
EFN [63]	R-101	GRU	-	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [65]	DL-101	S-Att.	-	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [99]	DL-101	GRU	-	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [62]	DN-53	GRU	-	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [20]	DN-53	GRU	-	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65	49.76
Ours (CNN)	DN-53	GRU	**69.33**	**67.88**	**70.82**	**65.02**	**56.98**	**61.26**	**50.11**	**54.79**	**58.21**	**50.96**
ReSTR [74]	ViT-B	Transf.	70.18	67.22	69.30	64.45	55.78	60.44	48.27	-	-	54.48
CRIS [72]	CLIP	CLIP	-	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
LAVT [71]	Swin-B	BERT	-	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
VLT+ [21]	Swin-B	BERT	-	72.96	75.96	69.60	63.53	68.43	56.92	63.49	66.22	62.80
Ours (Swin)	Swin-B	BERT	**72.97**	**73.60**	**76.23**	**70.36**	**65.34**	**70.50**	**56.98**	**64.92**	**67.37**	**63.90**

(DL: DeepLab, R: ResNet, R-MCN: ResNet+Mask R-CNN, DN: Darknet, S-Att.: Self-Attention, Transf.: Transformer)

TABLE 5.5: Results of the Precision metric on the val set of the RefCOCO.

Model	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9
LSCM [64]	70.84	63.82	53.67	38.69	12.06
CMPC [98]	71.27	64.44	55.03	39.28	12.89
MCN [17]	76.60	70.33	58.39	33.68	5.26
LTS [62]	75.16	69.51	60.74	45.17	14.41
VLT [20]	76.20	-	-	-	-
Ours	**79.01**	**74.94**	**68.16**	**51.21**	**17.70**

the better understanding of the language expression and the denser interaction of the information between the features from two modalities. This shows that our proposed modules leverage the information in the given language expression more effectively, and better fuse them with the vision information.

5.2.5 Failure Cases

We examine two typical categories of failure cases: (1) instances where the input expression refers to uncommon or unexpected areas. For instance, in example (a), the expression asks us to locate a "gap between newspaper and sandwich," which, in reality, was a part of the table. Such expressions are atypical and complex. (2) Instances where the expression is ambiguous or seeks an excessive amount of detail. In example (b), the expression "man using oven" was used. From the picture, it was apparent that both men were operating machines in the kitchen, and both machines resembled an oven. As a result, our model highlighted both individuals. Nonetheless, if we look very carefully, the machine on top also seems like a microwave. In such cases, the expression is rather ambiguous, and the model is unable to handle them. Dealing with such situations could be an interesting topic for future research.

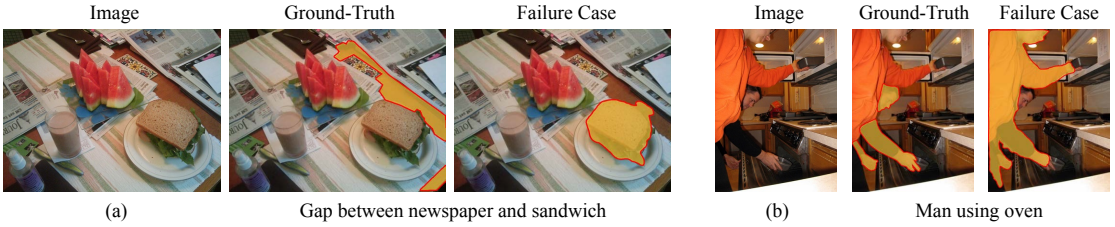


FIGURE 5.10: **Visualization of representative failure cases** of our method.

5.3 Chapter Summary

In this chapter, we address the referring image segmentation problem by designing a framework that enhances the multi-modal fusion performance. Towards this, we propose a Multi-Modal Mutual Attention (M^{3}Att) mechanism and Multi-Modal Mutual Decoder (M^{3}Dec) optimized for processing multi-modal information. Moreover, we design an Iterative Multi-Modal Interaction (IMI) scheme to further boost the feature fusing ability in the M^{3}Dec, and introduce a Language Feature Reconstruction (LFR) module to ensure that the language information is not distorted in the network. Extensive experiments show that the proposed modules can effectively promote the interactions between the language and vision information, leading the model to achieve new state-of-the-art performance on referring image segmentation.

Chapter 6

GRES: Generalized Referring Expression Segmentation

6.1 Introduction

FIGURE 6.1: Classic Referring Expression Segmentation (RES) only supports expressions that indicate a single target object, *e.g.*, the top-left sample. Compared with classic RES, the proposed **Generalized Referring Expression Segmentation (GRES)** supports expressions indicating an ***arbitrary number*** of target objects, for example, no-target expressions like the bottom-right sample and multi-target expressions like other samples.

In previous chapters, we discussed the task of classic Referring Expression Segmentation (RES). However, most classic RES methods have some strong pre-defined constraints to the task. First, the classic RES does not consider no-target expressions that do not match any object in the image. This means that the behavior of the existing RES methods is undefined if the target does not exist in the input image. When it comes to practical applications under such constraint, the input expression has to match an object in the image, otherwise problems inevitably occur. Second, most existing datasets, *e.g.*, the most popular RefCOCO [3, 4], do not contain multi-target expressions that point to multiple instances. This means that multiple inputs are needed to search objects one by one. *E.g.*, in Figure 6.1, four distinct expressions with four times of model calls are required to segment *"All people"*. Our experiments show that classic RES methods trained on existing datasets cannot be well-generalized to these scenarios.

TABLE 6.1: Comparison among different referring expression data-sets, including ReferIt[2], RefCOCO(g)[3, 4], PhraseCut[5], and our proposed gRefCOCO. Multi-target: expression that specifies multiple objects in the image. No-target: expression that does not touch on any object in the image.

	ReferIt	RefCOCO(g)	PhraseCut	gRefCOCO
Image Source	CLEF[117]	COCO[78]	VG[80]	COCO[78]
Multi-target	✗	✗	(fallback)	✓
No-target	✗	✗	✗	✓
Expression type	free	free	templated	free

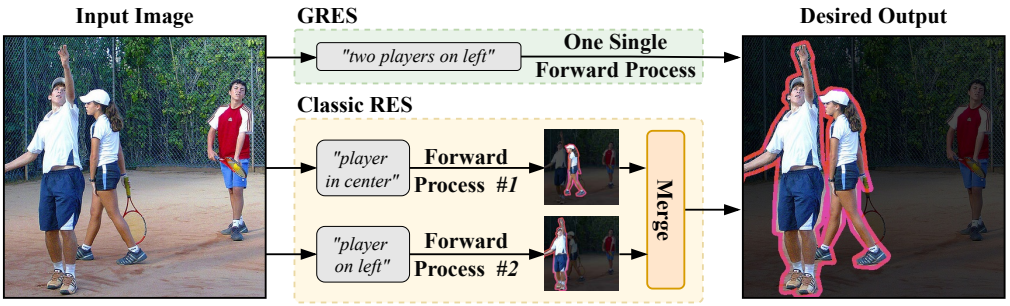
In this chapter, we propose a new benchmark, called Generalized Referring Expression Segmentation (GRES), which allows expressions indicating any number of target objects. GRES takes an image and a referring expression as input, the same as classic RES. Different from classic RES, as shown in Figure 6.1, GRES further supports multi-target expression that specifies multiple target objects in a single expression, *e.g.*, *"Everyone except the kid in white"*, and no-target expression that does not touch on any object in the image, *e.g.*, *"the kid in blue"*. This provides much more flexibility for input expression, making referring expression segmentation more useful and robust in practice. However, existing referring expression datasets [2–4] do not contain multi-target expression nor no-target samples, but only have single-target expression samples, as shown in Table 6.1. To facilitate research efforts on realistic referring segmentation, we build a new dataset for GRES, called gRefCOCO. It complements RefCOCO with two kinds of samples: multi-target samples, in which the expression points to two or more target instances in

the image, and no-target samples, in which the expression does not match any object in the image.

A baseline method. Moreover, we design a baseline method based on the objectives of the GRES task. It is known that modeling relationships, *e.g.*, region-region interactions, plays a crucial role in RES [76]. However, classic RES methods only have one target to detect so that many methods can achieve good performance without explicit region-region interaction modeling. But in GRES, as multi-target expressions involve multiple objects in one expression, it is more challenging and essential to model the long-range region-region dependencies. From this point, we propose a region-based method for GRES that explicitly model the interaction among regions with sub-instance clues. We design a network that splits the image into regions and makes them explicitly interact with each other. Moreover, unlike previous works where regions come from a simple hard-split of the input image, our network soft-collates features for each region, achieving more flexibility. We do extensive experiments on our proposed methods against other RES methods, showing that the explicit modeling of interaction and flexible region features greatly contributes to the performance of GRES.

In summary, our contributions of this chapter are listed as follows:

- We propose a benchmark of Generalized Referring Expression Segmentation (GRES), making RES more flexible and practical in real-world scenarios, by adding support to multi-target expressions and no-target expressions.
- We propose a large-scale GRES dataset gRefCOCO. To the best of our knowledge, this is the first referring expression dataset that supports expressions indicating an arbitrary number of target objects.
- We propose a solid baseline method ReLA for GRES to model complex **ReLA**tionships among objects, which achieves the new state-of-the-art performance on both classic RES and newly proposed GRES tasks.
- We do extensive experiments and comparisons of the proposed baseline method and other existing RES methods on the GRES, and analyze the possible causes of the performance gap and new challenges in GRES.



(A) Multi-target: Selecting multiple objects in one single forward process.

(B) No-target: Retrieving images that contain the object.

FIGURE 6.2: **More applications of GRES** brought by supporting multi-target and no-target expressions compared to classic RES.

6.2 Task Setting and Dataset

6.2.1 GRES Settings

Revisit of RES. Classic Referring Expression Segmentation (RES) takes an image and an expression as inputs. The desired output is a segmentation mask of the target region that is referred by the input expression. The current RES does not consider no-target expressions, and all samples in current datasets only have single-target expressions. Thus, existing models are likely to output an instance incorrectly if the input expression refers to nothing or multiple targets in the input image.

Generalized RES. To address these limitations in classic RES, we propose a benchmark called Generalized Referring Expression Segmentation (GRES) that allows expressions indicating arbitrary number of target objects. A GRES data sample contains four items: an image I, a language expression T, a ground-truth segmentation mask M_{GT} that covers pixels of all targets referred by T, and a

binary no-target label E_{GT} that indicates whether T is a no-target expression. The number of instances in T is not limited. GRES models take I and T as inputs and predict a mask M. For no-target expressions, M should be all negative.

The applications of multi-target and no-target expressions are not only finding multiple targets and rejecting inappropriate expressions matching nothing, but also bringing referring segmentation into more realistic scenarios with advanced usages. For example, with the support of multi-target expressions, we can use expressions like "*all people*" and "*two players on left*" as input to select multiple objects in a single forward process (see Figure 6.2a), or use expressions like "*foreground*" and "*kids*" to achieve user-defined open vocabulary segmentation. With the support of no-target expressions, users can apply the same expression on a set of images to identify which images contain the object(s) in the language expression, as in Figure 6.2b. This is useful if users want to find and matte something in a group of images, similar to image retrieval but more specific and flexible. What's more, allowing multi-target and no-target expressions enhances the model's reliability and robustness to realistic scenarios where any type of expression can occur unexpectedly, for example, users may accidentally or intentionally mistype a sentence.

Evaluation. To encourage the diversity of GRES methods, we do not force GRES methods to differentiate different instances in the expression though our dataset gRefCOCO provides, enabling popular one-stage methods to participate in GRES. Besides the regular RES performance metric cumulative IoU (cIoU) and Precision@X, we further propose a new metric called generalized IoU (gIoU), which extends the mean IoU to all samples including no-target ones. Moreover, No-target performance is also separately evaluated by computing No-target-accuracy (N-acc.) and Target-accuracy (T-acc.). Details are given in Section 6.4.1.

6.2.2 gRefCOCO: A Large-scale GRES Dataset

To perform the GRES task, we construct the gRefCOCO dataset. It contains 278,232 expressions, including 80,022 multi-target and 32,202 no-target expressions, referring to 60,287 distinct instances in 19,994 images. Masks and bounding boxes for all target instances are given. Part of single-target expressions are inherited from RefCOCO. We developed an online annotation tool to find images,

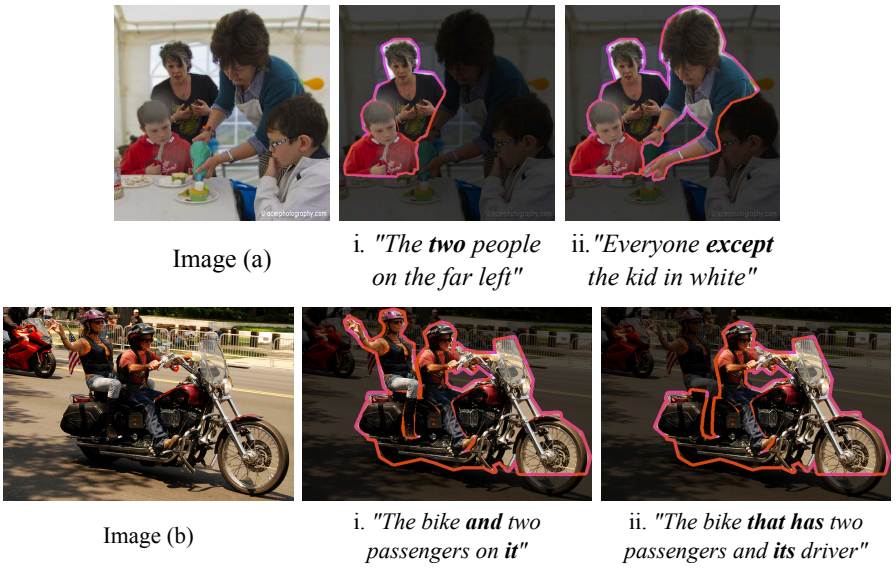


FIGURE 6.3: Examples of the proposed gRefCOCO dataset.

select instances, write expressions, and verify the results. The basic annotation procedure follows ReferIt [2] to ensure the annotation quality. The data split is also kept the same as the UNC partition of RefCOCO [3]. We compare the proposed gRefCOCO with RefCOCO and list some unique and significant features of our dataset as follows.

Multi-target samples. In practice, users usually cluster multiple targets of an image by describing their logical relationships or similarities. From this point, we let annotators select target instances rather than randomly assembling them. Then annotators write an unambiguous referring expression for the selected instances. There are four major features and challenges brought by multi-target samples:

1. **Usage of counting expressions**, *e.g.*, "*The **two** people on the far left*" in Figure 6.3(a). As the original RefCOCO already has ordinal word numbers like "*the second person from left*", the model must be able to differentiate cardinal numbers from ordinal numbers. Explicit or implicit object-counting ability is desired to address such expressions.

2. **Compound sentence structures without geometrical relation**, like compound sentences *"A **and** B"*, *"A **except** B"*, and *"A **with** B **or** C"*, as shown in Figure 6.3. This raises higher requirements for models to understand the long-range dependencies of both the image and the sentence.

3. **Domain of attributes.** When there are multiple targets in an expression, different targets may share attributes or have different attributes, *e.g.*, *"the right lady in blue and kid in white"*. Some attributes may be shared, *e.g.*, *"right"*, and others may not, *e.g.*, *"blue"* and *"white"*. This requires the model to have a deeper understanding of all the attributes and map the relationship of these attributes to their corresponding objects.

4. **More complex relationships**. Since a multi-target expression involves more than one target, relationship descriptions appear more frequently and are more complicated than in sing-target ones. Figure 6.3(b) gives an example. Two similar expressions are applied on the same image. Both expressions have the conjunction word *"and"*, and *"two passengers"* as an attribute to the target *"bike"*. But the two expressions refer to two different sets of targets as shown in Figure 6.3(b). Thus in GRES, relationships are not only used to describe the target but also indicate the number of targets. This requires the model to have a deep understanding of all instances and their interactions in the image and expression.

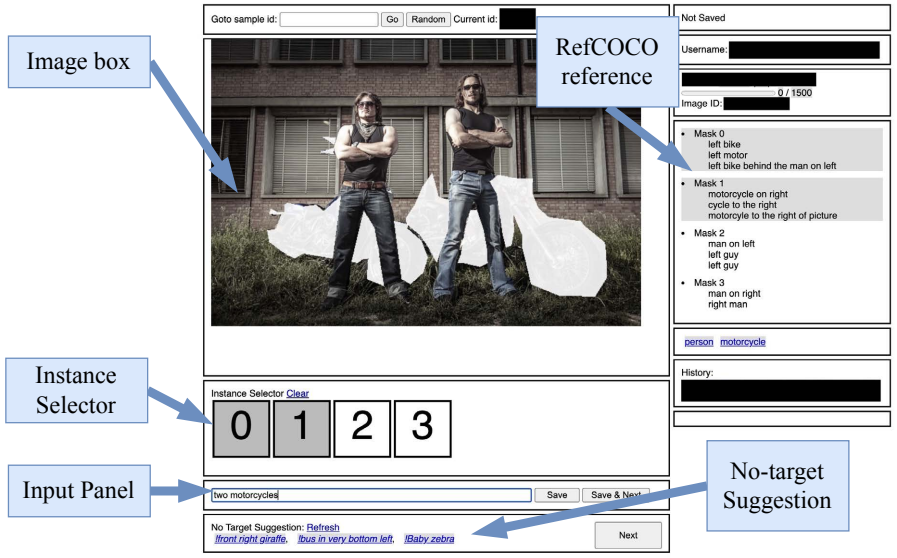
No-target samples. During the annotation, we found that if we do not set any constraints for no-target expressions, annotators tend to write a lot of simple or general expressions that are quite different from other expressions with valid targets. *E.g.*, annotators may write duplicated "*dog*" for all images without dogs. To avoid these undesirable and purposeless samples in the dataset, we set two rules for no-target expressions:

1. **The expression cannot be totally irrelevant to the image**. For example, given the image in Figure 6.3(a), "*The kid in blue*" is acceptable as there do exist kids in the image, but none of them is in blue. But expressions like "*dog*", "*car*", *"river" etc.* are unacceptable as they are totally irrelevant to anything in this image.

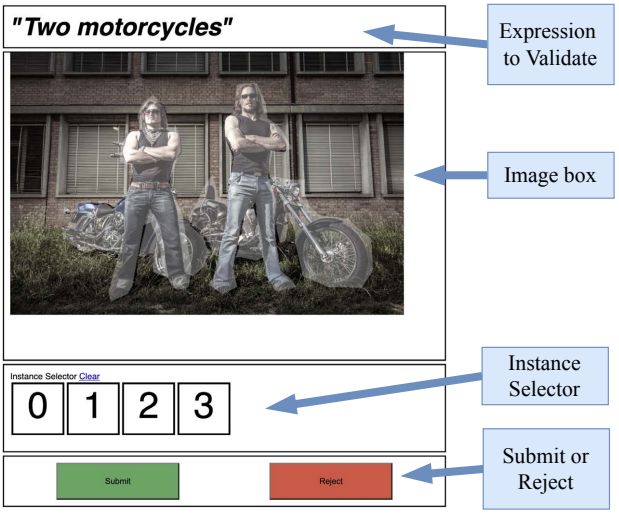
2. **The annotators could choose a deceptive expression** drawn from other images in RefCOCO's same data split, if an expression required by in 1. is hard to come up with

These rules greatly improve the diversity of no-target expressions and keep our dataset at a reasonable difficulty.

Dataset Partitioning. gRefCOCO follows the UNC splitting of RefCOCO [3] and have four non-overlapped sub-sets: *train*, *val*, *testA*, *testB*. The *train* set is a superset of the *train* set of RefCOCO, with new images from the training set of MSCOCO added. Images for validation and testing (*val*, *testA* and *testB*) are strictly identical with RefCOCO, to avoid the risk of data leakage.



(A) Annotation tool



(B) Validation tool

FIGURE 6.4: The screenshots of the developed annotation system used for building gRefCOCO.

Annotation Procedure and Tool. Following ReferIt [2], the gRefCOCO is constructed in a game-like interactive manner, in which annotations and validations

are done alternatively by two players: one annotator and one validator. We developed a web-based annotation system to facilitate the annotation and validation work. The system contains two parts: an annotation tool and a validation tool. Screenshots are shown in Figure 6.4.

Annotation. As shown in Figure 6.4a, the annotation tool can randomly draw an image from the COCO dataset, load all object masks of this image, and display them in the Image Box. The annotator is required to select a set of targets from the image using the Instance Selector, and write the referring expression in the Input Panel. The annotator is allowed to check the RefCOCO's referring expressions of this image for reference if possible. Finally, after the annotator clicks the submit button, the annotated sample will be automatically sent to the validation side.

As we mentioned in Sec. 3.2 in the main paper, our system can generate no-target expression suggestions by randomly drawing expressions of other images in RefCOCO. Annotators can either write no-target expressions by themselves or select a deceptive expression from the suggestions. All suggested expressions are drawn from the same split as the current annotating split to avoid data leakage, *e.g.*, if the annotator is annotating the *train* set of gRefCOCO, all suggestions will come from the *train* set of RefCOCO.

Validation. Figure 6.4b shows a screenshot of the validation tool. After the validation side receives a sample from the annotation side, it displays the sample's image and expression on the top of the page, then asks the validator to select and submit the targets referred by this expression. The annotator's selected targets will not be shown to the validator, so the validator needs to find targets independently. After the validator submits their selection, the backend system compares the targets found by the validator with the annotation submitted by the annotator. If they are identical, *i.e.*, the validator and the annotator independently selected the same targets, this sample is accepted as a valid gRefCOCO sample. Otherwise, this sample will be sent to another validator for a second check. Then if the second validator still fails to target this sample, it will be discarded. Validators can also directly reject samples that are inappropriate or do not meet the quality requirements. For no-target samples, the validator also needs to do a submission

without instance selection to confirm. They are also required to reject no-target expressions that are totally irrelevant to the image.

6.3 The Proposed Method for GRES

As discussed earlier, the relationship and attribute descriptions are more complex in multi-target expressions. Compared with classic RES, it is more challenging and important for GRES to model the complex interaction among regions in the image, and capture fine-grained attributes for all objects. We propose to explicitly interact different parts of image and different words in expression to analyze their dependencies.

6.3.1 Architecture Overview

The overview of our framework is shown in Figure 6.5. The input image is processed by a transformer encoder based on Swin [38] to extract vision features $F_i \in \mathbb{R}^{H\times W\times C}$, in which H, W are the spatial size and C is the channel dimensions. The input language expression is processed by BERT [45], producing the language feature $F_t \in \mathbb{R}^{N_t\times C}$, where N_t is the number of words in the expression. Next, F_i is sent to a pixel decoder to obtain the mask feature F_m for mask prediction. Meantime, F_i and F_t are sent to our proposed **ReLA**tionship modeling block (see Section 6.3.2 for details), which divides the feature maps into $P \times P = P^2$ regions, and models the interaction among them. These "regions" correspond to the image's $P \times P$ patches like ViT [118]. However, the shape and sizes of their spatial areas are not predefined but found by ReLA dynamically, which is different from previous works using hard-split [74, 103, 118, 119]. ReLA generates two sets of features: region feature $F_r = \{f_r^n\}_{n=1}^{P^2}$ and region filter $F_f = \{f_f^n\}_{n=1}^{P^2}$. For the n-th region, its region feature f_r^n is used to find a scalar x_r^n that indicates its probability of containing targets, and its region filter f_f^n is multiplied with the mask feature F_m, generating its regional segmentation mask $M_r^n \in \mathbb{R}^{H\times W}$ that indicates the area of this region. We get the predicted mask by weighted aggregating these masks:

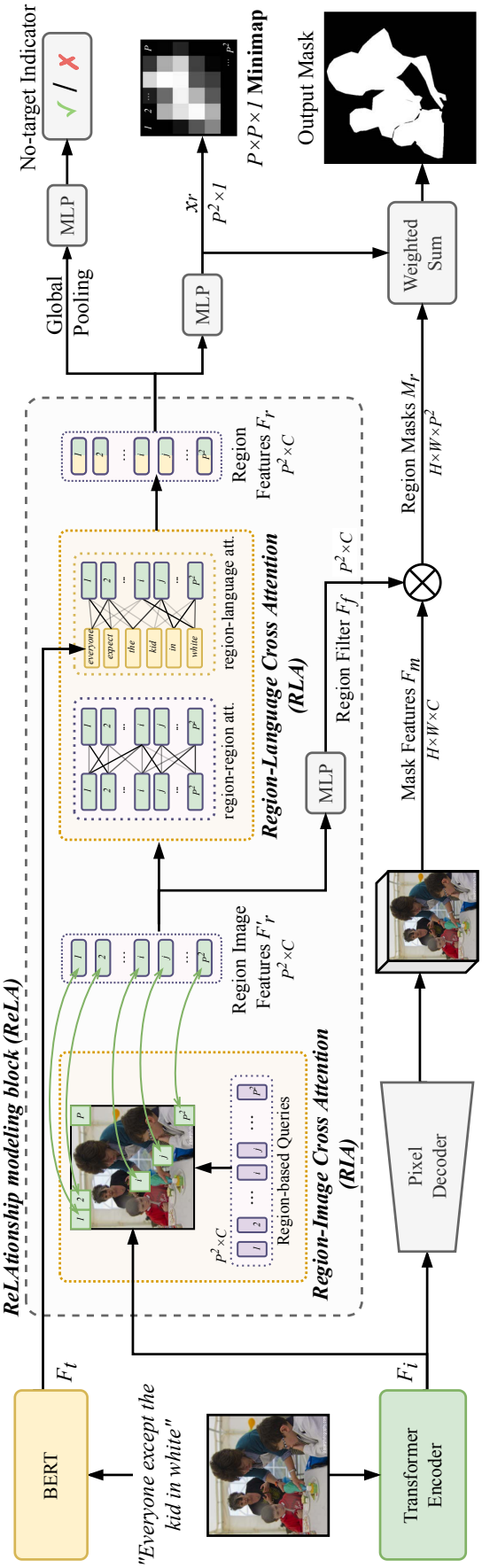


FIGURE 6.5: **Architecture overview of the GRES baseline model ReLA**. Firstly, the given image and expression are encoded into vision feature F_i and language feature F_t, respectively. F_i is fed into a pixel decoder to produce mask features F_m. **ReLA**tionship modeling block takes both F_i and F_t as inputs and output 1) region filter F_f that produces region masks M_r, 2) region probability map x_r, and 3) no-target judgement score E. Output mask is obtained by weighted fusion of region masks M_r.

$$M = \sum_{n} (x_r^n M_r^n). \tag{6.1}$$

Outputs and Loss. The predicted mask M is supervised by the ground-truth target mask M_{GT}. The $P \times P$ probability map x_r is supervised by a "minimap" downsampled from M_{GT}, so that we can link each region with its corresponding patch in the image. We also take the global average of all region features F_r to predict a no-target label E. In inference, if E is predicted to be positive, the output mask M will be set to empty. M, x_r and E are guided by the cross-entropy loss.

6.3.2 ReLAtionship Modeling

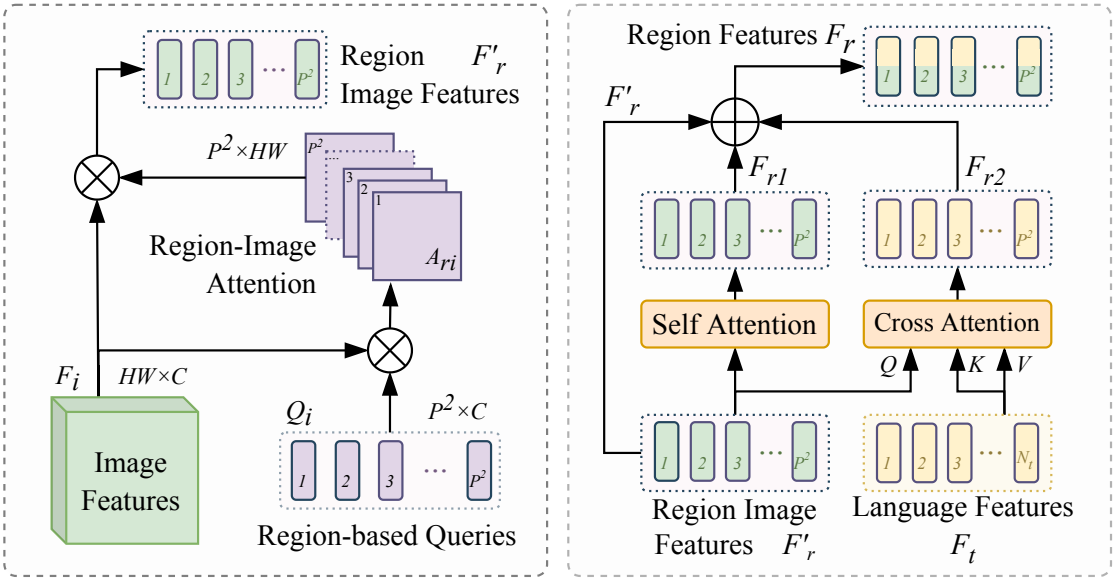
The proposed **ReLA**tionship modeling has two main modules, Region-Image Cross Attention (RIA) and Region-Language Cross Attention (RLA). The RIA flexibly collects region image features. The RLA captures the region-region and region-language dependency relationships.

Region-Image Cross Attention (RIA). RIA takes the vision feature F_i and P^2 learnable Region-based Queries Q_r as input. Supervised by the minimap shown in Figure 6.5, each query corresponds to a spatial region in the image and is responsible for feature decoding of the region. The architecture is shown in Figure 6.6a. First, the attention between image feature F_i and P^2 query embeddings $Q_r \in \mathbb{R}^{P^2 \times C}$ is performed to generate P^2 attention maps:

$$A_{ri} = \text{softmax}(Q_r \sigma(F_i W_{ik})^T), \tag{6.2}$$

where W_{ik} are $C \times C$ learnable parameters and σ is GeLU [120]. The resulting $A_{ri} \in \mathbb{R}^{P^2 \times HW}$ gives each query a $H \times W$ attention map indicating its corresponding spatial areas in the image. Next, we get the region features from their corresponding areas using these attention maps: $F_r' = A_{ri} \sigma(F_i W_{iv})^T$, where W_{iv} modalities $C \times C$ learnable parameters. In such a way, the features of each region modalities dynamically collected from their relevant positions. Compared to hard-splitting the image into patches, this method gives more flexibility. An instance may be represented by multiple regions in the minimap (see Figure 6.5), making regions

represent more fine-grained attributes at the sub-instance level, *e.g.*, the head and upper body of a person. Such sub-instance representations are desired for addressing the complex relationship and attribute descriptions in GRES. A region filter F_f containing region clues is obtained based on F'_r for mask prediction. F'_r is further fed into RLA for region-region and region-word interaction modeling.



(A) RIA (B) RLA

FIGURE 6.6: Architectures of Region-Image Cross Attention (RIA) and Region-Language Cross Attention (RLA).

Region-Language Cross Attention (RLA). Region image features F'_r come from collating image features that do not contain relationship between regions and language information. We propose RLA module to model the region-region and region-language interactions. As in Figure 6.6b, RLA consists of a self-attention for region image features F'_r and a multi-modal cross attention. The self-attention models the region-region dependency relationships. It computes the attention matrix by interacting one region feature with all other regions and outputs the relationship-aware region feature F_{r1}. The cross attention takes language feature F_t as Value and Key input, and region image feature F'_r as Query input. This firstly models the relationship between each word and each region:

$$A_l = \text{softmax}(\sigma(F'_r W_{lq})\sigma(F_t W_{lk})^T), \tag{6.3}$$

where $A_l \in \mathbb{R}^{P^2 \times N_t}$. Then it forms the language-aware region features using the derived word-region attention: $F_{r2} = A_l F_t$. Finally, the interaction-aware region feature F_{r1}, language-aware region feature F_{r2}, and region image features F'_r are added together, and a MLP further fuses the three sets of features: $F_r = \text{MLP}(F'_r + F_{r1} + F_{r2})$.

6.4 Experiments and Discussion

6.4.1 Evaluation Metrics

Besides the widely-used RES metrics cumulative IoU (cIoU) and Precision@X (Pr@X), we further introduce No-target accuracy (N-acc.), Target accuracy (T-acc.), and generalized IoU (gIoU) for GRES.

cIoU and Pr@X. cIoU calculates the total intersection pixels over total union pixels, and Pr@X counts the percentage of samples with IoU higher than the threshold X. Notably, no-target samples are excluded in Pr@X. And as multi-target samples have larger foreground areas, models are easier to get higher cIoU scores. Thus, we raise the starting threshold to 0.7 for Pr@X.

N-acc. and T-acc. evaluates the model's performance on no-target identification. For a no-target sample, prediction without any foreground pixels is true positive (TP), otherwise false negative (FN). Then, N-acc. measures the model's performance on identifying no-target samples: N-acc. $= \frac{TP}{TP+FN}$. T-acc. reflects how much the generalization on no-target affects the performance on target samples, *i.e.* how many samples that have targets are misclassified as no-target: T-acc. $= \frac{TN}{TN+FP}$.

gIoU. It is known that cIoU favors larger objects [5, 71]. As multi-target samples have larger foreground areas in GRES, we introduce generalized IoU (gIoU)[1] that

[1]Disambiguition: Not the Generalized Intersection over Union (GIoU) [121] proposed by Rezatofighi *et al.* .

treats all samples equally. Like mean IoU, gIoU calculates the mean value of per-image IoU over all samples. For no-target samples, the IoU values of true positive no-target samples are regarded as 1, while IoU values of false negative samples are treated as 0.

FIGURE 6.7: Example predictions of the same model being trained on RefCOCO *vs.* gRefCOCO.

6.4.2 Ablation Study

Dataset necessity. To show the necessity and validity of gRefCOCO on the task of GRES, we compare the results of the same model trained on RefCOCO and gRefCOCO. As shown in Figure 6.7, image (a) is a multi-target sample using a shared attribute (*"in black jacket"*) to find *"two guys"*. The model trained on RefCOCO only finds one, even though the expression explicitly points out that there are two target objects. Image (b) gives a no-target expression, and the RefCOCO-trained model outputs a meaningless mask. The results demonstrate that models trained only on single-target referring expression datasets, *e.g.*, RefCOCO, cannot be well generalized to GRES. In contrast, the newly built gRefCOCO can effectively enable the model to handle expressions indicating an arbitrary number of objects.

Design options of RIA. In Table 6.2, we investigate the performance gain brought by RIA. In model #1, we follow previous methods [74, 118] and rigidly

TABLE 6.2: Ablation study of RIA design options.

#	Methods	P@0.7	P@0.8	P@0.9	cIoU	gIoU
#1	Hard split, input	63.02	59.81	19.26	54.43	55.34
#2	Hard split, decoder	70.34	65.23	21.47	60.08	60.93
#3	w/o minimap	72.19	66.02	21.07	61.30	62.06
#4	**ReLA** (ours)	**74.20**	**68.33**	**24.68**	**62.42**	**63.60**

split the image into $P \times P$ patches before sending them into the encoder. Table 6.2 shows that this method is not suitable for our ReLA framework, because it makes the global image information less pronounced due to compromised integrity. In model #2, RIA is replaced by average pooling the image feature into $P \times P$. The gIoU gets a significant gain of 5.59% from model #1, showing the importance of global context in visual feature encoding. Then, another 2.67% gIoU gain can be got by adding our proposed dynamic region feature aggregation for each query (Eq. (6.2)), showing the effectiveness of the proposed adaptive region assigning. Moreover, we study the importance of linking queries with actual image regions. In model #3, we removed the minimap supervision so that the region-based queries Q_r become plain learnable queries, resulting in a 1.54% gIoU drop. This shows that explicit correspondence between queries and spatial image regions is beneficial to our network.

TABLE 6.3: Ablation study of RLA design options.

#	Methods	P@0.7	P@0.8	P@0.9	cIoU	gIoU
#1	Baseline	69.94	61.10	19.38	57.24	58.53
#2	+ language att.	72.03	65.42	21.04	59.86	60.53
#3	+ region att.	73.52	67.01	23.43	61.00	62.38
#4	**ReLA** (ours)	**74.20**	**68.33**	**24.68**	**62.42**	**63.60**

Design options of RLA. Table 6.3 shows the importance of dependency modeling to GRES. In the baseline model, RLA is replaced by point-wise multiplying region features and globally averaged language features, to achieve a basic feature fusion like previous works [17, 69]. In model #2, the language cross attention is added onto the baseline model, which brings a gIoU gain of 2%. This shows the validity of region-word interaction modeling. Then we further add the region self-attention to investigate the importance of the region-region relationship. The region-region relationship modeling brings a performance gain of 3.85% gIoU. The region-region and region-word relationship modeling together bring a significant improvement of 5.07% gIoU.

TABLE 6.4: Ablation study of Number of Regions

# Regions	P@0.7	P@0.8	P@0.9	cIoU	gIoU
4×4	68.48	60.25	20.33	56.57	57.01
8×8	72.36	66.85	23.56	59.74	61.23
10×10	**74.20**	**68.33**	**24.68**	**62.42**	**63.60**
12×12	74.14	67.56	23.90	62.02	63.50

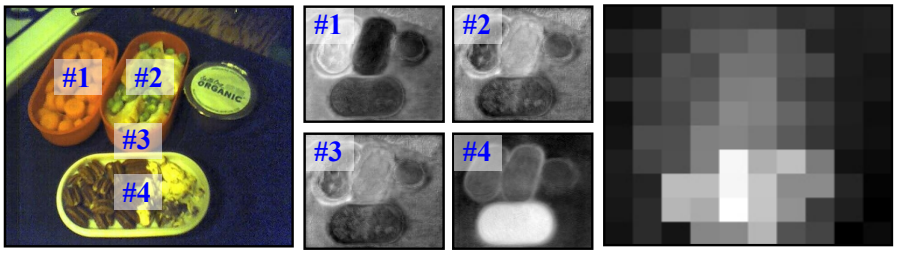


FIGURE 6.8: Visualization of the predicted minimap & region masks.

Number of regions P. Smaller P leads to coarser regions, which is not good for capturing fine-grained attributes, while larger P costs more resources and decreases the area of each region, making relationship learning difficult. We do experiments on the selection of P in Table 6.4 to find the optimized P. The model's performance improves as P increases until 10, which is selected as our setting. In Figure 6.8, we visualize the predicted minimap x_r and region maps M_r. x_r displays a rough target probability of each region, showing the effectiveness of minimap supervision. We also see that the region masks capture the spatial correlation of the corresponding regions. With flexible region size and shape, each region mask contains not only the instance of this region but also other instances with strong relationships. For example, region #4 is located inside the bottom lunch box, but as the input expression tells that all three boxes are targets, the top two also cause some responses in the output mask of region #4.

TABLE 6.5: Comparison on gRefCOCO dataset.

Methods	val		testA		testB	
	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU
MattNet [76]	47.51	48.24	58.66	59.30	45.33	46.14
LTS [62]	52.30	52.70	61.87	62.64	49.96	50.42
VLT [69]	52.51	52.00	62.19	63.20	50.52	50.88
CRIS [72]	55.34	56.27	63.82	63.42	51.04	51.79
LAVT [71]	57.64	58.40	65.32	65.90	55.04	55.83
VLT+ReLA	58.65	59.43	66.60	65.35	56.22	57.36
LAVT+ReLA	61.23	61.32	67.54	66.40	58.24	59.83
ReLA (ours)	**62.42**	**63.60**	**69.26**	**70.03**	**59.88**	**61.02**

TABLE 6.6: No-target results comparison on gRefCOCO dataset.

Methods	val		testA		testB	
	N-acc.	T-acc.	N-acc.	T-acc.	N-acc.	T-acc.
MattNet [76]	41.15	96.13	44.04	97.56	41.32	95.32
VLT [69]	47.17	95.72	48.74	95.86	47.82	94.66
LAVT [71]	49.32	96.18	49.25	95.08	48.46	95.34
ReLA-50pix	49.96	96.28	51.36	96.35	49.24	95.02
ReLA	**56.37**	**96.32**	**59.02**	**97.68**	**58.40**	**95.44**

6.4.3 Results on GRES

Comparison with state-of-the-art RES methods. In Table 6.5, we report the results of classic RES methods on gRefCOCO. We re-implement these methods using the same backbone as our model and train them on gRefCOCO. For one-stage networks, output masks with less than 50 positive pixels are cleared to all-negative, for better no-target identification. For the two-stage network MattNet [76], we let the model predict a binary label for each instance that indicates whether this candidate is a target, then merge all target instances. As shown in Table 6.5, these classic RES methods do not perform well on gRefCOCO that contains multi-target and no-target samples. Furthermore, to better verify the effectiveness of explicit modeling, we add our ReLA on VLT [69] and LAVT [71] to replace the decoder part of them. From Table 6.5, our explicit relationship modeling greatly enhances model's performance. *E.g.*, adding ReLA improves the cIoU performance of the LAVT by more than 4% on the val set.

In Table 6.6, we test the no-target identification performance. As shown in the table, T-acc. of all methods are mostly higher than 95%, showing that our gRefCOCO does not significantly affect the model's targeting performance while being generalized to no-target samples. But from N-acc. of classic RES methods, we see that even being trained with no-target samples, it is not satisfactory to identify no-target samples solely based on the output mask. We also tested our model with the no-target classifier disabled and only use the positive pixel count in the output mask to identify no-target samples ("ReLA-50pix" in Table 6.6). The performance is similar to other methods. This shows that a dedicated no-target classifier is desired. However, although our N-acc. is higher than RES methods, there are still around 40% of no-target samples are missed. We speculate that this is because many no-target expressions are very deceptive and similar with real instances in the image. Given the current suboptimal performance, coupled with the broad potential applications of no-target sample detection — as illustrated but not limited

TABLE 6.7: Results on classic RES in terms of cIoU. U: UMD split. G: Google split.

Methods	Visual Encoder	Textual Encoder	RefCOCO			RefCOCO+			G-Ref		
			val	test A	test B	val	test A	test B	$\text{val}_{(U)}$	$\text{test}_{(U)}$	$\text{val}_{(G)}$
MCN [17]	Darknet53	bi-GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
VLT [69]	Darknet53	bi-GRU	67.52	70.47	65.24	56.30	60.98	50.08	54.96	57.73	52.02
ReSTR [74]	ViT-B	Transformer	67.22	69.30	64.45	55.78	60.44	48.27	-	-	54.48
CRIS [72]	CLIP-R101	CLIP	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
LAVT [71]	Swin-B	BERT	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
VLT [21]	Swin-B	BERT	72.96	75.96	69.60	63.53	68.43	56.92	63.49	**66.22**	**62.80**
ReLA (ours)	Swin-B	BERT	**73.82**	**76.48**	**70.18**	**66.04**	**71.02**	**57.65**	**65.00**	65.97	62.70

to those shown in Figure 6.2b — we believe that no-target identification will be one of our key focus on the future research for the GRES task.

Qualitative results. Some qualitative examples of our model on the val set of gRefCOCO are shown in Figure 6.9. In Image (a), our model can detect and precisely segment multiple targets of the same category (*"girls"*) or different categories (*"girls and the dog"*), showing the strong generalization ability. Image (b) uses counting words (*"two bowls"*) and shared attributes (*"on right"*) to describe a set of targets. Image (c) has a compound sentence showing that our model can understand the excluding relationship: *"except the blurry guy"* and makes a good prediction.

FIGURE 6.9: Example results of our method on gRefCOCO dataset.

Failure cases & discussion. We show some failure cases of our method in Figure 6.10. Image (a) introduces a possession relationship: *"left girl and **her** laptop"*. This is a very deceptive case. In the image, the laptop in center is more dominant and closer to the left girl than the left one, so the model highlighted the center laptop as *"her laptop"*. Such a challenging case requires the model to have

a profound understanding of all objects, and a contextual comprehension of the image and expression. In the second case, the expression is a no-target expression, referring to *"man in gray shirt sitting on bed"*. In the image, there is indeed a sitting person in grey shirt, but he is sitting on a black chair very close to the bed. This further requires the model to look into the fine-grained details of all objects, and understand those details with image context.

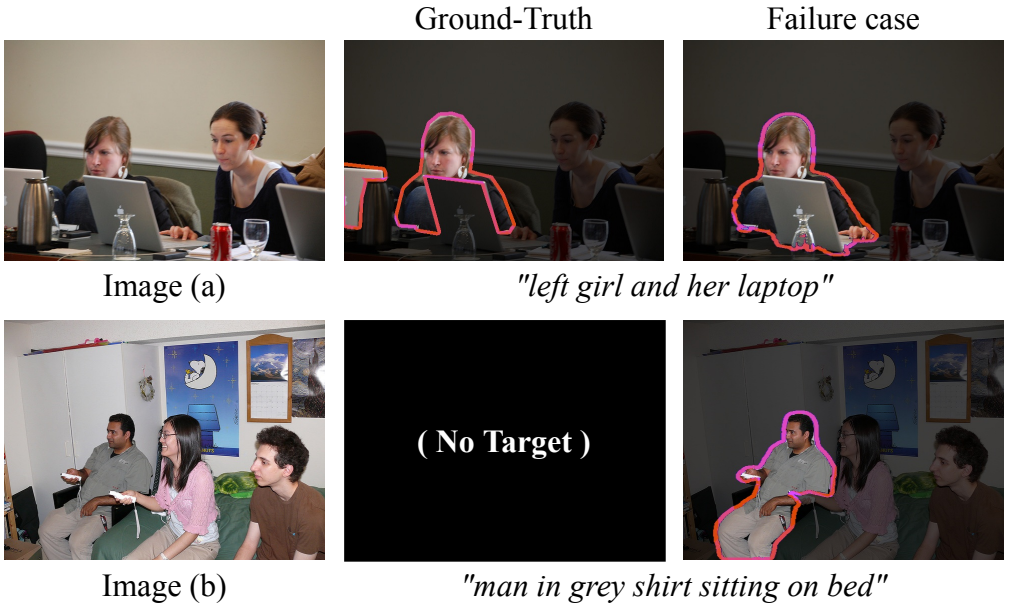


FIGURE 6.10: Failure cases of our method on gRefCOCO dataset.

6.4.4 Results on Classic RES

We also evaluate our method on the classic RES task and report the results in Table 6.7. In this experiment, our model strictly follows the setting of previous methods [69, 71] and is only trained on the RES datasets. As shown in Table 6.7, the proposed approach ReLA outperforms other methods on classic RES. Our performance is consistently higher than the state-of-the-art LAVT [71] with a margin of 1%∼4% on three datasets. Although the performance gain of our proposed method over other methods on classic RES is lower than that on GRES, the results show that the explicit relationship modeling is also beneficial to classic RES.

6.5 Chapter Summary

We analyze and address the limitations of the classic RES task, *i.e.*, it cannot handle multi-target and no-target expressions. Based on that, a new benchmark, called Generalized Referring Expression Segmentation (GRES), is defined to allow an arbitrary number of targets in the expressions. To support the research on GRES, we construct a large-scale dataset gRefCOCO. We propose a baseline method ReLA for GRES to explicitly model the relationship between different image regions and words, which consistently achieves new state-of-the-art results on the both classic RES and newly proposed GRES tasks. The proposed GRES greatly reduces the constraint to the natural language inputs, increases the application scope to the cases of multiple instances and no right objects in image, and opens possible new applications such as image retrieval.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

In this thesis, we investigate the task of Referring Expression Segmentation (RES). We found four major challenges for the task. For CNN-based frameworks, we found that both one-stage and two-stage methods have their own limitations. A new framework is needed to be designed to achieve better performance. Next, we argue that the way that CNN achieves large receptive fields is not perfect for modeling long-range dependencies. and it is worth to try to change CNN with new meta-architecture. We found that for attention-based models, the generic dot-product attention mechanism is not optimized for fusing multi-modal features, which is also not desirable for the performance of attention-based RES models. Finally, the "single-target only" constraint for classic RES also limits the practical usages of classic RES models.

From these challenges and gaps, we propose four new models and a new task to address these issues. In the first model, we combine the one-stage and two-stage RES methods together, making it possible to analyse the relationship among instances without introducing an extra instance segmentation network. Secondly, We propose to employ the attention-based Transformer to replace the CNN prediction head, achieving better performance. Further, in the third model, we propose to empower the generic dot-product attention with multi-modal feature fusing functionality, making it more suitable for the RES task. Finally, we propose a new

benchmark called GRES, which add supports to multi-target and no-target expressions, making the RES task more practical.

Based on the challenges and motivations discussed in this thesis, we summarize and conclude the contributions of this thesis and the four works that have been proposed as follows:

1. In the first model, we have tackled the challenging task of referring segmentation. We have introduced a new framework that addresses the limitations of previous methods while incorporating their strengths. Our approach simultaneously segments instances in the image and establishes interactions among them for identification. We have introduced a feature propagation module to capture the comparison relationships between instances. Additionally, we have proposed a refinement module that leverages low-level and high-resolution features to enhance the spatial details of the predicted segmentation mask. The experimental results demonstrate that our approach significantly improves both the targeting performance and the quality of the segmentation mask. By focusing on effectiveness rather than unnecessary complexity, we have achieved new state-of-the-art results on three referring segmentation datasets. This confirms the efficacy of our proposed approach.

2. In the second model, we have addressed the challenging task of referring segmentation using a multi-modal approach. We have introduced the transformer to facilitate long-range information exchange, overcoming the limitations of traditional convolutional networks. Our Vision-Language Transformer (VLT) framework formulates referring segmentation as a direct attention problem, incorporating spatial-dynamic multi-modal fusion to emphasize pixel/object differences. To handle ambiguous referring expressions, we propose the Query Generation Module and Query Balance Module, improving understanding through the referred image information. Additionally, inter-sample learning enhances comprehension of different language expressions for one object, while masked contrastive representation learning enables distinguishing features for the same object and different objects. The lightweight VLT model achieves new state-of-the-art performance on three referring image segmentation datasets and two referring video object segmentation datasets.

3. In the third model, our focus is on improving referring image segmentation by introducing a framework that enhances the performance of multi-modal fusion. To achieve this, we propose two key components: the Multi-Modal Mutual Attention (M^3Att) mechanism and the Multi-Modal Mutual Decoder (M^3Dec), both specifically designed to handle multi-modal information. Additionally, we propose an Iterative Multi-Modal Interaction (IMI) scheme to further enhance the feature fusion capability of the M^3Dec. To ensure the integrity of the language information, we introduce a Language Feature Reconstruction (LFR) module. Through extensive experiments, we demonstrate the effectiveness of these modules in promoting interactions between language and vision information. As a result, our proposed approach achieves new state-of-the-art performance in referring image segmentation.

4. Moreover, we analyze and address the limitations of the classic Referring Expression Segmentation (RES) task, which lacks the ability to handle multi-target and no-target expressions effectively. To overcome these limitations, we introduce the Generalized Referring Expression Segmentation (GRES) benchmark, allowing for arbitrary numbers of targets in expressions. Additionally, we construct a large-scale dataset called gRefCOCO to support research on GRES. We propose ReLA, a baseline method for GRES that explicitly models the relationship between image regions and words, achieving consistently improved results in both the classic RES and GRES tasks. The introduction of GRES reduces constraints on natural language inputs, expands application possibilities to cases with multiple instances and no correct objects, and opens up new potential applications such as image retrieval.

7.2 Future Works

Alghouth a lot of work has been done in this thesis, there are still a lot of things to do in the future. We will discuss the future works and the potential improvements of our proposed models in the following aspects:

- **Weakly or self-supervised referring segmentation.** Current RES task is fully supervised, and heavily relys on the annotations of the dataset. However, the annotation process is very expensive, especially for RES tasks that

requires both segmentation mask and referring expressions as annotations. Therefore, it would be very useful if we can train the model with weakly supervised data, or develop self-supervised models for the RES task. Recent works have shown potential on contrastive learning [122–124] and self-supervised learning [49], but most of them learn from general data, which is not suitable for training RES models. One possible way is that we may combine the task of referring expression generation and segmentation together, but this is still an open question. We will investigate the possibilities of this direction in the future.

- **Generative referring expression processing.** The task of RES is a typical discriminative task, which means that the model is trained to predict the segmentation mask given the referring expression and the image. However, it would be more interesting if the model can generate referring instructions, such as *"replace the person on the right with a dog"*. This would be a great enhancement on the functionality of the RES task, and it would be very useful for many applications, such as image editing. With the recent development of generative models such as the Latent Diffusion model [125], we believe that this is a promising direction for the RES task.

- **Intergrating RES with other tasks.** The RES task is an integrated task of semantic segmentation and object detection. However, it is still a standalone task. We believe that the RES task can be integrated with other tasks, such as image matting and image retrieval. For example, we can use the RES model to generate the segmentation mask of the image, and then use image matting techniques to find an accurate alpha matte of the object, or we can use a referring expression to search objects in a large database. This would be a very interesting direction to explore.

- **Utilizing Large Language Models (LLM) and multi-modal models.** Thanks to models like Swin Transformer and BERT, we have already witnessed a performance boost in the RES task by using larger models. This shows that the scale backbone model is very important for the performance of RES. However, most recently, multi-modal models and "large models", such as CLIP [50] and GPT-4 [49], have been proposed. These models are trained on much larger datasets, and have achieved impressive performance on many tasks and some of them are believed to have human's "common

knowledge" [49]. We already seen CLIP has been utilized in the RES task [72], however, its performance does not significantly surpass other regular transformer based works. This shows that how to utilize these large models in the RES task is still an open question. We believe that the performance of RES models can be further improved by utilizing these large models.

List of Author's Publications

- **Chang Liu**, Henghui Ding, and Xudong Jiang. (2021). Towards enhancing fine-grained details for image matting. In IEEE/CVF Winter Conference on Applications of Computer Vision 2021 (WACV 2021).
- Henghui Ding*, **Chang Liu***, Suchen Wang, and Xudong Jiang. (2021). Vision-language transformer and query generation for referring segmentation. In IEEE/CVF International Conference on Computer Vision 2021 (ICCV 2021).
- Henghui Ding, Hui Zhang, **Chang Liu**, and Xudong Jiang. (2022). Deep interactive image matting with feature propagation. IEEE Transactions on Image Processing (TIP).
- **Chang Liu**, Xudong Jiang, and Henghui Ding. (2022). Instance-specific feature propagation for referring segmentation. IEEE Transactions on Multimedia (TMM).
- Henghui Ding*, **Chang Liu***, Suchen Wang, and Xudong Jiang. (2022). VLT: Vision-Language Transformer and Query Generation for Referring Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
- **Chang Liu**, Henghui Ding, and Xudong Jiang. (2023). GRES: Generalized Referring Expression Segmentation. In The IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (CVPR 2023, highlight).
- **Chang Liu**, Henghui Ding, Yunlun Zhang, and Xudong Jiang. (2023). Multi-Modal Mutual Attention and Iterative Interaction for Referring Image Segmentation. IEEE Transactions on Image Processing (TIP).

- Henghui Ding*, **Chang Liu***, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. (2023). MOSE: A new dataset for video object segmentation in complex scenes. In IEEE/CVF International Conference on Computer Vision 2023 (ICCV 2023) (under review).

- **Chang Liu**, Henghui Ding, Gang Wang, and Xudong Jiang. (2022). PrimitiveNet: Decomposing the Global Constraints for Referring Segmentation. Pattern Recognition (PR) (under review).

*: equal contribution, co-first authorship.

Bibliography

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. Eur. Conf. Comput. Vis.*, pages 213–229. Springer, 2020. xxi, 10, 17, 43, 49, 53, 59, 61

[2] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proc. of the Conf. on Empirical Methods in Natural Language Process.*, pages 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL `https://www.aclweb.org/anthology/D14-1086`. xxii, 18, 96, 100, 102

[3] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Proc. Eur. Conf. Comput. Vis.*, pages 69–85. Springer, 2016. xxii, 19, 33, 85, 90, 96, 100, 102

[4] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11–20, 2016. xxii, 19, 33, 85, 90, 96

[5] Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. Phrasecut: Language-based image segmentation in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10216–10225, 2020. xxii, 20, 96, 108

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1, 10

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016. 10, 68, 71

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

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7132–7141, 2018. 1, 10, 27

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017. 1, 10, 29, 68

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3431–3440, 2015. 1, 10, 13, 16, 17, 42

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 1, 27, 60, 68

[15] Ping Hu, Jun Liu, Gang Wang, Vitaly Ablavsky, Kate Saenko, and Stan Sclaroff. Dipnet: Dynamic identity propagation network for video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1904–1913, 2020. 1

[16] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *Proc. Eur. Conf. Comput. Vis.*, pages 108–124. Springer, 2016. 3, 16, 17, 24, 42, 48, 56, 76

[17] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10034–10043, 2020. 16, 17, 33, 34, 43, 48, 51, 60, 65, 76, 85, 90, 91, 92, 110, 113

[18] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10502–10511, 2019. 3, 16, 24, 34, 42, 48, 65, 71, 91

[19] Chang Liu, Xudong Jiang, and Henghui Ding. Instance-specific feature propagation for referring segmentation. *IEEE Trans. Multimedia*, 2022. 6, 17

[20] Henghui Ding, Hui Zhang, Jun Liu, Jiaxin Li, Zijian Feng, and Xudong Jiang. Interaction via bi-directional graph of semantic region affinity for scene parsing. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2021. 6, 73, 75, 76, 77, 80, 85, 86, 87, 90, 91, 92

[21] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vlt: Vision-language transformer and query generation for referring segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 6, 85, 90, 91, 113

[22] Chang Liu, Henghui Ding, Yulun Zhang, and Xudong Jiang. Multi-modal mutual attention and iterative interaction for referring image segmentation. *IEEE Transactions on Image Processing*, 2023. 7

[23] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23592–23601, 2023. 7

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. 10, 14, 17, 20, 24, 68

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998. 10

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2012. 10

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 10

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 10

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 10

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 10

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 10

[33] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 10

[34] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 10

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 11

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 11, 12, 16, 42, 59, 79

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017. 13

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 13, 59, 68, 71, 90, 104

[39] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 13

[40] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 13

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 13

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 779–788, 2016. 14, 24, 28, 60, 68, 85

[43] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[44] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 14, 24

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, volume 1, pages 4171–4186. Association for Computational Linguistics, 2019. 14, 59, 68, 104

[46] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 2018. 14

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 14

[48] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 15

[49] OpenAI. Gpt-4 technical report, 2023. 15, 120, 121

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 15, 17, 68, 120

[51] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4555–4564, 2016. 16

[52] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1960–1968, 2019.

[53] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4673–4682, 2019. 20

[54] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4683–4693, 2019.

[55] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4252–4261, 2018. 20

[56] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *Proc. Eur. Conf. Comput. Vis.*, volume 12359, pages 387–404. Springer, 2020. 21

[57] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10880–10889, 2020. 16, 21

[58] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1271–1280, 2017. 16, 17, 48

[59] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5745–5753, 2018. 16, 34, 65, 91

[60] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *Proc. Eur. Conf. Comput. Vis.*, pages 630–645, 2018. 16, 34, 42, 65, 91

[61] Y.-W. Chen, Y.-H. Tsai, T. Wang, Y.-Y. Lin, and M.-H. Yang. Referring expression object segmentation with caption-aware consistency. In *Proc. Brit. Mach. Vis. Conf.*, 2019. 16, 65, 91

[62] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9858–9867, 2021. 16, 17, 48, 65, 68, 73, 75, 80, 85, 90, 91, 92, 111

[63] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021. 16, 17, 65, 91

[64] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *Proc. Eur. Conf. Comput. Vis.*, pages 59–75. Springer, 2020. 16, 34, 65, 91, 92

[65] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021. 16, 17, 65, 91

[66] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7794–7803, 2018. 16, 42

[67] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4424–4433, 2020. 16, 34, 42, 48, 65, 91

[68] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr – modulated detection for end-to-end multi-modal understanding. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2021. 17

[69] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 16321–16330, 2021. 17, 48, 110, 111, 112, 113, 115

[70] Zizhang Li, Mengmeng Wang, Jianbiao Mei, and Yong Liu. Mail: A unified mask-image-language trimodal network for referring image segmentation. *arXiv preprint arXiv:2111.10747*, 2021. 17, 65, 68, 69

[71] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 18155–18165, 2022. 17, 65, 69, 73, 85, 91, 108, 111, 112, 113, 115

[72] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11686–11695, 2022. 17, 65, 68, 91, 111, 113, 121

[73] Kanishk Jain and Vineet Gandhi. Comprehensive multi-modal interactions for referring image segmentation. *arXiv preprint arXiv:2104.10412*, 2021.

[74] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Restr: Convolution-free referring image segmentation using transformers. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 18145–18154, 2022. 17, 65, 91, 104, 109, 113

[75] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proc. Int. Conf. Mach. Learn.*, pages 5583–5594. PMLR, 2021. 17, 68

[76] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1307–1315, 2018. 17, 20, 24, 25, 33, 34, 43, 51, 60, 65, 85, 91, 97, 111, 112

[77] Hugo Jair Escalante, Carlos A Hernández, Jesus A Gonzalez, Aurelio López-López, Manuel Montes, Eduardo F Morales, L Enrique Sucar, Luis Villasenor, and Michael Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer vision and image understanding*, 114(4):419–428, 2010. 19

[78] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 19, 85, 96

[79] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *Proc. Eur. Conf. Comput. Vis.*, pages 792–807. Springer, 2016. 19, 33

[80] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73, 2017. 20, 96

[81] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5958–5966, 2018. 20

[82] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2264–2273, 2015. 20

[83] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, December 2013. 20

[84] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Proc. Asi. Conf. Comput. Vis.*, pages 123–141. Springer, 2018. 20, 71

[85] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Proc. Eur. Conf. Comput. Vis.*, pages 208–223. Springer, 2020. 20, 71

[86] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020. 20

[87] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 20

[88] Henghui Ding, Scott Cohen, Brian Price, and Xudong Jiang. Phraseclick: toward achieving flexible interactive segmentation by phrase and click. In *Proc. Eur. Conf. Comput. Vis.*, pages 417–435. Springer, 2020. 20

[89] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1115–1124, 2017. 20

[90] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4158–4166, 2018.

[91] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(2):684–696, 2022. 20

[92] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. *arXiv preprint arXiv:1812.03426*, 2018. 21

[93] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1769–1779, 2021. 21

[94] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4694–4703, 2019. 21

[95] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations, 2019. 24, 28

[96] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2117–2125, 2017. 27, 28

[97] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proc. of the Conf. on Empirical Methods in Natural Language Process.*, pages 1532–1543, 2014. 27, 33, 60, 85

[98] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10488–10497, 2020. 33, 34, 51, 65, 91, 92

[99] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped attention network for referring expression segmentation. In *ACM Int. Conf. Multimedia*, pages 1274–1282, 2020. 34, 65, 91

[100] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *Proc. Eur. Conf. Comput. Vis.*, pages 38–54, 2018. 42

[101] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3286–3295, 2019. 49

[102] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6881–6890, 2021. 49

[103] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Proc. Adv. Neural Inform. Process. Syst.*, volume 34, pages 12077–12090, 2021. 104

[104] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 17864–17875, 2021. 49

[105] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018. 57

[106] Hui Zhang and Henghui Ding. Prototypical matching and open set rejection for zero-shot semantic segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 6974–6983, 2021. 59

[107] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 7454–7463, 2019. 65, 91

[108] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):4761–4775, 2022. 65, 91

[109] Zihan Ding, Tianrui Hui, Shaofei Huang, Si Liu, Xuan Luo, Junshi Huang, and Xiaoming Wei. Progressive multimodal interaction network for referring video object segmentation. *The 3rd Large-scale Video Object Segmentation Challenge*, page 7, 2021. 71

[110] Chen Liang, Yu Wu, Tianfei Zhou, Wenguan Wang, Zongxin Yang, Yunchao Wei, and Yi Yang. Rethinking cross-modal interaction from a top-down perspective for referring video object segmentation. *arXiv preprint arXiv:2106.01061*, 2021. 71

[111] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4974–4984, 2022. 71

[112] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 71

[113] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3202–3211, June 2022. 71

[114] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3349–3364, 2020. 71

[115] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 2736–2746, 2022. 71

[116] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 85

[117] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International workshop ontoImage*, volume 2, 2006. 96

[118] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Represent.*, 2021. 104, 109

[119] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 7262–7272, 2021. 104

[120] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 106

[121] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 108

[122] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 120

[123] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[124] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 120

[125] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10684–10695, 2022. 120